UNIVERSITY OF BELGRADE
FACULTY OF CIVIL ENGINEERING

Dušan M. Isailović

DIGITAL REPRESENTATION OF AS-DAMAGED REINFORCED CONCRETE BRIDGES

Doctoral Dissertation

Belgrade, 2020

УНИВЕРЗИТЕТ У БЕОГРАДУ
ГРАЂЕВИНСКИ ФАКУЛТЕТ

Душан М. Исаиловић

ДИГИТАЛНИ ПРИКАЗ ОШТЕЋЕНИХ АРМИРАНОБЕТОНСКИХ МОСТОВА

докторска дисертација

Београд, 2020

SUPERVISOR AND THE COMMITTEE

Supervisor:

Prof. Dr. Rade Hajdin
Faculty of Civil Engineering, University of Belgrade

Committee:

Prof. Dr. Rade Hajdin
Faculty of Civil Engineering, University of Belgrade

Prof. Dr. Markus König
Faculty of Civil and Environmental Engineering,
Ruhr University, Bochum, Germany

Prof. Dr. Miloš Kovačević
Faculty of Civil Engineering, University of Belgrade

Ass. Prof. Dr. Đorđe Nedeljković
Faculty of Civil Engineering, University of Belgrade

Dr. Igor Svetel
Innovation Center, Faculty of Mechanical Engineering,
University of Belgrade

Dissertation defense date: _____

ПОДАЦИ О МЕНТОРУ И ЧЛАНОВИМА КОМИСИЈЕ

Ментор:

проф. др Раде Хајдин
Грађевински факултет, Универзитет у Београду

Чланови комисије:

проф. др Раде Хајдин
Грађевински факултет, Универзитет у Београду

проф. др Markus König
Факултет грађевинарства и инжењерства животне средине,
Универтзитет Рур, Бохум, Немачка

проф. др Милош Ковачевић
Грађевински факултет, Универзитет у Београду

доц. др Ђорђе Недељковић
Грађевински факултет, Универзитет у Београду

др Игор Светел, Иновациони центар Машинског факултета
Универзитета у Београду

Датум одбране дисертације: _____

мом деди, Драгићу Исаиловићу

to my grandfather, Dragić Isailović

Acknowledgements

First of all, I would like to express gratitude to my family for their selfless support during the entire course of my education.

I would like to thank my mentor, professor Dr. Rade Hajdin, for exceptional guidance and great support. Thanks for introducing me to the world of Building Information Modeling, and for the great balance between encouragements and criticism.

I would also like to thank my colleague and friend, Vladeta Stojanović, who helped me sort out great problems of computer graphics.

Thanks to Roads of Serbia and Dr. Nikola Tanasić, for providing the bridge inventory data and photos of bridges, without which this research would be impossible.

I owe a great appreciation to Peter Bonsma and RDF company from Sofia, Bulgaria, for providing great support and education in using their dynamic library IFCEngine.

I owe great gratitude to a selfless man and a great programmer, Lazar Banković, who helped me in debugging hundreds of lines of C++ code.

Many thanks to my colleagues from the Department of Construction Project Management, who was always full of support for my research work.

Many thanks to my friends from Ngoma Servia, for numerous joyful moments in Capoeira trainings and Rodas during the breaks from the research.

I wish to thank all of my classmates from the Faculty of Civil Engineering, my old friends from Kragujevac, and all other kind people who enriched my life with their serenity in the past four years.

Special thanks to my friend Marina, who gave me tremendous support and precious conversations.

However, the greatest gratitude I owe to my girlfriend Tamara, who has always been beside me, to share happy moments of epiphanies and acceptation of publications, as well as the moments of despair and frustration. Thank you for your endless patience and love.

Захвалница

Пре свега, хтео бих да изразим захвалност својој породици на бескрајној љубави и несебичној подршци током целог мог школовања.

Желим да се захвалим свом ментору, професору др Радету Хајдину, на изузетном менторству и великој подршци. Хвала Вам што сте ме увели у свет Информационог моделирања грађевинских објеката и хвала Вам на беспрекорној равнотежи између охрабрења и критика.

Хтео бих да захвалим и колеги и пријатељу Владети Стојановићу који ми је помогао да решим велике проблеме рачунарске графике.

Хвала Путевима Србије и др Николи Танасићу на пружању података о инвентару и фотографијама мостова, без којих би ово истраживање било немогуће.

Дугујем велику захвалност Петеру Бонсми и компанији RDF из Софије, за пружање велике подршке и обуке у коришћењу њихове динамичне библиотеке IFCEngine.

Дугујем велику захвалност Лазару Банковићу, несебичном човеку и сјајном програмеру, који ми је помогао у отклањању грешака из стотина редова C++ кода.

Велико хвала мојим колегама са Катедре за управљање пројектима у грађевинарству, који су увек били пуни подршке за мој истраживачки рад.

Хвала мојим пријатељима из Ngoma Servia, за бројне радосне тренутке на тренинзима и ходама капуере током пауза од истраживања.

Желим да се захвалим свим колегама и пријатељима са Грађевинског факултета, старим пријатељима из Крагујевца, као и свим осталим драгим људима који су својим ведрим духом обогаћивали мој живот у последње четири године.

Посебна захвалност мојој пријатељици Марини, која ми је пружила огромну подршку и драгоцене разговоре.

Ипак, највећу захвалност дугујем својој девојци Тамари, која је увек била поред мене и са којом сам поделио све срећне тренутке просветљења и прихватања публикација, као и тренутке очаја и фрустрација. Хвала ти на бескрајном стрпљењу и љубави.

Abstract

Inspection of bridges has been a standard assessment procedure for decades. Its purpose is to identify and record all defects of the bridge structure. Normally used inspection techniques are rather simple, mainly relying on visual assessment. This dissertation proposes an improvement of concrete bridge inspection in terms of visual data acquisition, damage identification and digital representation of the bridge with identified damages. Instead of depending strictly on the human eye, photogrammetrically obtained 3D point clouds are used to identify and extract concrete damage features. As the most comprehensive substitute for the old-fashioned inspection report, Bridge Information Model (BrIM) is used as an inventory and inspection data repository. An Industry Foundation Classes (IFC) semantic enrichment framework is proposed to inject the extracted and reconstructed damage features into the as-is IFC model.

After the general data model for damage description and its IFC representation are established, the method for generating the as-is IFC model of the bridge is proposed. Damage is identified as a deviation of the as-is geometry, represented by the 3D point cloud, from the as-built geometry, represented by BrIM.

Geometric and semantic enrichment of the IFC model is achieved by injecting the reconstructed 3D meshes representing damaged regions and corresponding BMS catalog-based damage information. The proposed method uses Constructive Solid Geometry (CSG) Boolean operations to geometrically enrich the IFC geometry elements, which align with corresponding damage regions from the as-is point cloud. Damage information (e.g., type, extent, and severity) is structured so that it complies to the BMS data structure.

Finally, the proposed data model, damage identification, feature extraction, and semantic enrichment method are validated in the presented case study.

Keywords: Reinforced concrete bridge, Damage, Building Information Modeling, Bridge Information Modeling, Industry Foundation Classes, 3D Point Cloud, Bridge Management System, Unmanned Aerial Vehicle

Scientific field: Civil Engineering

Scientific subfield: Information Technologies in Civil Engineering and Geodesy

UDC number: 624:005.8(043.3)

Резиме

Инспекција је већ деценијама стандардни поступак оцењивања стања моста. Њена улога је идентификовање и евидентирање свих оштећења конструкције моста. Тренутно коришћене технике инспекције су прилично једноставне и углавном се ослањају на визуелну процену. Ова дисертација предлаже побољшање инспекције армиранобетонских бетонских мостова у смислу прикупљања визуелних података, идентификације оштећења и дигиталног приказа моста са идентификованим оштећењима. Да би се избегла субјективна процена људског ока, предложено је коришћење фотограметријски добијених тродимензионалних облака тачака за идентификацију и издвајање карактеристика оштећења бетона. За складиштење података о инвентару и инспекцијама коришћен је информациони модел моста (BrIM), као најсвеобухватнија замена за старомодни инспекцијски извештај. Предложен је метод за семантичко обогаћивање основних индустријских класа (IFC) претходно издвојеним и реконструисаним информацијама о оштећењима.

Након успостављања општег модела података за опис оштећења и његове имплементације коришћењем IFC-а, предлаже се метода за генерисање IFC модела оштећеног моста. Оштећења су идентификована као одступања геометрије оштећеног моста, представљене тродимензионалним облаком тачака, од геометрије моста у нетакнутом стању, представљене BrIM-ом.

Геометријско и семантичко обогаћивање IFC модела постиже се убацивањем реконструисаних тродимензионалних троугаоних мрежа које представљају оштећења, као и особина оштећења, преузетих из каталога система за управљање мостовима (BMS). Предложени метод користи Булову операцију одузимања конструктивне стереометрије (CSG) за геометријско обогаћивања IFC модела, који се поравнавају с одговарајућим оштећеним елементима тродимензионалног облака тачака оштећеног моста. Информације о оштећењима (нпр. врста, степен и озбиљност) структуриране су по угледу на BMS.

Коначно, ефикасност предложеног модела података, методе за идентификацију оштећења и издвајање њихових особина, као и метода за семантичко обогаћивање IFC-а, потврђени су у представљеној студији случаја.

Кључне речи: армиранобетонски мост, оштећење, информациони модел грађевинског објекта, информациони модел моста, основне индустријске класе, тродимензионални облак тачака, систем управљања мостовима, беспилотне летелице

Научна област: Грађевинарство

Ужа научна област: Информационе технологије у грађевинарству и геодезији

УДК број: 624:005.8(043.3)

Table of contents

List of figures

List of tables

List of abbreviations

API	Application Programming Interface
BIM	Building Information Modeling
BMS	Bridge Management System
BPA	Ball Pivoting Algorithm
BPMN	Business Process Modeling and Notation
BRep	Boundary Representation
BrIM	Bridge Information Modeling
bSI	buildingSMART International
CNN	Convolutional Neural Network
CSG	Constructive Solid Geometry
DLL	Dynamic Link Library
DT	Digital Twin
ERD	Entity Relationship Diagram
FAA	Federal Aviation Administration
FHWA	Federal Highway Administration
GPS	Global Positioning System
GUID	Globally Unique Identifier
IAI	International Alliance for Interoperability
ICP	Iterative Closest Point
IFC	Industry Foundation Classes
IoT	Internet of Things
ISO	International Organization for Standardization
KPI	Key Performance Indicator
LiDAR	Light Detection and Ranging
LOD	Level Of Development
MFC	Microsoft Foundation Classes
NBIS	National Bridge Inspection Standard
NDT	Non-destructive Testing
NRA	National Road Authority
RANSAC	Random Sample Consensus
RoS	Roads of Serbia
SDAI	Standard Data Access Interface
SfM	Structure from Motion
SHM	Structural Health Monitoring
SIFT	Scale Invariant Feature Transforms
STEP	STandard for the Exchange of Product model data
TLS	Terrestrial Laser Scanner

TRB	Transportation Research Board
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language
XML	Extensible Markup Language

1 INTRODUCTION

1.1 Background

"Speed now illuminates reality whereas light once gave objects of the world their shape"

Paul Virilio [1]

One of the cornerstones of our global society is transportation of people and goods. By providing a wide range of social and economic benefits, the transportation infrastructure significantly improves the quality of life. As early as 1776, Smith [2] identifies the division of labor as the main drive for the improvements in the productive powers of labor. The reason for the division of labor, Smith finds in the power of exchange, whereas the limits of this division he sees in the extent of the market. In 2017, Rodrigue [3] states that the efficient transport systems implicitly help the economy by providing better accessibility to markets, employment and additional investments. Lacking capacity and/or reliability of transport systems, on the other hand, can cause a negative economic impact, i.e. decrease the quality of life. The precise estimation of the transportation impact on the macroeconomy depends on the level of observation. In case of advanced economies, Rodrigue [3] roughly estimates this impact somewhere between 6% and 12% of the GDP. However, the transportation's role in the society goes far beyond the economics. It literally shapes the human perception of space and time by providing a flawless mobility of people and goods. A cultural theorist, urbanist, and aesthetic philosopher Paul Virilio claimed that time and space are compressing due to the shortening distances and travel times [4]. On the micro scale, however, transportation is a default mean by which an average person is able to participate in the social, cultural and political activities.

Once realized as manifold beneficial, the road infrastructure must be responsibly and efficiently maintained to ensure the exploitation of these benefits in the future. However, the explanation of the importance of transportation for society and the economy should be followed by the analysis of the used transport modes distribution. Transportation modal split provided by Eurostat ([5] and [6]), shows that more than 90% of inland passenger transport and more than 75% of inland freight transport in EU in 2015 is carried out using road infrastructure. The World Road Association (PIARC) report about the importance of road maintenance [7] states that the added value of commercial road transport to GDP lies somewhere between 3% and 5%.

Providing a safe and fast passage across the valley, river or other kind of obstacle, bridges are considered as critical components of the road infrastructure. The need for bridge management, or taking care of bridges, is recognized centuries ago. Even in Ancient Rome, the builder was responsible for the structural integrity of the bridge for 40 years since commission [8]. Ever since, systems and procedures for bridge management have been permanently improving. However, tragic events, such as Morandi bridge collapse in Genoa

in 2018 (Figure 1.1), painfully demonstrated some flaws of the current bridge management practice.

Figure 1.1: Collapse of Morandi bridge in Genoa, on August 14th 2018 (retrieved from [9])

1.1.1 Bridge Management Systems

Bridge management is defined by Lauridsen et al. [10] as a set of certain procedures with the purpose of ensuring the traffic safety and maintaining the bridge stock in the desired condition at the lowest possible cost. Road bridges are usually governmentally owned. Thus, they are managed by National Road Authorities (NRAs). Number of bridges in the average national infrastructure network is measured in tens of thousands. Moreover, bridges vary in structural type, number and length of spans, construction year, etc. Therefore, the need for a specialized highly sophisticated information system to deal with this large stock of data is recognized in the late 1980s. Currently, the majority of NRAs in the world use information systems, named Bridge Management Systems (BMSs), as a support for managing their bridges. As described in [10], a BMS assists in management of activities such as inventory and inspection data collection, bridge condition assessment, heavy transport administration and allocation of funds for maintenance interventions. Although there are a variety of different BMS solutions on the market, a widely adopted modular BMS structure will be further discussed. According to Ryall [8], common BMS includes following modules:

- Inventory module

- Inspection module
- Maintenance module
- Financial module
- Bridge Condition module

Figure 1.2 shows the common BMS structure. Whereas all the listed modules use the data stored in BMS database, the Inspection module can alter the bridge data according to the inspection findings and Maintenance module can even introduce new data corresponding to the repaired, replaced, and/or added component.

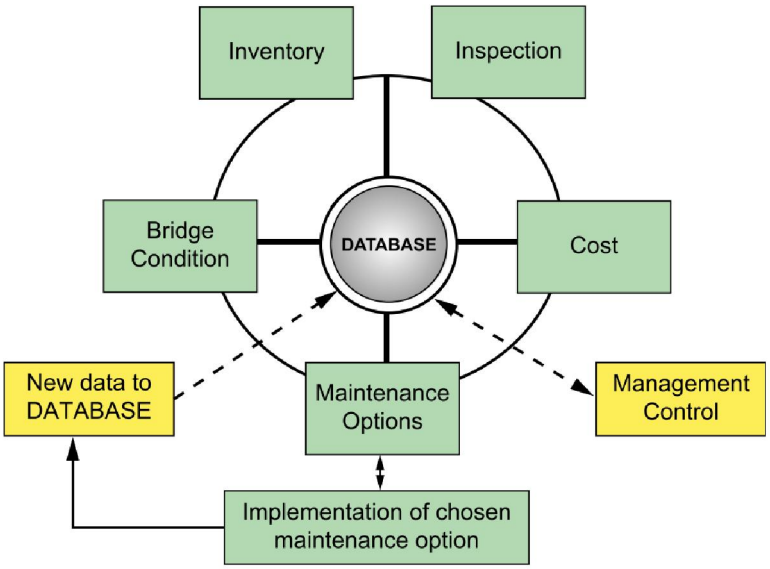
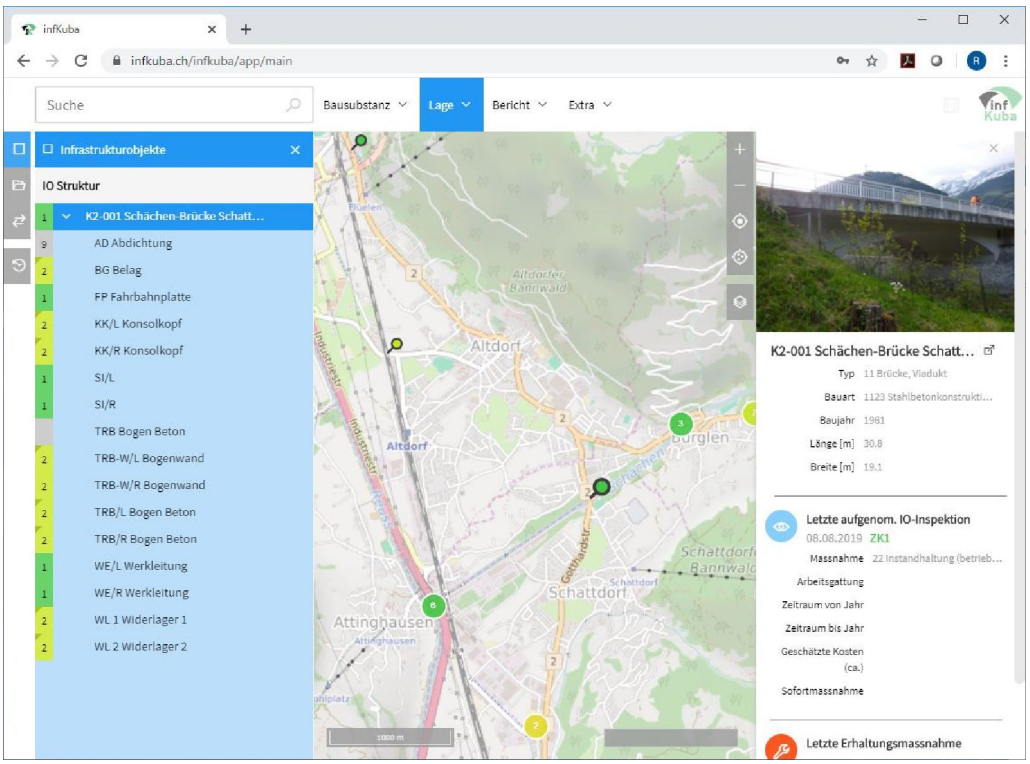


Figure 1.2. Common BMS structure (retrieved from [8]).

Inventory module contains administrative and technical bridge data, such as bridge ID, road ID, geolocation, bridge type, number and length of spans, dimensions, materials, etc. This module can optionally include the photo of the bridge. Screenshots of the Inventory component of the Swiss BMS, named KUBA are shown in Figure 1.3.

(a)



(b)

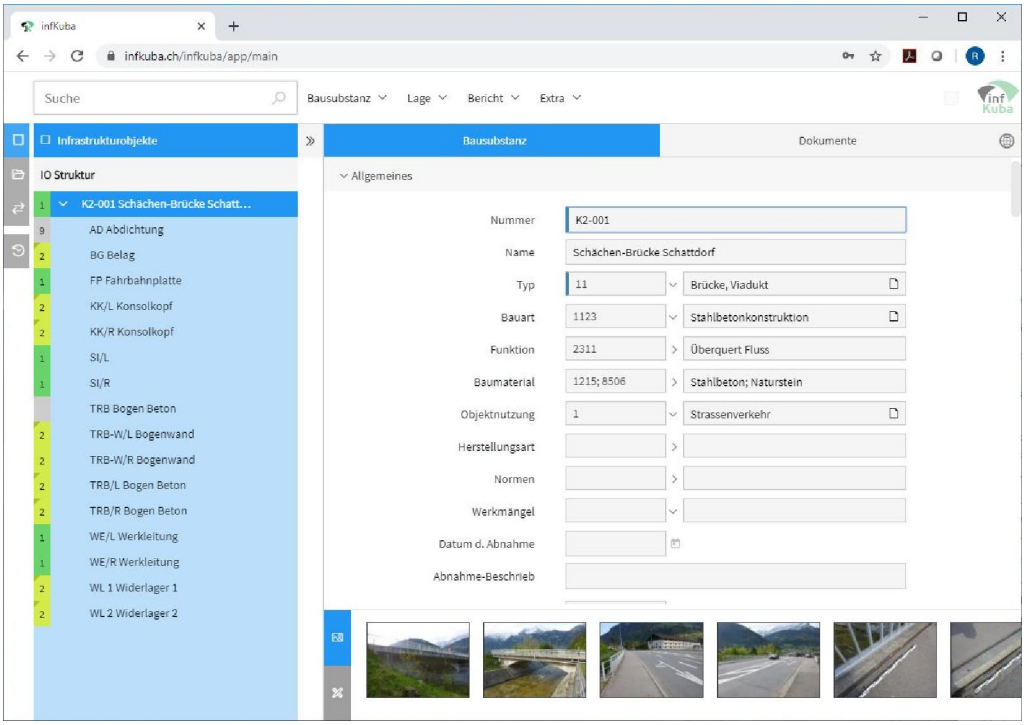


Figure 1.3. KUBA BMS Inventory module dialog box: (a) bridge identification data, (b) bridge description.

Maintenance of bridges is performed through periodical inspections. Inspection procedure is implemented differently from country to country. However, the basic approach is common all around the world. For instance, Serbian regulations for maintenance of bridges [11] stipulate four different types of inspection: control (performed twice a year), regular (performed once every two years), principal (performed once every six years) and emergency inspection (performed in special cases e.g. natural hazards). Inspector treats a

bridge as an assembly of structural elements (e.g. deck, piers, abutments) and non-structural elements (e.g. safety rail, pavement, drainage system). After examination of each element, searching for any defect in a bridge structure, inspector fills the inspection report that includes element ratings. Rating system consists of six ordinal categories, among which the appropriate one is chosen by an inspector to describe the element condition. Report is usually paper-based, and its content is automatically generated based on the inspection data from a BMS. Such a report, generated by KUBA BMS, is shown in Figure 1.4. The report consists of alphanumeric fields, mainly with predefined values. In addition to the mandatory condition ratings of elements and the entire bridge, photos of detected defects with textual description can be added to the database.

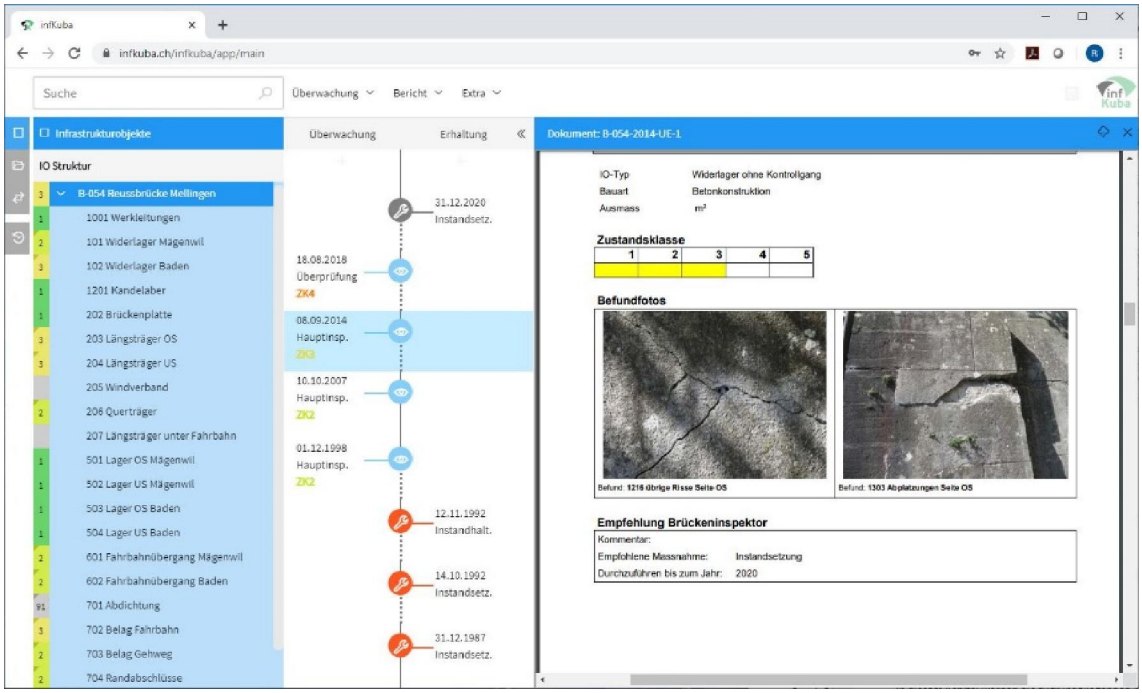


Figure 1.4. KUBA BMS Inspection module - automatically generated inspection report.

The purpose of maintenance module is to support maintenance planning and scheduling. As described by Hawk [12], maintenance module records a historical log of all performed maintenance interventions for each bridge in the network with regard to the type and extent of the works, as well as the cost. Based on the exhaustive maintenance cost record, the module is able to estimate the future maintenance actions cost. Finally, this module is used to schedule maintenance actions.

Financial module performs the most important task in BMS: the allocation of funds. Of course, similar to other modules, its output is more a proposal than the obligation. The final decision is on the NRA. Based on the inventory and inspection data (i.e. condition ratings), as well as the assigned set of constraints and available budget, the maintenance funds are optimally allocated. Hawk [12] defines constraints as the required performance goals of a certain or entire bridge stock in a determined period of time (e.g. 10 or 20 years).

To be able to allocate the available funds in an optimal way, it is necessary to predict the future bridge condition in case the maintenance actions are not performed. This is the job of Bridge condition module. According to Zambon et. al. [13], this module includes models for bridge condition prediction. Prediction models can be either deterministic (e.g. one-dimensional regression analysis) or probabilistic (e.g. failure rate model or Markov chains). Whether deterministic or probabilistic, the model considers the current condition as well as the occurred deterioration process to predict the future bridge condition.

1.1.2 Bridge representation in Bridge Management Systems

As shown in Figure 1.2, BMS data repository is mostly a relational database. This means that every piece of information is represented by an alphanumeric value of a table cell. Whereas table columns hold categories of the data (*attributes*), each row in the table (*record* or *tuple*), contains a unique identifier (*key*) for every category defined in columns. Connections between tables are implemented via the used of so-called "*primary – foreign key*" relationships. Namely, each table has a field called a *foreign key*, which is identical to *primary key* of the table, whose record is attributed to the record of the foreign key.

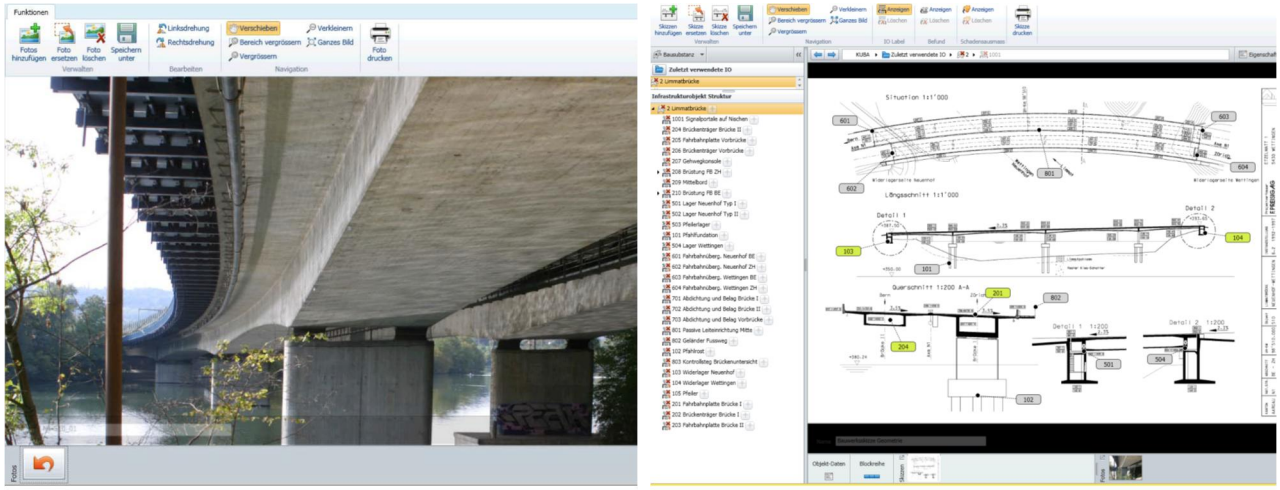
Rigidly structured and clearly defined, the information from database is easily used by the Maintenance, Financial and Bridge Condition modules to generate various analytics. However, the following problems occur when inspection findings are to be inserted into database:

- Imprecise descriptive positioning of inspection findings
- Loss of inspection finding information caused by fitting to the limited set of predefined damage types and discrete values of damage extent and severity
- Inspector's free interpretation of the damage severity

From the data model point of view, bridge is usually represented by an element assembly (i.e. mutually related tables representing each bridge element). Having said that, the inspector needs to put a tremendous effort to imagine the bridge he or she stands in front of as a database table. Moreover, in case of any detected damage, the inspector needs to find a way to adjust the inspection finding to fit the table attribute format. This means that the detected damage type, severity and extent need to be inserted into database as predefined types and discrete values. This leads to the loss of valuable information about the detected damage by fitting it into the required format. Furthermore, the crucial information about damage location is not even included in the table. Combined with the extent and type, damage location is interpreted by the inspector and recorded implicitly as a condition rating of a damaged bridge element.

All the listed issues indicate the need to introduce the bridge geometry to BMS. As shown in Figure 1.5, so far, some systems such as the Swiss Federal Roads Authority BMS named

KUBA [14], enabled the attachment of photos in JPG, BMP, GIF, PNG or TIF file format.



(a) (b)

Figure 1.5. Swiss BMS KUBA: Bridge geometry data as attached files in the BMS database. (a) Photo, (b) Captured drawing of the bridge (retrieved from [15]).

This kind of visual information improves the spatial perception of the bridge and its elements. In the KUBA BMS, the inspection findings can be inserted into the bridge drawing by dragging the damage icon to the appropriate location (Figure 1.6). In the upper part of the BMS screenshot, shown in (Figure 1.6), the bridge segments with their damage extent are listed. Although much more intuitive, this way to record inspection findings is rather symbolic. Comparing to the direct database input, the only value this functionality adds to BMS is a rough damage location.

A significant effort was made to improve the spatial perception of the bridge further than added photos. Namely, some BMSs such as KUBA enabled the attachment of digital 3D models of bridges in 3DS file format (Figure 1.7). While the navigation functionality of the embedded model viewer undoubtedly improved the visual experience of the bridge, nothing changed in terms of data model. 3D model of the bridge remained the independent entity, incapable of any data exchange with the BMS database.

According to Mirzaei et al. [16], none of the existing BMSs include geometric representation of bridges.

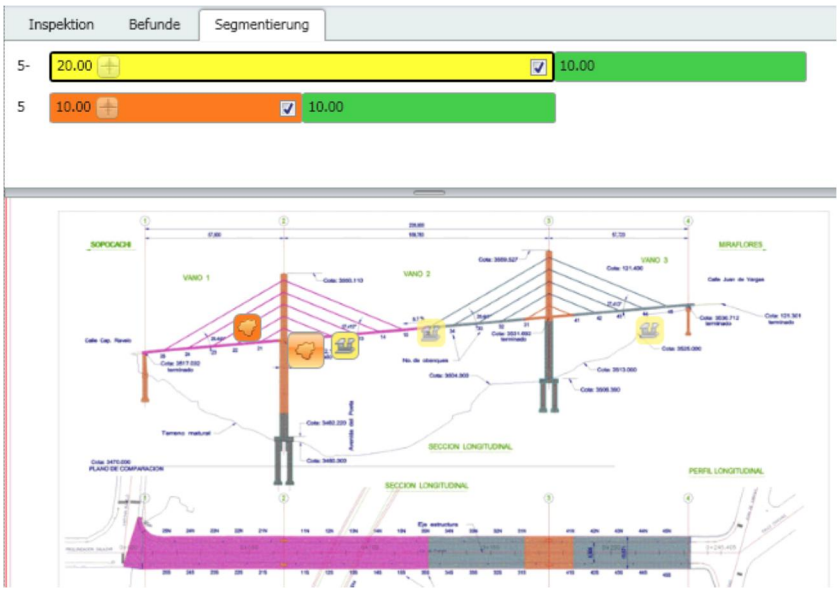


Figure 1.6. Swiss BMS KUBA: Inspection findings icon representation on the lower side and damage extent list on the upper side (retrieved from [15]).

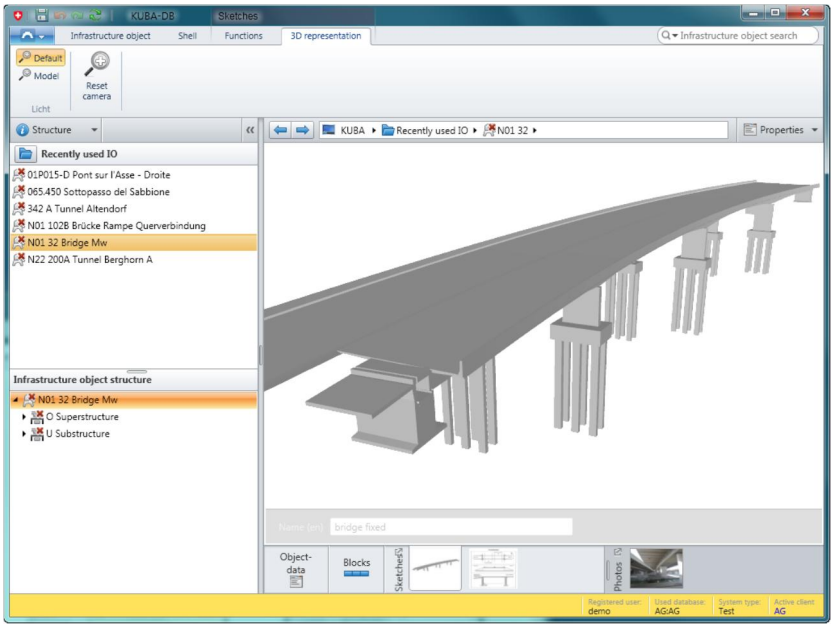


Figure 1.7. Swiss BMS KUBA: Digital 3D model of the bridge (retrieved from [15]).

1.1.3 Building Information Modeling

Recognizing the need for more efficient way to manage construction projects, minimizing the information loss between the project stages, back in 1974, Eastman et al. published the report named "An Outline of the Building Description System" [17]. Few decades later, based on this idea, a new technology named Building Information Modeling (BIM) is developed. In 2007, the United States National Institute for Building Sciences [18] defined BIM as "an improved planning, design, construction, operation, and maintenance process using a standardized machine-readable information model for each facility, new or old, which contains all appropriate information created or gathered about that facility in

a format useable by all throughout its lifecycle". Previously analyzed and disassembled into simple components, common processes and information related to the planning, design, construction and operation phases of the building are digitalized by insertion into BIM. BIM itself is an object-oriented model of a building, capable of complex modifications, highly precise analysis, automatic generation of drawings and reports, as well as various 3D visualizations.

Having said that BIM is actually a set of virtual interrelated processes, it certainly implies the existence of a digital model of the object. It is an exhaustive repository of various information about the building. BIM includes information from multiple domains (i.e. structural, architectural, mechanical, electrical, etc.) and it is structured in an object-oriented manner. This means that each building element is represented by an object, which is an instance of a specific object class, defined by certain properties (i.e. attributes). Besides building elements, an object classes representing various types of relationships between them are also defined. The feature by which BIM is best known is the 3D visual representation of each building element. The accuracy of the building description in the BIM can vary from purely conceptional to highly detailed and it is defined by the Level-Of-Detail (LOD) [19]. Figure 1.8 shows the BIM detail of an interior masonry wall, LOD 400. At this LOD, BIM includes information about reinforcing, connections, grouting material, jams, bond beams, lintels, member fabrication part number, and any part required for complete installation [19].

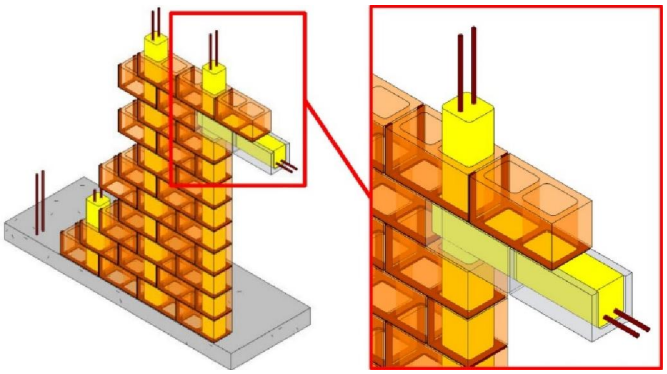


Figure 1.8. BIM detail: interior masonry wall at LOD 400 (retrieved from [19]).

1.1.2 Hypothetical utilization of Bridge Information Models (BrIMs) by BMS

Although few decades old [20], only the recent government mandates promoted BIM in the construction industry as the most efficient technology to create, store and modify data for the built environment throughout its entire lifetime. Started in 2007 in Norway, Denmark and Finland, followed by the USA in 2008 and South Korea in 2010, public organizations in more than 15 countries around the world announced their plans to mandate BIM. Among the several benefits of the government BIM mandate noticed by Sacks et al. [21], the most interesting for this research is certainly the impact on public

construction projects. Namely, a great portion of construction projects, especially the infrastructure ones are publicly financed. This means that in a foreseeable future every newly constructed infrastructure asset will have a corresponding BIM representation. Furthermore, it is reasonable to expect the government BIM mandate for maintenance as well, which will eventually result in a comprehensive collection of Bridge Information Models (BrIMs) for the entire bridge inventory.

If utilized by BMS, BrIMs could tremendously improve both the quantity and quality of information in BMS. Contrary to the current practice of ignoring or losing a huge amount of data about the bridge, produced during the design phase, BrIM would store it and, if needed, provide it to BMS. Information about bridge geometry, structure, materials, or loads, stored in an object-oriented model, provides an accurate inventory representation, which is beyond the alphanumeric representation in current bridge databases. Figure 1.9 shows the BrIM detail of a prestressed concrete box girder bridge.

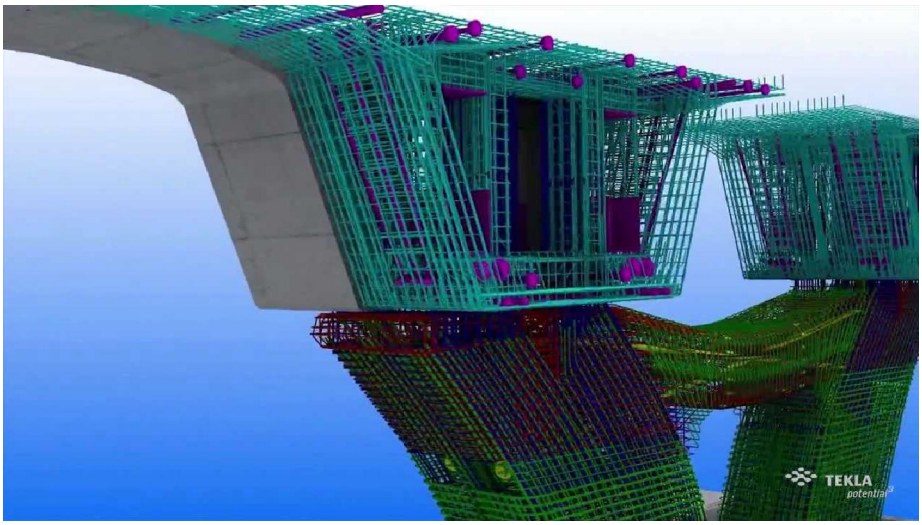


Figure 1.9. BrIM detail: prestressed concrete box girder bridge segment (model by Tekla Structures [22]).

Although potentially highly beneficial, the utilization of BrIMs by BMS is currently impossible for two reasons. The first reason is the lack of relationship between the BrIM and BMS database. As described in chapter 1.1.2, the bridge is represented in BMS by a relational database, whereas BrIM is an object-oriented model. The second reason is the lack of BrIM framework for modeling of inspection findings. Whereas commercial BIM software is capable of creating 3D bridge models with highly accurate geometry, there is no software solution or even a universal guideline for modeling damages of other findings. Moreover, an existing BrIM data model lacks object classes for damage description.

To overcome differences in existing data models, the new one needs to be developed, to include the currently used semantics in BMS and BrIM. To enable the efficient use of all the information contained in BrIM, the newly established data model needs to include a much more detailed semantics for a description of inspection findings than the one currently used in BMS.

1.2 Research objectives

The general objective of the research in the scope of this dissertation is to enable the utilization of BrIM by BMS. The specific objectives are summarized below:

1) Development of the geometric and semantic data model for description of damages of reinforced concrete bridges, compliant with BMS requirements.
2) Development of the process model for the geometric and semantic enrichment of BIM models with inspection findings
3) Quantification of efficiency of the developed data model and process model
4) Formulation of recommendations for utilization of BIM models by BMS

Objectives of this research have been implemented through the following tasks:

1) Analysis of the current inspection procedures
2) Analysis of the content and structure of the data in the existing BMSs
3) Testing the usability of remote sensing technologies in bridge inspection, regarding the spatial approachability and accuracy
4) Analysis of the remote sensing output data formats
5) Analysis of the Industry Foundation Classes (IFC) specification
6) Establishment of the geometric and semantic data model for description of damages of reinforced concrete bridges, based on the previous analysis
7) Modeling the process for the semantic and geometric enrichment of BIM models with inspection findings
8) Learning the programming tools and techniques for the BIM software development
9) Development of the prototypical BIM software for the semantic and geometric enrichment of BIM models with inspection findings
10) Quantification of efficiency of the developed data model and process model in the case study
11) Formulation of recommendations for utilization of BIM models by BMS

1.3 Research methodology

The research in this dissertation is conducted by means of:

- Analysis
  - of the literature and state of the art in the relevant scientific and engineering fields,
  - of currently applied data models in legacy Bridge Management Systems,
  - of capabilities of the available data acquisition technologies,

  - of the available standards for building data model representation,
- Experimentation
  - by testing the available commercial software applications,
  - by bridge surveying on field,
  - by testing the developed data model in the implementation of the proposed method,

and

- Synthesis
  - of the findings based on the case study results.

1.4 Dissertation outline

This dissertation is organized in seven chapters.

Chapter 1 introduces the reader with the general background and motivation for the research, by description of Bridge Management Systems, Building Information Modeling, and possibilities and obstacles for enabling the synergy of those two systems. Afterwards, the research objectives and methodology are presented.

Chapter 2 provides the review of literature addressing systematization of common bridge damages, current bridge inspection procedure, some novel technologies applied in bridge inspection and condition assessment, bridge point clouds, Industry Foundation Classes, and semantic enrichment of IFC models.

Chapter 3 thoroughly describes the establishment of data model for the as-damaged bridge by introducing the data modeling concepts and notation, analyzing the current structure of inspection data in BMS, and selecting the essential concepts from the existing data model. Afterward, the implementation of the established data model by using Industry Foundation Classes is proposed.

Chapter 4 presents the method, i.e. procedure for generating the as-is Bridge Information Model by implementing the data model proposed in Chapter 3. The process is modeled by means of Business Process Modeling and Notation standard, after which the entire process is described step by step.

Chapter 5 presents the practical evaluation of the proposed data model, Industry Foundation Classes implementation, and the method for generation of as-is Bridge Information Model. An inspection case study has been performed on the relatively short reinforced concrete roadway Bridge over River Gročica.

Chapter 6 discusses various aspects of the proposed data model by analyzing results of the case study. Discussed aspects include the representation of damage semantics in the

as-is Bridge Information Model, automation of damage detection in context of the proposed method, benefits for the BMS resulting from the possible implementation of the proposed method, economic viability of the method, and current legal limitations for the broad use of this method.

Chapter 7 concludes the dissertation by pointing the advantages and disadvantages of the proposed data model and method, based on the results of the case study. Finally, the recommendations for future research, naturally resulting from the one presented in the dissertation are provided.

2 LITERATURE REVIEW

2.1 Damage on reinforce concrete bridges

Aging, aggressive environment, exploitation and neglect are causes for severe damages all the infrastructure assets are facing. According to Transportation Research Board of the U.S National Research Council (TRB) [23], reinforced concrete bridges deteriorate mainly due to deicing salts, overloading, freeze–thaw cycle induced stresses, fatigue, and corrosion of rebar. The most severe deterioration process is reinforcement corrosion, decreasing the rebar cross-section and inducing stresses which lead to the concrete delamination and spalling experience. In the following section, both corrosion-induced and other, less frequent damages, will be described. Further on, the damage susceptible parts of reinforced concrete bridges will be discussed.

2.1.1 Common damage types

Number of researches, such as the report of the Working Group 1 of COST Action TU1406 [24], or the one by Bień et al. [25] provide very detailed classifications of bridge defects, addressing degradation mechanisms and processes, as well as it's manifestations. According to [24], the following types of damages affect concrete bridges:

- Abrasion
- Cavities
- Corrosion
- Cracks
- Delamination
- Deflection
- Fatigue
- Insufficient concrete cover
- Insufficient concrete quality
- Scour
- Spalling

TRB emphasize reinforcement corrosion, concrete delamination and spalling, vertical cracking, and concrete degradation as dominant defects reinforced concrete bridges experience [23]. However, there is a wide consensus that the main cause of the bridge deterioration is the corrosion of reinforcement steel [26].

Besides carrying the compression stresses, concrete serves as a corrosion protection for the reinforcement steel in reinforced concrete structures. This protection is not just physical, but chemical too. The high alkalinity of concrete causes a formation of a thin, anti-corrosive, passive oxide layer on the rebar surface. Once the aggressive elements (i.e. water, de-icing chemicals, etc.) penetrate inside the concrete, due to the high permeability and/or cracks, and come in contact with the passive film and destroy it, the reinforcement

is not protected anymore and the corrosion starts. It is the electrochemical reaction, such as the one happening in a battery. According to Ball and Whitmore [27], for a corrosion to occur, four elements are required: an anode, a cathode, ionic continuity between anode and cathode (i.e. electrolyte), and a metallic connection between the anode and cathode. Whereas some areas along the bar act as anodes, others act as cathodes. The metallic connection between the anode and the cathode is provided by any bar in contact with both of them (e.g. stirrups can be a metallic connection for two parallel reinforcement bars, playing the anode and the cathode). The wet concrete acts as an ionic medium i.e. electrolyte.

Ball and Whitmore [27] claim that the destruction of the passive oxide film (i.e. the corrosion initiation) is usually caused by the introduction of chlorides to the rebar, either by penetrating water, de-icing salt, or a contaminated aggregate. Additionally, the oxide film can be destroyed by the loss of alkalinity around rebar (i.e. drop of the pH value bellow 7[1]), due to a carbonation. Carbonation is a reaction between carbon dioxide from the atmosphere and calcium hydroxide from the cement paste, in the presence of water. As a result, the calcium hydroxide reacts to calcium carbonate with the approximate pH value of 8.5.

Reinforcement corrosion has a twofold effect on reinforced concrete structure. Firstly, the rebar cross section is reduced (sometimes the bar is even ruptured), thus the structural capacity of an entire reinforced concrete element is reduced as well. Secondly, the corrosion induces the internal stresses, which cause concrete cracking, delamination, and eventually spalling (Figure 2.1).

Concrete discontinuities (i.e. cracks) can result from deterioration processes unrelated with the reinforcement corrosion. Factors such as plastic shrinkage, hydration heat, changes in environment temperature, geometric constraints during the concrete curing, traffic load, freeze-thaw cycles, alkali–silica reaction, and delayed ettringite formation can cause cracks as well. Moreover, each crack by exposing reinforcement speed up the corrosion process and induces more damage [23].

[1] According to Ball and Whitmore [27], even a drop of the pH value bellow 10 leaves the chloride ion free concrete unprotected from corrosion.

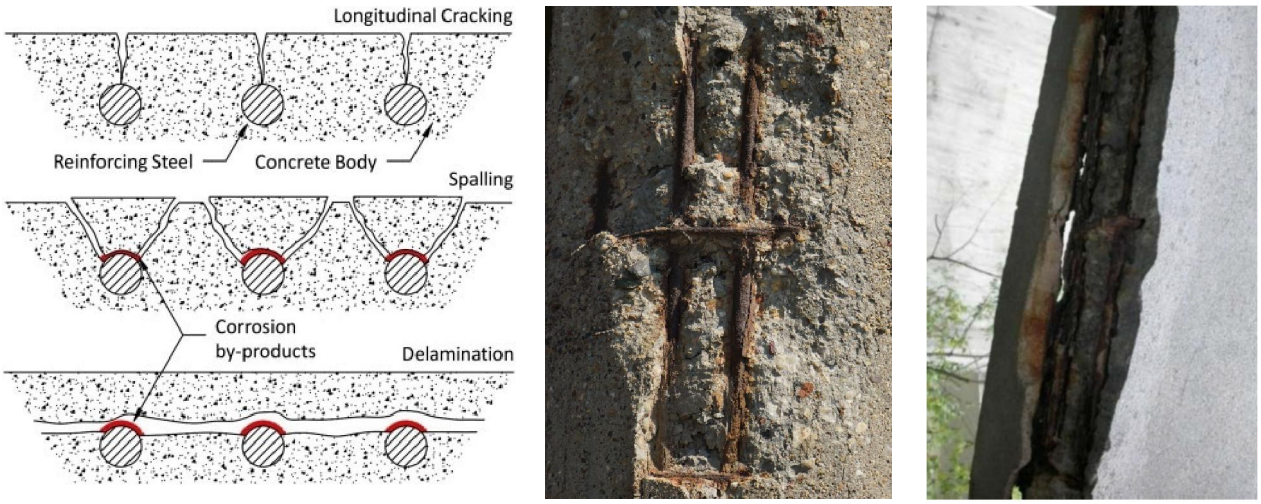


Figure 2.1. (a) Mechanism of corrosion-induced damage (retrieved from [28]), (b) Spalling, (c) Delamination.

2.1.2 Bridge parts susceptible to damage

In their comprehensive "Classification of damage in concrete bridges", Zakić et al. [29] pointed some bridge parts susceptible to damage. They indicate the vulnerability of bridge parts or areas where big displacement or dislocation may occur. This group include expansion joints, joints between elements, and bridge ends. Figure 2.2 shows damage on the column head, ends of precast girders, and expansion joint.

Figure 2.2. Damage due to the water penetration between elements: (a) Spalled ends of deck elements in the joint, (b) Detached pavement from the dilatation mechanism.

The second damage susceptible group includes almost horizontal parts under high dynamic load, such as pavement, sidewalks, and rails. Figure 2.3 shows the damaged pavement and sidewalk.

The third group includes elements in contact or right under the drainage installations. Improperly designed and/or poorly maintained drainage system misdirects water from the roadway to the bridge structural elements (Figure 2.4).

Not only the bridge superstructure suffers from damage, but a substructure too. According to Steinkamp [30], bearing areas on caps, piers and abutments deteriorate due to the saline water penetration through the joints from above (Figure 2.5 and Figure 2.6).

(a) (b)

Figure 2.3. Damage of mainly horizontal elements under high dynamic load: (a) Pavement hole, (b) Completely deteriorated sidewalk structure.

(a) (b)

Figure 2.4. Damage induced by degraded drainage pipe: concrete deterioration in its (a) initial and (b) advanced phase.

(a) (b)

Figure 2.5. Damage of bearing area: (a) Corroded steel and demolished rubber of the bearing, (b) Large spalling on the abutment and the girder end.

(a) (b)

Figure 2.6. Damaged piers: (a) Cracked pier cap, (b) large spalling on the pier.

2.2 Current bridge inspection procedure

2.2.1 Inspection types and frequencies

Currently, bridges are inspected mostly manually. Although each country stipulates regulations for the bridge inspection, basic principles are common all around the world. Serbian regulations [11] stipulate following four types of inspection: Control (performed twice a year), Regular (performed once every two years), Principal (performed once every six years), and Emergency inspection (performed in special cases). Whereas the Regular inspection is rather superficial, the rest tree types of inspection examine a bridge in detail. The core examination method, used in any inspection, is visual observation. Besides this method, in Principal and Emergency inspection a specialized equipment for indirect condition assessment is utilized. Regular inspection, on the other hand, relies purely on visual observation. Conducted once every two years, this inspection is critical for the identification of any kind of deterioration. Besides these four types, some NRAs perform the additional inspections. As described by Lindbladh [31], Regular Inspection in Sweden focuses on the condition of pavement and road embankments and it is done in conjunction with the road inspection. The most frequent bridge inspection in Sweden, named "Superficial", is independent from the road inspection. It is performed twice a year. The US Federal Highway Administration (FHWA) [32] stipulates the Fracture Critical Inspection, performed in conjunction with the Routine Inspection. The aim of this inspection is a close-up investigation of fracture critical members (i.e. bridge member in tension, whose failure can cause the collapse of a part of a structure or entire bridge). FHWA also requires Element Level Bridge Inspection, In-Depth Inspection, and Post-Earthquake Inspection, all of which are self-explanatory. According to Kentucky Bridge Inspection Procedures Manual [32], The FHWA also require the establishment of a guideline for inspecting every complex bridge (i.e. cable-stayed, suspension, or other bridge

with unusual characteristics). Inspection regulations in the rest of the world are similar to the described ones, although names and frequencies of particular inspection types slightly vary. Table 2.1 shows the inspection types and frequencies of NRAs from eleven countries. Table 2.2 shows the depth or intensity of certain types of inspections in different countries. Namely, it is shown that short-interval inspections are rather cursory, whereas the more frequent inspections require the visual examination of each bridge element, or even an in-depth investigation using sophisticated testing equipment. Both tables are originally retrieved from the report by Hearn [33], and enhanced with data from Serbia and Switzerland.

Today, a number of non-destructive-testing (NDT) methods are available for the in-depth investigation of the bridge structural elements. In 2013, TRB [23] evaluated the common NDT methods for identifying concrete bridge deck deterioration. Highly sophisticated technologies, such as impact echo, ultrasonic pulse echo, ultrasonic surface waves, impulse response, ground-penetrating radar, microwave moisture technique, Eddy current, half-cell potential, galvanostatic pulse measurement, electrical resistivity, infrared thermography, and chloride concentration measurement, are state of the art methods, capable of detecting the inner defects in structural elements. Even less sophisticated methods, such as chain dragging, hammer sounding, and visual inspection, can indicate the presence of non-visible, subsurface concrete defects. Among all the listed NDT methods, visual inspection is certainly the most important. Even by detecting visible damage only, it can indicate the surface symptoms of inner concrete deterioration processes. Being the most frequent, visual inspection triggers the less frequent, but more detail in-depth investigation.

Table 2.1. Bridge Inspections around the world (adapted from [33]).

Inspection Interval	Serbia	Switzerland	U.S.	Denmark	Finland	France	Germany	Norway	South Africa	Sweden	United Kingdom
3 months							Superficial				Superficial
6 months	Control										
1 year				Routine	Annual	Annual		General	Monitoring	Superficial	
2 years	Regular		Routine								General
3 years						IQOA*	Minor			General	
4 years			Routine 48-month								
5 years		Principal			General 5-year			Major	Principal		
6 years	Principal			Principal		Detailed	Major			Major	Principal
7 years											
8 years					General 8-year						
10 years			In-depth 12-month								
For project	Emergency	Intermediate **, Special	Special	Economic Special	Special		Special	Special	Project-level	Special	Special

*** IQOA = IMAGE DE LA QUALITÉ DES OUVRAGES D'ART.**
**** Swiss Intermediate inspection is executed approximately twice as often than Principal, focusing on the key elements of the bridge.**

Table 2.2. Intensity of Bridge Inspections around the world (adapted from [33]).

Inspection Access	Serbia	Switzerland	U.S.	Denmark	Finland	France	Germany	Norway	South Africa	Sweden	United Kingdom
Drive-By	Control			Daily		Routine				Routine	
Visible	Regular	Main	Routine Routine 48-month	Routine Principal		Annual IQOA*	Minor	General	Monitoring	Superficial General	Superficial General
Arm's Length	Principal	Principal Intermediate	In-depth 12-month		General 5-year*		Major	Major	Principal	Major	Principal**

** SAID TO BE "ARM'S-LENGTH," BUT TRAFFIC LANE CLOSURES ARE RARELY PROVIDED.
* IQOA = IMAGE DE LA QUALITÉ DES OUVRAGES D'ART.

2.2.2 Visual inspection

In spite of the established frameworks for in-depth examination, using highly sophisticated NDT equipment, the core bridge inspection technique is undoubtedly the visual inspection. The reasons for this are numerous. Firstly, the in-depth examination of each bridge in its entirety is not economically feasible. Secondly, such an approach would result in huge amounts of hardly interpretable data. Therefore, it would require a human labor and powerful computers to process the data, thus substantially increasing the inspection cost. The last, but not least, the input data requirements of the current BMSs are rather rigid, i.e. interpretation of inspection findings is necessary prior to insertion into BMS. This is why the current inspection practice entails the visual routine inspection. Moreover, all types of inspections are mainly visual, whereas the NDT equipment is used rather as an aid for the confirmation of the suspicions related with possible bridge deterioration processes. Visual examination implies the interpretation of the findings prior to filling the report or rating the bridge element or the entire bridge condition. Nevertheless, this so much appreciated interpretation of findings is the visual inspection's great strength, but a great weakness as well. Namely, the inspector's subjectivity is unavoidable.

In 2001, Moore et al. [34] conducted a research analyzing the reliability of visual inspection of highway bridges. They analyzed two generalized types of visual inspection: routine and in-depth. The inspections were performed by 49 inspectors at seven bridges from south central Pennsylvania and northern Virginia. One of the main goals of the research was to find the correlation between the assessed condition rating accuracy and various inspectors attributes, such as mental and physical condition, fears of hazards related to inspection, and experience level. The results showed a significant variability of the assessed condition ratings. Only 68% of assigned condition ratings were within 1 point of the average condition rating. Given that the U.S. National Bridge Inspection Standard (NBIS) [35] provides a ten-grades condition rating system (from 9 (excellent) to 0 (failed)), and that the low and high ratings are not used frequently, effectively reducing the system to only 6 grades, the presented results indicated a great subjectivity in rating the bridge condition.

Using the multivariate nonlinear regression analysis, Moore et al. [34] managed to determine the following human and environmental factors influencing the condition rating:

- Near and far visual acuity
- Color vision
- Fear of traffic
- Fear of height
- Formal education in bridge inspection
- Experience in bridge inspection
- Level of structure maintenance
- Accessibility and complexity of the structure
- Time available for performing the inspection
- Applied tools
- Light intensity
- Wind speed

The report by Moore et al. [34] indicated some subjectivity-based differences in inspection reporting as well. Each damage has been documented in a form of a written notation by between 60% and 80% of the inspectors, but none damage is noted by more than 88% of the inspectors. Furthermore, the inspectors were providing photos of the most severe and easily accessible damages, rather than the less severe and unapproachable ones.

2.3 New data acquisition technologies for bridge condition assessment

The first and the most important step of making the entire condition assessment process digital, is the digitalization of data acquisition. Here, the term "digitalization" implies some extent of automation of the acquisition process. In the past few decades, a variety of data acquisition technologies emerged. Two approaches of data collection are established, and they can be roughly described as "internal" and "external". These approaches are: Structural Health Monitoring (SHM) and Remote sensing, respectively.

2.3.1 Structural Health Monitoring (SHM)

According to Farrar et al. [36], SHM can be defined as "the integration of sensing and possibly also actuation devices to allow the loading and damaging conditions of a structure to be recorded, analyzed, localized, and predicted in a way that NDT becomes an integral part of the structure and a material". Supporting the claim that SHM is the technology that tracks the structure condition from within, it implies the system of embedded sensors, which are meaningfully positioned to perform certain measurements indicating the

structure condition. According to Chen [37], various types of sensors are used to assess the bridge condition. These sensors can be ceramic, oxide-based, electromagnetic, microelectro mechanical, fibreoptic, wireless, and thin/thick film-based. They can be used to detect, evaluate and quantify performace indicators of the bridge. According to the Working group 1 of the COST Action TU1406 [24], the SHM is still in the research phase. Namely, questions about properties of performance indicators, such as type, mathematical formulation, threshold, etc. still need to be answered.

As opposed to the SHM, remote sensing is a contactless technology. The information about the observed object is acquired by a device that is not in contact with the object. According to Lillesand et al. [38], this technology is based on the utilization of electromagnetic sensors for recording variations in the way observed object reflects and emits electromagnetic energy. There are several types of remote sensing, depending on the source of electromagnetic energy. Perhaps, the most widely applied is the combination of digital photography and photogrammetry. As described in [38], digital photographs of an object, captured whether manually or by specialized Unmanned Aerial Vehicle (UAV) equipped with digital camera, are processed and interpreted in 2D or 3D manner. Besides the combination of digital photography and photogrammetry, there are also multispectral, thermal, hyperspectral, microwave, and Lidar sensing, as well as remote sensing from space satellites.

Due to the high cost and high weight of the early equipment, remote sensing has been used mainly for the large-scale land mapping, either from the airplane or from the ground. With the advance in the information technology and data acquisition, remote sensing and availability, remote sensing gained various applications in geology, agriculture, forestry, urban planning, etc. The most widely used are UAV-based photogrammetry and terrestrial laser scanning. Recently, the application of remote sensing in bridge inspection became the topic of numerous researches. The most significant obstacle for the full application of this technology in bridge inspection and condition assessment is the lack of guidelines for interpretation of remote sensing outputs (i.e. various imagery and 3D point clouds). Therefore, this application is currently limited to the manual inspection of 3D point clouds.

2.4 Remote Sensing utilization in Bridge Inspection

2.4.1 Point Clouds

3D point cloud is a set of data points, defined by x, z, and z coordinates. Additionally, in case of photogrammetric point clouds, point RGB color is provided, whereas the laser-based point clouds provide only the point reflectance intensity.

The most common and certainly the most reliable point cloud generating technology is terrestrial laser scanning (TLS). By measuring the time of flight for a given azimuth and elevation, TLS collects range (x, y, z) and intensity (reflectivity) of the points in the scene, generating a 3D point cloud. Although highly accurate, the heavy weight and fixed position makes TLS inconvenient for capturing the unapproachable bridge segments. Essentially the same technology, Light Detection and Ranging (LiDAR) in its latest lightweight variant can be mounted onto UAV. Like TLS, UAV-based LiDAR requires the determined position and orientation in the real time, which is difficult to obtain in Global Positioning System (GPS) - denied environments.

Unlike laser technology, photogrammetry-based point clouds are less GPS sensitive. As Greenwood et al. [39] elaborated, the imagery free of GPS data still carries more valuable information (i.e. camera positions can be photogrammetrically reconstructed based on consecutive images) than LiDAR point cloud without or with distorted GPS data.

Besides being more robust than UAV-based LiDAR, the UAV-based optical camera provides colors to the point cloud (i.e. the intensities of red, green and blue: RGB), which is very valuable information regarding classification. Leberl et al. [40] identified and reported sixteen advantages of the photogrammetry-based over the directly measured laser point clouds. The advantages addressed issues such as accuracy, economy and data types. One disadvantage of photogrammetry-based point clouds is the lower resolution comparing to laser-based ones. Nonetheless, the development of ultra-high-resolution cameras is likely to overcome this issue. The other disadvantage of photogrammetry-based point clouds is the presence of semi-penetrable objects as vegetation or water. This however cannot be avoided, nevertheless, the researchers such as Eschmann and Wundsam [41] combine the strengths of two technologies to overcome their weaknesses. Besides optical camera and LiDAR, the UAV platform often includes infrared sensors for moisture detection.

Once the imagery is acquired, the 3D scene reconstruction starts. The most common 3D reconstruction technique is Structure from Motion (SfM), comprehensively described by Golparvar-Fard et al. [42]. Besides camera characteristics and shooting options (intrinsic information), the SfM system requires camera position and orientation (extrinsic information) for each photo. The spatial coordinates obtained by GPS device in UAV certainly help in the position and orientation calculation, but as already mentioned, this information can be computed from images themselves. This can be done using Scale Invariant Feature Transforms (SIFT) and Random Sample Consensus (RANSAC) technique, as explained in [42]. The result of the 3D scene reconstruction is a point cloud.

Although powerful and widely used, UAV photogrammetry lacks image-capturing workflows for high quality deliverables. Only recently, a number of researches dealing with UAV flight path planning is published. Chen et al. [43] proposed framework for a reliable UAV-based inspection. The framework provides a method to achieve the target spatial resolution of generated point cloud by determination of UAV working distance. However,

the greatest value of the proposed framework is a method to evaluate the point cloud. It is an algorithm for elimination of the common point cloud defects such as incomplete data, outlier noise, non-uniform density, surface deviation and geometric deformation.

2.4.2 Point Cloud-based Bridge Inspection

The idea of substituting human visual perception with an automated, systematic and quantitative 3D point cloud assessment is currently an intensively investigated topic. This extends to the idea of using Unmanned Aerial Vehicles (UAVs) to acquire point cloud representations for further evaluation. Several commercial UAVs specialized for inspection were developed in the last ten years. Wells and Lovelace [44] analyzed a state-of-the-art hardware and software solutions for photogrammetry-based bridge inspection. They point out remarkable improvements of the UAVs (e.g. protected propellers, looking up camera, high resolution, distance lock, additional thermographic camera, etc.). However, 3D point clouds, as the main outputs of this kind of inspection, are not further processed, but rather used to perform the traditional inspection more conveniently. The UAVs are usually provided with an additional software for the 3D scene reconstruction. None of the analyzed software includes damage detection. Instead, the damage is manually detected and modeled as a pinned location with attached photos of the damaged region. Even though such as-is point cloud representation can be considered semantically poor, it still significantly decreases inspection costs. Wells and Lovelace [44] further compared traditional and UAV-based inspection in terms of cost and duration, inspecting 12 bridges of different types and sizes. Whereas both approaches took roughly the same time, in the same report, it is claimed that the UAV-based inspection was averagely 40% cheaper than the traditional one.

Damage detection for concrete bridges has been exhaustively investigated in the past two decades. Jahanshahi and Masri [45] developed a state of the art method for extracting an accurate two-dimensional geometry of a concrete crack from the image, whereas German, Brilakis, and Desroches [46] established a concrete spalling detection method providing the length and depth of the spalling region. None of the image-based damage detection techniques provides the damage location relative to the inspected structure.

The latest researches tend to systematize imagery acquisition techniques with damage detection and feature extraction methods into an automated bridge inspection system. Morgenthal et al. [47] proposed a conceptual framework for utilizing the state-of-the-art UAV-based bridge inspection techniques. Instead of inspection standardization, the authors suggest defining tasks and assessment criteria for each inspection. After the UAV flight path planning, the authors suggest the use of multi-scale crack centerline detection, also proposed by Sironi et al. [48], as well as the structural condition assessment by integrating the detected spalling damage into the previously generated finite element model of the bridge. Research by Hüthwohl and Brilakis [49] focused on the image-based

classification of concrete surfaces of highway bridges for damage detection, and used morphological operators to highlight the damage in surface textures that are then projected on to the given as-is model. Research by Xu and Turkan propose a framework for implementation and integration of BIM and UAV technologies for bridge maintenance. The proposed framework makes use of an image-based processing technique for detecting concrete cracks, and links this information with a string description field of the bridge IFC representation [50]. The benefits and challenges of using point-cloud data alongside BIM has been researched by Qu and Sun [51] and Tuttas et al. [52]. In these studies, the automated generation of a semantically rich model for further geometric reconstruction to as-is BIM models, are noted as the main benefits. These semantically enriched models can further be converted into BIM data or used for other type of data analysis. Additionally, Anil et al. [53] state that the use of point clouds alongside the as-designed or as-is BIM representation allows for the assessment of any conflicting differences. The use of point clouds can benefit the generation of digital documentation of new features of a structure that are added in the post-construction phase [54].

2.5 Industry Foundation Classes

Whereas the need for the development of the product model for buildings arose in the 1970s [17], only the development of the Standard for the Exchange of Product model data (STEP), in the 1980s, paved the way to a vendor-neutral format for building product models exchange. STEP is developed by the International Organization for Standardization (ISO), under the official label ISO 10303. STEP included a definition of the data modeling language, named EXPRESS, capable of describing both semantics and geometry of a building [55]. After the introduction of a vendor-neutral data modeling language, a few engineering consultancies, accompanied by the construction companies and software vendors, in 1995 founded an organization named International Alliance for Interoperability (IAI), with the purpose of standardization of building product modeling, today known as BIM. In 1997, IAI released the first version of the standard named Industry Foundation Classes 1.0 (IFC 1.0). Meanwhile, the IAI is re-branded to buildingSMART International (bSI), and the IFC 1.0 has been replaced several times by newer versions. The latest official version of the standard, IFC 4.1, is released in 2018. Borrmann et al. provided a thorough history of IFC in [56].

2.5.1 Data modeling language: EXPRESS

The biggest strength of this powerful data modeling language is the high readability by both humans and computers. The fundamental element of this language is an entity type. What class represents in an object-oriented paradigm, the entity type represents in EXPRESS. Each entity type is defined by attributes and relationships to other entity

types. As described by Borrmann et al. [56], EXPRESS is a standard by means of which it is possible to define a data model (i.e. schema), but not the instances of the defined model. For the generation of the specific model instances, several ways, such as STEP Physical file [57], or XML document are available. Although widely used, EXPRESS-G, the graphical form of EXPRESS, have not succeed in completely replacing its older brother due to the somewhat reduced expressivity. A thorough description of EXPRESS and EXPRESS-G structure and syntax is provided by Schenck and Wilson [58].

2.5.2 IFC schema structure

The object-oriented nature of the IFC data model definition implies the implementation of the inheritance concept, and thus the hierarchical classification system (i.e. taxonomy). As described by Koch and König, the inheritance allows the definition of generalized entity types (i.e. super-classes), and specialized entity types (i.e. sub-classes). Namely, the sub-class inherits all the attributes from the related super-class, but it contains additional attributes, not included in the super-class, thus specializing the super-class. On the other hand, by defining it in a superficial manner (i.e. reducing the number of attributes by which the entity is defined), super-class generalizes the sub-class [59]. This concept is utilized in the IFC data model by means of the object meaning. Here, the concept of an abstract super-class is utilized, as well. Abstract super-class is an entity type which cannot be instantiated directly, but rather via its sub-class. The example for an abstract super-class is the entity type *IfcBuildingElement*, which can be realized only by instantiation of its sub-class, such as *IfcColumn*, or *IfcWall*.

For the purpose of readability of the dissertation, the IFC [60] notation convention is provided below:

- Names of types, entities, rules, and functions start with the prefix *Ifc*, and continue with the CamelCase[2] English words.
- Names of attributes have no prefix and follow the CamelCase convention.
- Names of the standard property sets start with the prefix *Pset_* and continue with the CamelCase English words.
- Names of the standard quantity sets start with the prefix *Qto_*, and continue with the CamelCase English words.

Defined using previously defined EXPRESS language, IFC schema structured into four so-called conceptual layers (Figure 2.7), following the basic rule that element from the upper layer can reference an element from the layer below, but by no means vice versa [56]. In the following, a brief description of each layer is provided, whereas the detail description of all four layers can be found in the latest official schema documentation, IFC 4.1 [60].

[2] CamelCase is a naming convention, which stipulates capitalizing each word within a compound word.

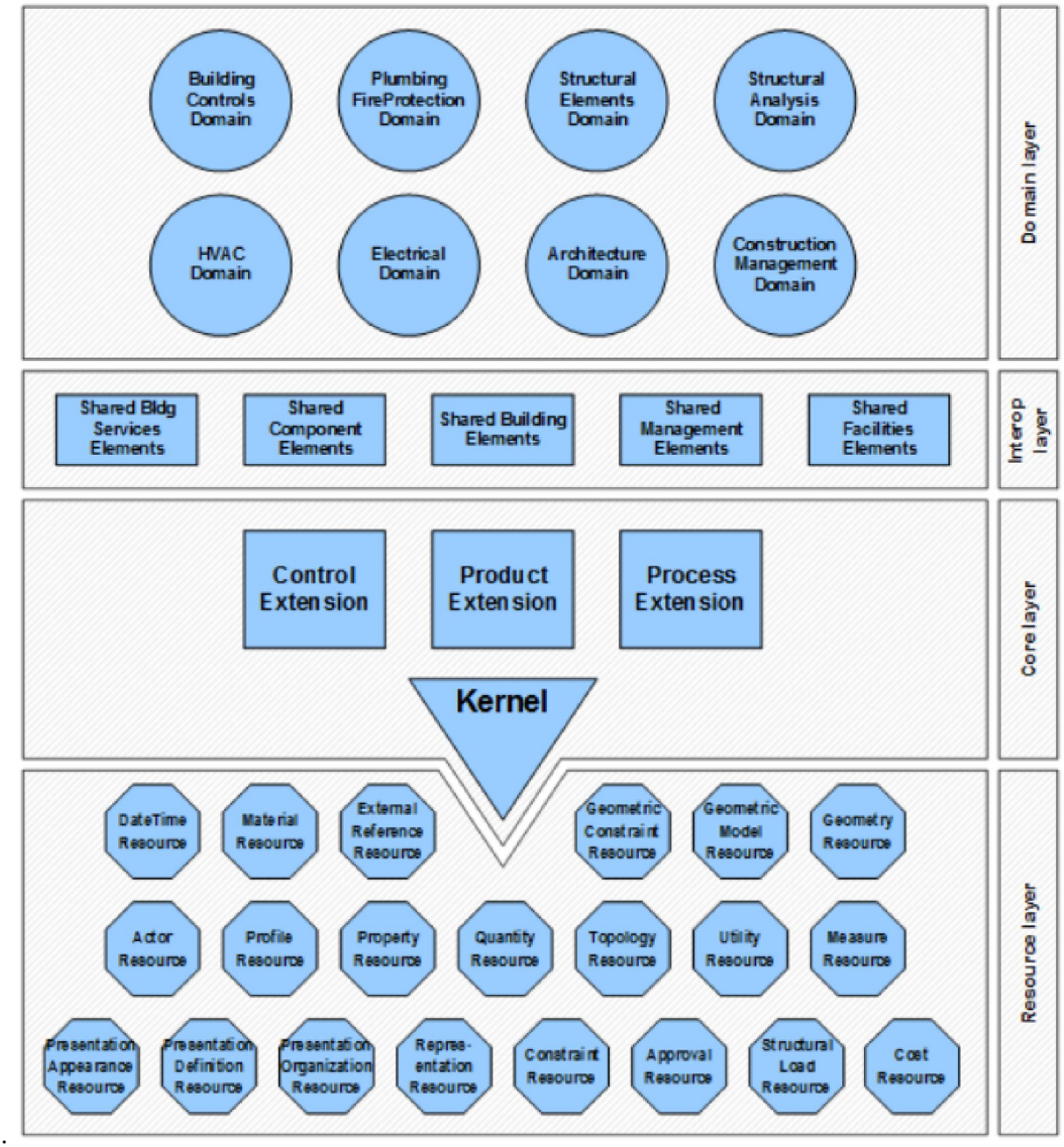


Figure 2.7. Conceptual layers in the IFC schema (retrieved from [60]).

2.5.2.1 Resource layer

This is the lowest layer, which contains definitions of basic concepts, such as measuring units, geometry, topology, properties, and property sets. According to Beetz [61], the resource level sub-schemas such as representation, geometry (includes definitions of basic geometric elements, such as points, vectors, and swept surfaces), and topology (includes definitions of classes for a solid representation), provide constructs for 2D and 3D geometry and topology description, and are therefore most used by commercial implementations.

2.5.2.2 Core layer

This layer defines the fundamental modeling constructs. It comprises *IfcControlExtension*, *IfcProcessExtension*, *IfcProductExtension*, and *IfcKernel*. The first two sub-schemas declare classes of control objects, their assignments, planning and scheduling processes. The third one, *IfcProductExtension*, defines the concepts of a physical product. By physical objects are meant components describable by the shape and placement in the project range [60]. *IfcKernel* is the core part of the IFC schema. The most abstract class of *IfcKernel* is the *IfcRoot*, providing basic attributes such as Globally Unique Identifier (GUID), owner, name and history.

As described by Beetz [61], the specificity of the IFC data model is that relationships between building elements themselves, as well as their relationships with other concepts, such as controls and processes are objects. Numerous types of relationships (e.g. decomposition, spatial connection, etc.), with instances easily identifiable by GUID, make the modification tracing easy.

2.5.2.3 Interoperability layer

This layer contains definitions of the entity types commonly used by all domains. Besides standard building elements, such as column, wall, and beam, the Interoperability layer includes definitions of management tasks, building services, and facilities.

2.5.2.4 Domain layer

This layer contains definitions of the engineering area-specific entity types and their properties. Using entities from this layer, domains such as structural elements and their analysis, HVAC, plumbing, fire protection, and electric planning can be fully described.

2.6 Semantic Enrichment of IFC model with Bridge Damage

Integrating the geometry and features of the detected damage into the Bridge Information Model (BrIM) has been a subject of research for a decade. Some researchers try to use the existing BIM software solutions to model damages and therefore commit to the proprietary data modeling formats. Others develop an openBIM-based data models. There are differences in damage data input too. Whereas some tend to keep the manual input based on inspection reports, others use digitally captured images and/or point clouds.

McGuire et al. [62] investigated the damage modeling capabilities of commercial BIM software. They tried to model damage features such as location, type, severity and volume by using LEAP Bridge, Tekla Structures and Revit. As it turned out none of the analyzed software had an embedded functionality to model a damage, they developed a Revit plugin. The proposed plugin models damage as a parametric solid parallelepiped. Relying

on traditional inspection procedure, the inspector is asked to detect damage and measure its features. Specifically, the location of the damage parallelepiped center relative to damaged element, as well as the parallelepiped dimensions. Additionally, the inspector is expected to rate the damage severity according to AASHTOWare Bridge condition state ratings [63]. McGuire et al. [62] afterwards proposed an Excel-based structural condition assessment tool implemented in VBA.

The IFC schema specifies the definitions of all the objects, their properties and mutual relationships. IFC data model strictly separates semantic and geometric representation of objects. Physical objects are defined in the Product Extension of the Core layer of IFC data mode. Objects defined in the Product Extension can have single or multiple geometric representations [64]. Targeting buildings, IFC schema is not appropriate for description of bridges. For this reason, efforts on extending the existing schema for the purpose of bridge modeling are ongoing. Started with initiative by Yabuki et al. [65], the IFCBridge schema extension, containing definitions of bridge-specific entities is recently developed and released by bSI. Specifically, it is a result of a 2-year fast-track project initiated by the Infra Room of the bSI [66]. For now, it is a candidate schema and its official acceptance is expected soon. Although the release of this schema extension will undoubtedly encourage the use of BIM technology in the design and construction phase, it still misses the definitions of Bridge Management-specific concepts, thus preventing the unfolding of BIM's full potential.

Hüthwohl et al. [67] deal with describing both the inspection details, as well as the defect type, nature, and properties. They distinguish the defect as a deterioration process ("defect") from the defect as visually observable damage on a surface of the bridge structure ("element defect"). Therefore, the "defect" semantics is modeled as IfcElementAssembly, capable of containing the aggregation of several "element defects". The condition rating of an "element defect" is represented by IfcPropertySet of predefined type Pset_Condition. The "defect" is connected with a damaged IFC element using the relationship IfcRelAggregates, so that the assignment of a single "defect" to multiple IFC elements is possible. For an "element defect" geometric representation, the IfcSurfaceFeature entity is proposed.

Isailović, Petronijević, & Hajdin [68] proposes the method for feeding the inspection database of BMS by BIMs enriched with damage information. They insert the manually detected point cloud-based damage geometry into the existing BIM by performing the CSG Boolean difference operation on damaged bridge elements.

3 DIGITAL MODEL OF AS-DAMAGED BRIDGE

3.1 Data model of as-damaged bridge

3.1.1 Data modeling concepts and notation

Koch and König [59] define a digital building model, i.e. BIM, as a „computer-based abstraction of a real facility focusing on a simplified and reduced extract of the entire set of all available information“. The same authors distinguish two basic steps in the establishment of a digital building model:

- Conceptualization – a conceptual data model, representing types of important information and their mutual relationships, is established as an abstraction of a real building.
- Realization – the conceptual model is implemented for a specific case.

For a data model to be unambiguously defined, a standardized notation should be used for the description of concepts such as entities (objects), entity types (object classes), attributes, methods, and relationships. Several conceptual data modeling notations that are widely used are Entity Relationship Diagrams (ERD) [69], Unified Modeling Language (UML) (ISO/IEC 19505), and Extensible Markup Language (XML) [70]. Besides these three, the EXPRESS is already mentioned as a language for definition of digital building models. For the establishment of the digital model for the purpose of the dissertation, UML and EXPRESS will be used. In the following, the object-oriented modeling basics are introduced using the UML notation, specifically, the essential part of UML, class diagrams.

The term entity type in data modeling is an abstraction of any kind of entity (object). In IFC, this concept is defined by the term entity type. Entity type represents a set of entities (objects), which can be described by the same properties, i.e. attributes. Furthermore, an entity type (class) describes the structure and behavior of its instances, i.e. entities (objects). Each entity type (class) is defined by attributes and methods. Whereas attributes are properties any entity (object) from the entity type (class) can have, methods can be understood as behaviors that entities (objects) from the same entity type (class) are capable of. Nevertheless, the IFC, as a data exchange format, does not provide any modification of a building model. It rather statically describes a model in a manner of a so called “time stamp”. For this reason, methods will be left out of the following discussion. Figure 3.1 shows an example of a Column entity type (class) that has two attributes: height and material. The description after the colon indicates the type of an attribute.

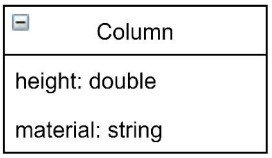


Figure 3.1. Object class: Column.

Besides the basic object generalization, by introducing entity types (classes), the entity types can be generalized as well, to form supertypes (superclasses). The opposite of the generalization is the specialization. Namely, an entity (object) derived from some supertype (superclass) can be described in more detail by specializing through the instantiation from a subtype (subclass). This way, some new attributes are introduced to the entity (object) description, on top of the ones originating from a supertype (superclass) definition. In object-oriented modeling, this concept is named inheritance. The concept of inheritance is shown in the UML class diagram in Figure 3.2. Whereas a common information about a column (i.e. height and material) is described by the attributes of a supertype Column, the information about a column base are described by the attributes of a Column subtypes: RectangularColumn and RoundColumn. Besides the generalization of subtypes (subclasses), the Column entity type represents a concept widely used in IFC, called an abstract supertype (superclass). This means that no object can be instantiated from this entity type (class), unless it is an instance of its subtype (subclass). In this example, the Column is an abstract supertype (superclass) due to the inability to instantiate a Column without the information about the base shape and dimensions.

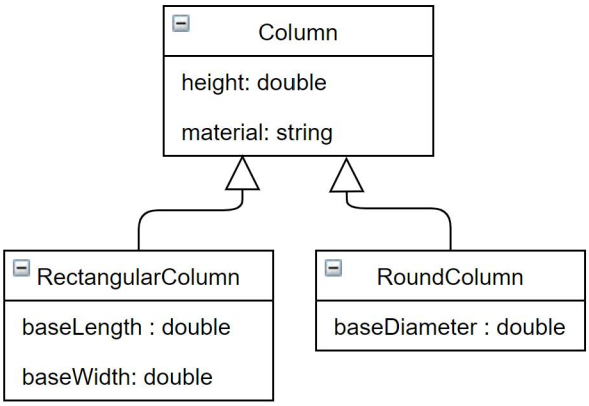


Figure 3.2. Inheritance: Subclasses of a Column superclass: RectangularColumn and RoundColumn.

In an object-oriented model, the entities (objects) of different classes are interconnected by the relation or association. According to Koch and König [59], the most common entity interconnection models are so-called binary relations. This model defines the relationship between two entities (objects) in terms of the roles and cardinality (i.e. multiplicity). Whereas the relation roles indicate the nature of a relationship, the cardinality determines the number of related entities (objects) on each relation side. Besides, the relation can be directed or undirected. The information about the entity (object) relations to other entities (objects) is stored in the entity attribute. Namely, an attribute can reference the related entity (object) either directly (Figure 3.3a) or by referencing the association class (Figure 3.3b). The association class determines the nature of a relationship, as well as the relating entities (objects).

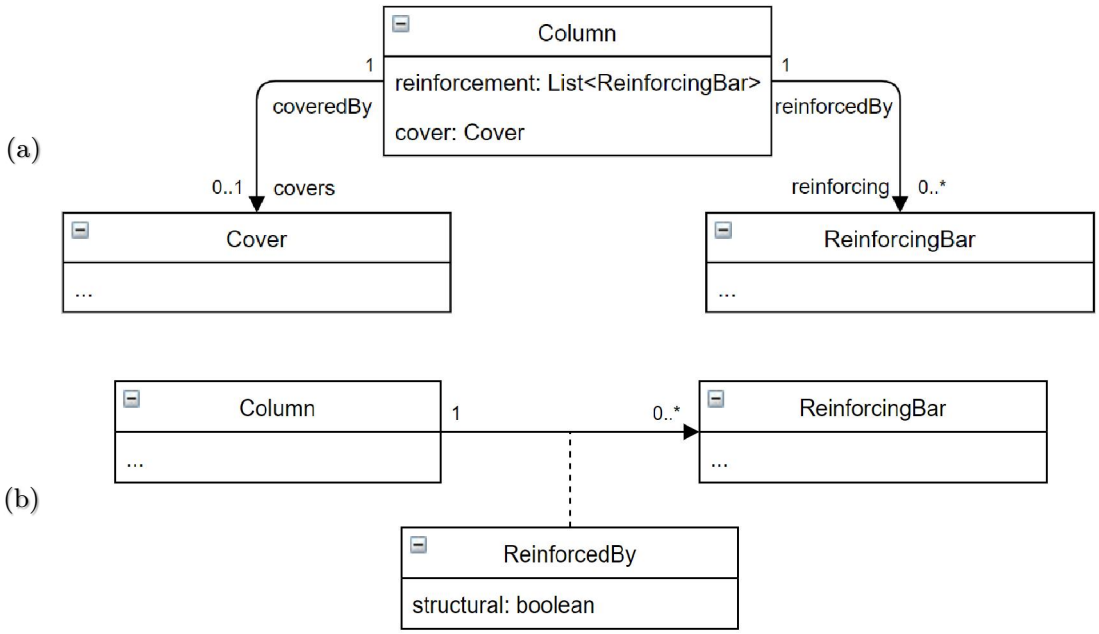


Figure 3.3. Entity relation: (a) direct (with defined roles), (b) by an association class.

For any building model, an important relationship type is a so-called "whole-part" relationship. In object-oriented modeling, this relationship is implemented through the aggregation and decomposition. These two are best described by the relations "consists-of" and "is-part-of", respectively. Koch and König [59] emphasize the strong character of relationships of this kind, explaining that parts cannot exist without the whole. To illustrate this concept, a *ReinforcingAssembly* is borrowed from the IFC 4.1 [60] (Figure 3.4). Here, the reinforcing stirrups are aggregated into the assembly, which is contained in the structural element. On one hand, the stirrups are not separately contained in the concrete element, but on the other, the assembly does not exist without single stirrups.

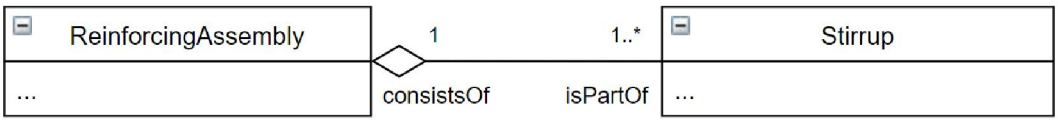


Figure 3.4. Aggregation relationship.

3.1.2 Structure of inspection data in BMS

The damage data structure is implemented differently in various BMSs around the world. Each BMS has a unique condition rating system, the format of an inspection report, damage types, etc. Semantics for the damage data model, established in the scope of this dissertation, are inspired by the Swiss Federal Roads Authority BMS, named KUBA [14].

Information about the potential bridge damage is gathered periodically, through the inspections. Each suspicious phenomenon must be recorded, as an inspection finding, in KUBA database. Observations made in course of inspections are not necessarily damages. These observations can be thought of as symptoms of damage processes, eventually leading to structural damage. It is therefore that in KUBA 5.0 the term inspection finding is used

instead of damage. Inspection finding in the KUBA inspection database must be defined by the following parameters:

- Type: To be selected from the catalog.
- Location: Textual description of the position and extent of the inspection finding.
- Placement on the inspection sketch: point, linear, or polygonal markers of the inspection finding on the inspection sketch.
- Photos: Single or multiple photos of the inspection finding, with the capture date provided.

After the inspection findings are recorded, the structural elements of the bridge are subdivided into segments. The segments correspond to the areas with the different suspected or detected long-term behavior, caused by various environmental conditions or deterioration processes, commonly named *damage processes*. Each segment is defined by the type of the *damage process* (KUBA catalog distinguishes nine types), and a measure of impact of this *damage process* on the structural element. This measure refers to the deterioration degree of progress, exposure to the environmental influences, or other influences to *damage process.*

Once the element segments are established, damages that are close to each other, caused by the same deterioration process, are identified and grouped into units named *damage groups*. *Damage group* can be only established within a segment. Same as for the segments, for each *damage group* the type of a *damage process* and the measure of impact are defined.

The final output of the inspection is the condition assessment. In KUBA, condition assessment is established on three levels: *damage groups*, building components, and an entire infrastructure asset. Swiss condition rating system distinguishes five ratings: 1 (good), 2 (acceptable), 3 (defective), 4 (poor), and 5 (alarming). All the information on the inspection planning, execution, as well as the inspection findings in KUBA is organized in an object-oriented manner. Figure 3.5 shows the adapted UML class diagram of the KUBA class *Finding*, retrieved from the IT documentation of KUBA 5.0 [71]. This diagram represents a part of the entire KUBA BMS data model, relevant for the representation of inspection and damages. The essential classes for the establishment of a digital model of as-damaged bridge are shown in red.

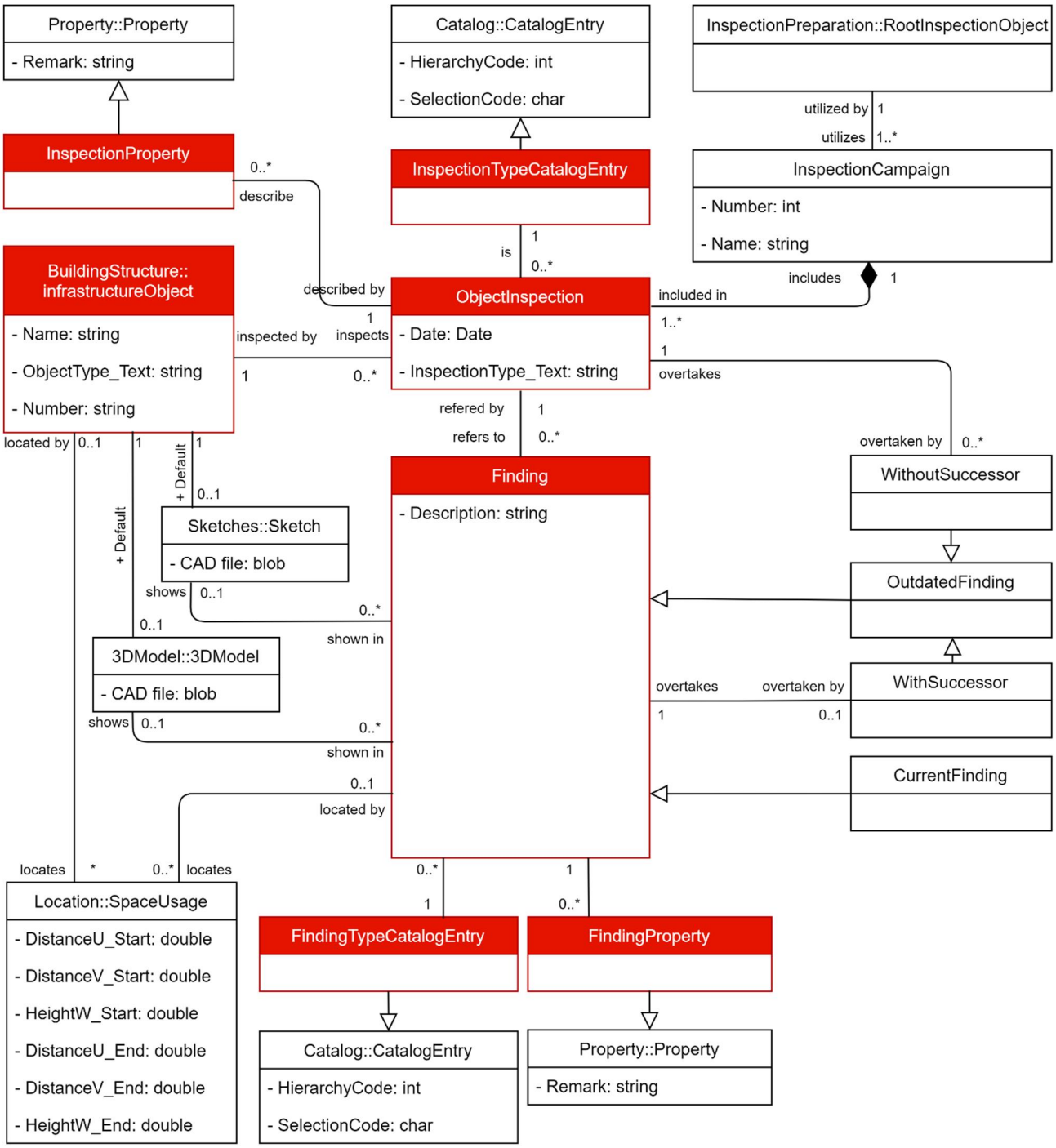


Figure 3.5. Inspection Finding UML class diagram (adapted from [71]).

The presented data model will be thoroughly analyzed by addressing each specific object class. The goal of the analysis is to derive the necessary and useful concepts from the KUBA data model, for the later utilization in the future BrIM-based data model. Therefore, a special attention in the analysis is payed to the identification of data redundancy between BrIM and KUBA.

Finding is a central class in the inspection data model. An instance of this class represents the note of a single damage, or damage symptom, detected during the inspection. The information about the damage is stored in a single textual attribute, named *Description*.

Given that the inspection is a periodically repeatable procedure, each time generating instances of the *Finding,* often addressing the same damage, damage symptom, or other suspicious phenomenon, a certain way of historical data storing needed to be established. Whereas the *CurrentFinding* represents the last valid *Finding*, which is not set to obsolete during the subsequent inspection, the *OutdatedFinding* represents the *Finding* overtaken by the subsequent one, or the *Finding* no longer detected during the subsequent inspection. Accordingly, the *WithSuccessor* represent an *OutdatedFinding*, overtaken by the subsequent one, while the *WithoutSuccessor* is an *OutdatedFinding* set to obsolete because it was no longer detected.

The *Finding* is defined by its type, represented by the *FindingTypeCatalogEntry,* and properties, represented by the *FindingProperty.* These two classes inherit from the *Catalog::CatalogEntry* and *Property::Property,* respectively. Both parent classes are parts of external packages, discoverable by the first word in a title, separated by a double-colon from the rest of the title. The *Catalog::CatalogEntry* is the part of the *Catalog* package, providing the relationship between the inspection interface and the KUBA catalog. *Catalog::CatalogEntry* points to the KUBA catalog item, identified by two attributes: *HierarchyCode* and *SelectionCode. HierarchyCode* is a numerical unique identifier of the catalog item, determining the entry position in the hierarchy of the catalog. *SelectionCode,* on the other hand, is of char type, and it defines the selectability of the entry. In particular, "-" means the entry is just a generalization of a group of entries, and thus cannot be selected, "+" represents the lower level of entry generalization, and "*" stands for the lowest entries in the catalog hierarchy, which are, of course, selectable. Figure 3.6 shows the snippet of the KUBA catalog, illustrating the described identification of specific catalog entries. *Property* package provides further information about the *Finding,* in addition to the simple catalog classification. The *Property::Property* represents either an elementary property (single object) or a complex one (an aggregation of elementary properties). The information contained in *Property::Property* can refer to the person participating in the inspection (providing all the personal and corporate information), to the assigned *FindingTypeCatalogEntry* (if a further description is needed and a property is applicable to that specific catalog entry), or it can be a free property (providing simple pieces of information, such as date, floating point number, integer, or text). Last but not least, the *Property::Property* has a textual attribute *Remark*, carrying all the additional information about the *Finding,* which cannot be included in any of the described classes.

CTK_SEL	xCTK_HIERARCHYCODE_VL	English
+	\	Another observation
-	1	Mechanical integrity
-	10	Settlements, displacements
+	1001	Scour around the structure
+	1002	Settlement
+	1003	Changes in configuration of the land
+	1004	Displacements
+	1005	Rotations
+	1006	Relative displacement
-	11	Deformation
+	1100	Deformation due to impact
+	1101	Rutting
+	1102	Bumps
+	1103	Blisters
+	1104	Shear Deformation
+	1105	Squeezed elastomer
+	1106	Deformed slide plate
+	1107	Deformed supports
+	1108	Deformed bearing plates
+	1109	Other deformations
+	1110	Deformation due to alkali aggregate reaction
+	12	Cracks
+	1201	Flexural cracks
+	1202	Shear cracks
+	1203	Cracks due to restraint deformation
+	1204	Cracks due to reinforcement corrosion
+	1205	Cracks due to sulfate attack
+	1206	Crazing
+	1207	Cracks due to the execution
+	1208	Cracks in the weld
+	1209	Cracks in the joint
+	1210	Longitudinal cracks
+	1211	Transverse cracks
+	1212	Cracks of elastics profiles
+	1213	Cracked cables
+	1214	Cracked Supports
+	1215	Cracked rollers
+	1216	Other cracks

Figure 3.6. KUBA catalog snippet.

Findings are possible outcomes of an inspection and therefore the *Finding* class is related with the *ObjectInspection* class. Inspections are scheduled or triggered according to a certain plan, or a strategy for bridge management and maintenance. All the performed inspections (i.e. *ObjectInspection*) of a single bridge are grouped into a so-called campaign (i.e. *InspectionCampaign*). Each campaign is identified by a unique number and an optional name. *InspectionCampaign* is implemented by means of the classes from the *InspectionPreparation* package, whose represent in the class diagram in Figure 3.5 is *InspectionPreparation::RootInspectionObject.* It is the top inspection object in the inspection tree, which allows various ways to group several bridges or several elements of one bridge, to correspond to the specific inspection campaign.

Same as *Finding,* each *ObjectInspection* instance must be related to the specific catalog entry and can be further described by certain properties. This is implemented by relating the *ObjectInspection* with the *InspectionTypeCatalogEntry* and *InspectionProperty,* specializations of *Catalog::CatalogEntry* and *Property::Property,* respectively.

Apart from being related with the *Finding*, *ObjectInspection* is also related with the *BuildingStructure::InfrastructureObject.* This class is a part of the package used for the description of the inspected infrastructure assets, named *BuildingStructure.* The inspected bridge, its structural and non-structural elements, their mutual relationships, as well as corresponding metadata is represented by the *InfrastructureObject* class and similar classes from the *BuildingStructure* package. All the information modeled by the resources of this package is already included in BrIM, therefore it will not be discussed further.

The UML definition of classes *Location::SpaceUsage, Sketches::Sketch,* and *3DModel::3DModel* in Figure 3.5 are considered self-explanatory, thus it will not be discussed further. Moreover, the existence of these classes in the KUBA data model can be understood as a compensation of a comprehensive geometric model, such as BrIM. Therefore, the introduction of BrIM makes these classes, as well as the corresponding packages redundant.

3.1.3 Establishment of a new data model for as-damaged bridge

Based on the analysis from the previous section and the structure and content of the current BrIM data model, requirements for the data model on as-damaged bridge will be formulated. Afterwards, a selection of relevant concepts and information from the existing data model for damage description will be established. Finally, a new data model will be proposed in a form of a UML class diagram.

General requirements the newly established data model needs to meet are as follows:

- Complete preservation of a BrIM and BMS data structure
- Highly detailed damage geometry and semantics
- Straightforward linking between BrIM and BMS objects

For each noticed damage, the following information needs to be assessed and documented:

- Damage type: classification of a visible surface defect, selected from the BMS catalog.
- Deterioration process: physical-chemical process causing surface defects, selected from the BMS catalog (KUBA distinguishes nine deterioration processes).
- Damage position: rough distance measure, relative to the dimension of the inspected element.
- Damage extent: an approximate measure of the damaged region (areal dimension or percentage of the damaged region relative to the overall element surface).
- Damage severity: damage condition rating complying with BMS damage rating system.

The position and the extent will be implicitly determined by the as-is IFC geometry. State of the art tools for BIM analysis, such as spatial query language QL4BIM [72], are capable of sophisticated analysis of mutual relationships between IFC objects. However, no straightforward solution appropriate for damage severity assessment for bridges is currently available. In most cases, due to the complexity of the task and the required expertise, the deterioration process has to be manually assessed by an experienced and in some countries licensed structural engineer.

For the easier distinction between the KUBA-based *Finding*, the corresponding class in the newly established data model is named *InspectionFinding.* Figure 3.7 depicts the proposed class diagram of *InspectionFinding*, compliant to the data structure of KUBA 5.0.

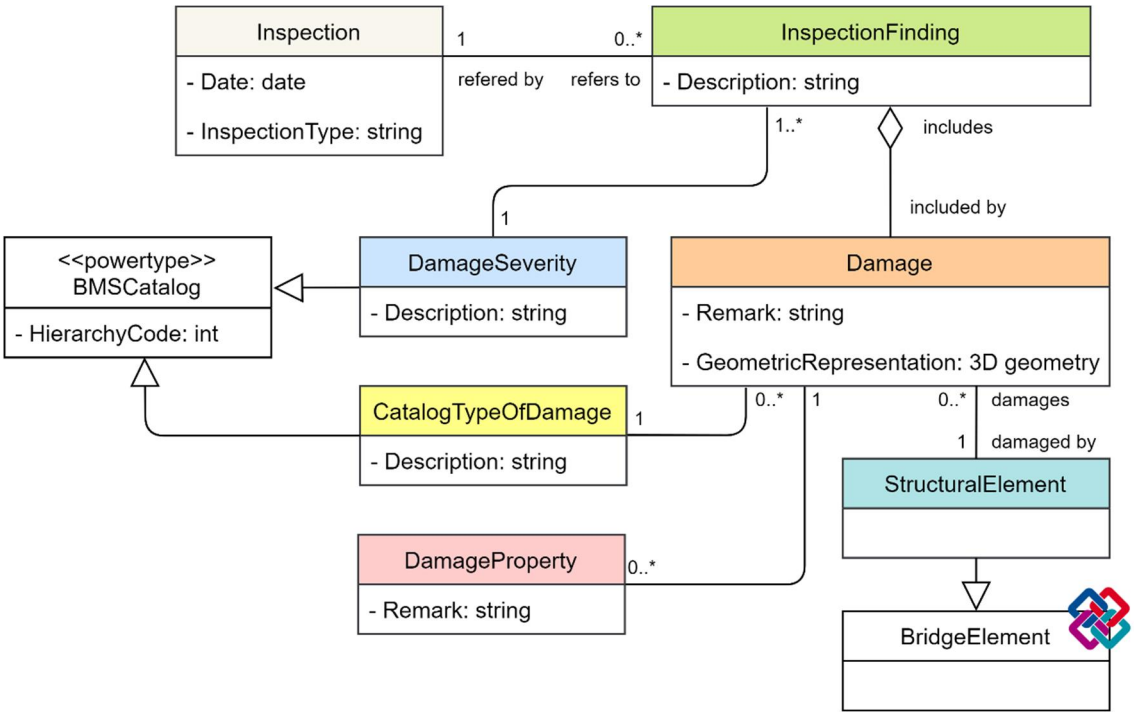


Figure 3.7. Newly proposed InspectionFinding class diagram.

In the proposed data model, *InspectionFinding* should be understood rather as a damage group, thus modeled as an aggregation of *Damage* instances. The only attribute of *InspectionFinding* is the textual description. It is associated with *Inspection*, described by date and type. The *InspectionFinding* as a damage group is described by *DamageSeverity*, which is a predefined catalog entry, or a free text. *Damage* class has two attributes: *Remark* (string) and *GeometricRepresentation* (at this point unidentified format). Besides by its attributes, *Damage* is described by *CatalogTypeOfDamage* and *Damage property*. As opposed to *CatalogTypeOfDamage*, the *DamageProperty* is optional, added only if the damage extent cannot - at the moment - be precisely derived (e.g., crack width)

from the *GeometricRepresentation*. Finally, *Damage* is related to the damaged *StructuralElement,* coming from BrIM.

3.2 Data model implementation in IFC

In order to enable the use of BrIM by BMS, mapping of the proposed classes (entity types) to the IFC is necessary. IFC data model strictly separates semantic from the geometric representation of a bridge. Therefore, these two will be addressed separately by the following sections. Rather than proposing the schema extension, the existing schema [60] definitions are used. Colors of IFC entities in the proposed EXPRESS-G diagrams in the following figures correspond to the colors of the classes in UML diagrams.

3.2.1 Semantics

As previously described, findings are results of carefully planned and scheduled inspections. Adopting the proposal of Hüthwohl et al. [67], in the IFC model *Inspection* is represented by *IfcTask*. Its attributes *TaskTime* (time stamp) and user-defined *ObjectType* perfectly correspond to the definition of *Inspection* class (Figure 3.7), and therefore, the *ObjectInspection* (Figure 3.5) as well. *Inspection* (*IfcTask*) is connected with *InspectionFinding* (*IfcElementAssembly*) by relationship *IfcRelAssignToProcess* (Figure 3.8).

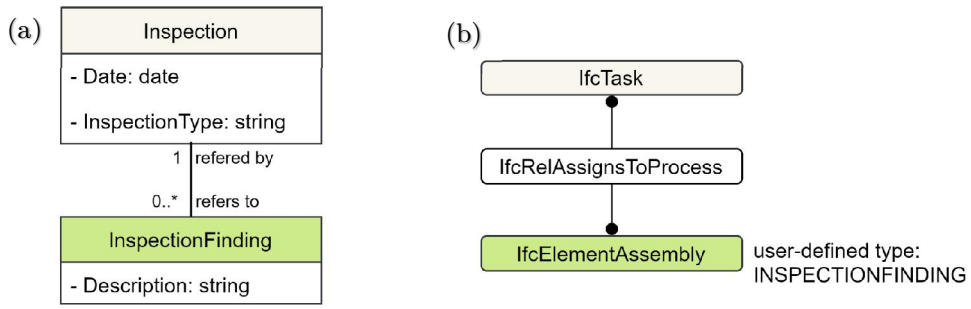


Figure 3.8. Mapping the *Inspection* class and its relationship with *InspectionFinding* to IFC: (a) Proposed UML class diagram, (b): proposed IFC structure in EXPRESS-G.

As described in Section 3.1.2., KUBA 5.0 groups mutually close damages related to the same deterioration process. Therefore, the *InspectionFinding* is modeled as *IfcElementAssembly* of the user-defined type *INSPECTIONFINDING* (Figure 3.8). The members of this aggregation, instances of the *Damage* class, are represented in IFC by the *IfcSurfaceFeature* (Figure 3.9).

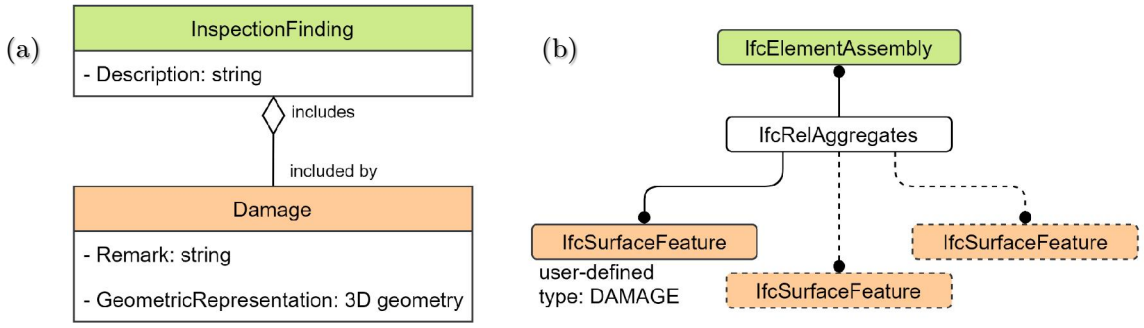


Figure 3.9. Mapping the *InspectionFinding* class and its relationship with *Damage* class to IFC: (a) Proposed UML class diagram, (b): proposed IFC structure in EXPRESS-G.

IFC schema defines the *IfcSurfaceFeature* as "a modification at (onto, or into) of the surface of an element" [60]. This modification can affect only the part of an element surface, or the entire surface. However, the consequential change of the element's volume must be proportionally small. Originally intended for description of the designed surface features, such as wall covers, or small holes for the plumbing pipes, *IfcSurfaceFeature* is able to represent damage as well. This have been firstly noticed by Hüthwohl et al. [67], who used *IfcSurfaceFeature* to map an image of a damage on to the bridge element surface. Moreover, the IFC Bridge schema extension [73], which will be official soon, suggests using *IfcSurfaceFeature* to describe any surface defect. The IFC Bridge expands the current set of *IfcSurfaceFeature* predefined types with the type named *DEFECT*. Contrary to Hüthwohl et al. [67], in this research, *IfcSurfaceFeature* will be used to subtract the missing volume, caused by damage, from the original volume of the bridge element. The IFC Bridge's predefined *DEFECT* type is not used in this dissertation. Instead, the user-defined type of *IfcSurfaceFeature,* named *DAMAGE* is used to describe bridge damages. The reason for this is the chronology of the establishment of these two solutions. Namely, the user-defined *DAMAGE* is used in this research before the IFC Bridge schema extension [73] has been released. Although the choice of the *IfcSurfaceFeature* type could be easily changed in favor of the new extension, it has not been done. It is rather postponed for the moment when IFC Bridge is improved with entity type definitions addressing bridge management requirements.

KUBA partitions each bridge element based on damage groups found at that element. Furthermore, the condition rating (e.g., *DamageSeverity*) refers to the damage extent, which includes damages of the same group. Thus, *DamageSeverity* is represented by *AssesmentCondition*, *IfcPropertySingleValue* of the predefined *IfcPropertySet* named *Pset_Condition*, connected with the *InspectionFinding* by relationship *IfcRelDefinesByProperties* (Figure 3.10).

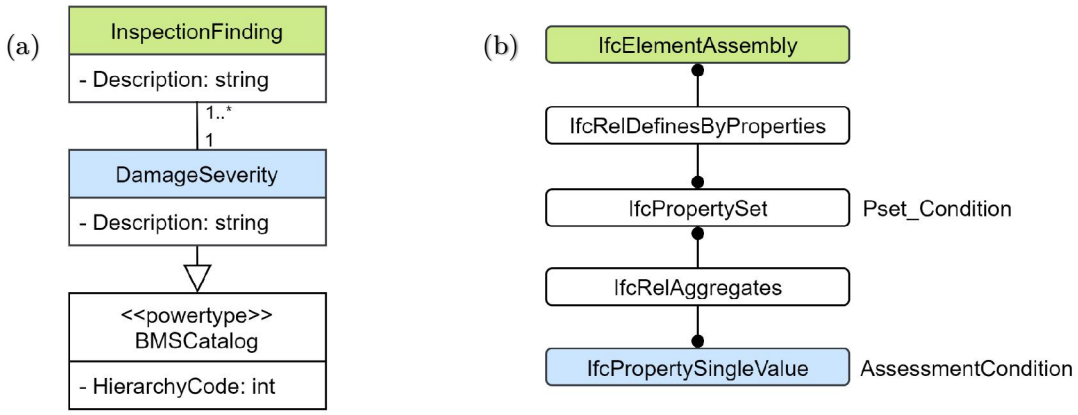


Figure 3.10. Mapping the *DamageSeverity* class and its relationship with *InspectionFinding* class to IFC: (a) Proposed UML class diagram, (b): proposed IFC structure in EXPRESS-G.

CatalogTypeOfDamage and *DamageProperty* are represented by instances of *IfcPropertySingleValue*, members of *IfcPropertySet*, connected with *Damage* by relationship *IfcRelDefinesByProperties* (Figure 3.11).

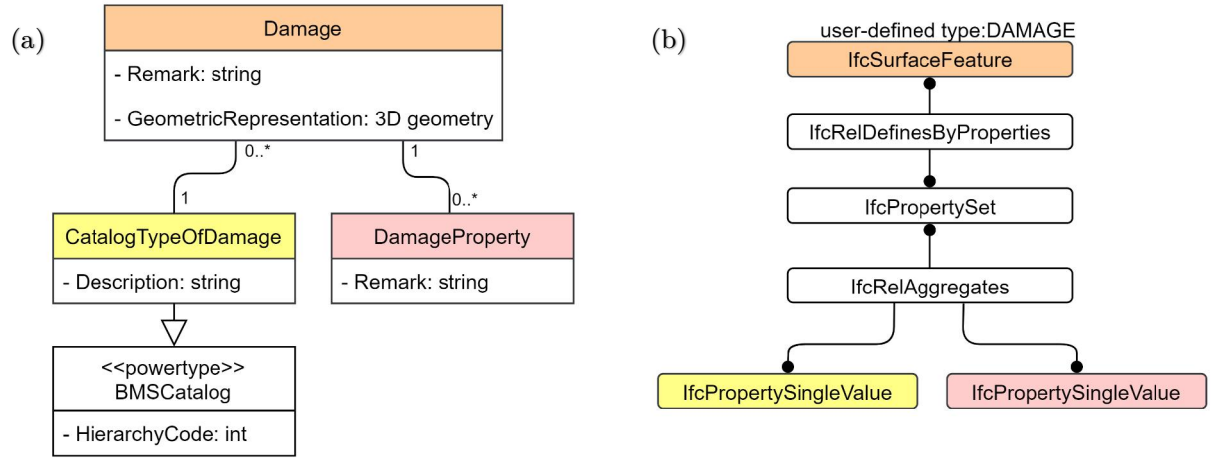


Figure 3.11. Mapping *CatalogTypeOfDamage, DamageProperty*, and its relationship with *Damage* class to IFC: (a) Proposed UML class diagram, (b): proposed IFC structure in EXPRESS-G.

Any of the structural bridge elements can be damaged, so *IfcElement*, an abstract superclass of all the structural components, is used to describe damaged elements in Figure 3.12. Damage (*IfcSurfaceFeature* instance) is connected with the damaged element of the bridge (represented by the instance of *IfcElement*) by the relationship *IfcRelVoidsElement*. The complete proposal of the IFC structure is shown in Figure 3.13.

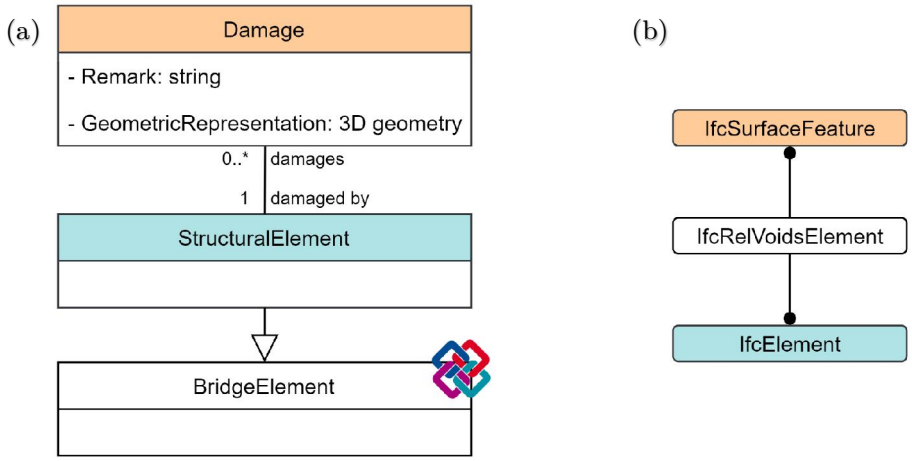


Figure 3.12. Mapping the relationship between *Damage* and damaged *BridgeElement*: (a) Proposed UML class diagram, (b): proposed IFC structure in EXPRESS-G.

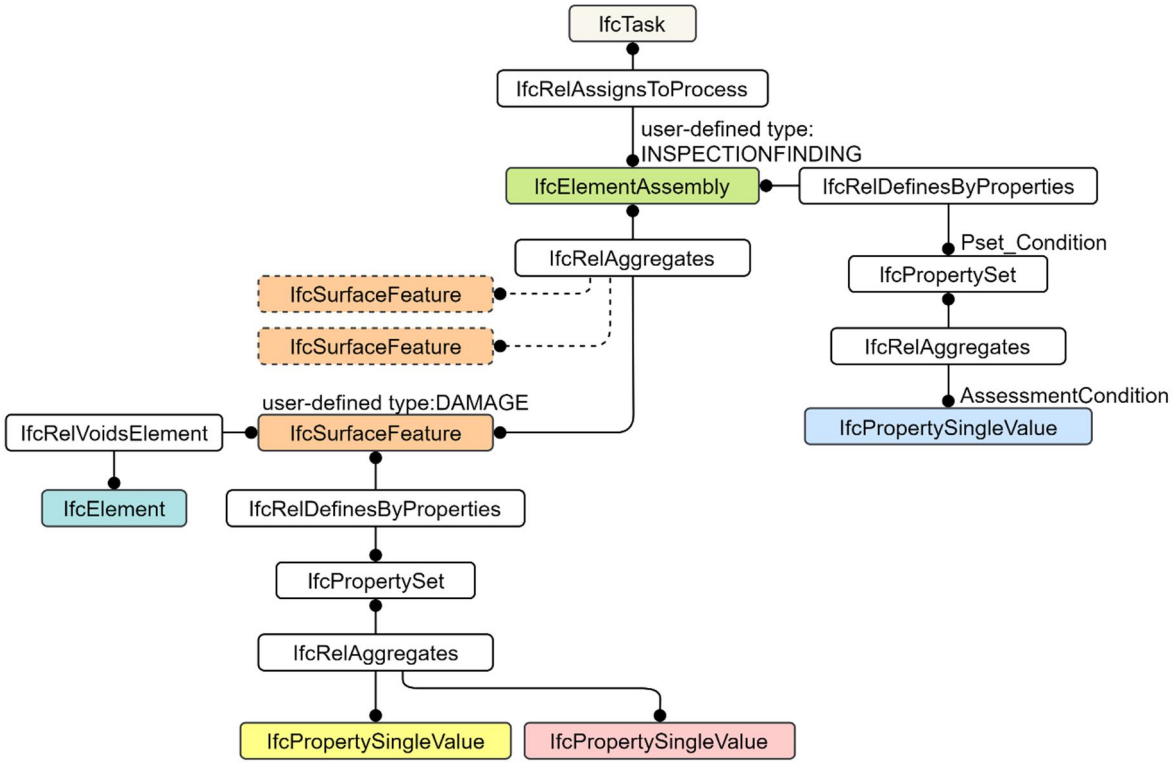


Figure 3.13. Complete IFC structure for description of inspection findings (EXPRESS-G).

3.2.2 Geometry

IFC data model strictly separates semantic information from its geometric representation. According to Borrmann et al. [56], every object in a building project is defined by its semantic identity, which can be further linked to a single or multiple geometric representation. Each instance of the *IfcProduct* supertype can have a geometric representation. In IFC, the geometry is modeled using the entity types from Geometry Resource, Geometric Model Resource, or Topology Resource sub-schema. The IFC schema offers several geometry definitions, specified in the abstract supertype

IfcGeometricRepresentationItem. Borrmann et al. have grouped those definitions into classes for definition of curves, spatial surfaces, and solids. Due to the topology of the BrIM geometry and UAV photogrammetry outputs, solid and surface representations were in the focus of the research. Defining geometry of damage is only possible in context of the entire bridge and therefore both the geometry of intact bridge, as well as the proposed geometric representation of damage is described.

3.2.2.1 Intact bridge elements

BrIM models are created using BIM software, whose user interface hides the internal process of generating graphic artefacts. Nevertheless, understanding of computer graphic basics is essential for the development of methods for introduction of new artefacts to the model. Being solid objects, bridges and their components are geometrically described in IFC as solids (Figure 3.14).

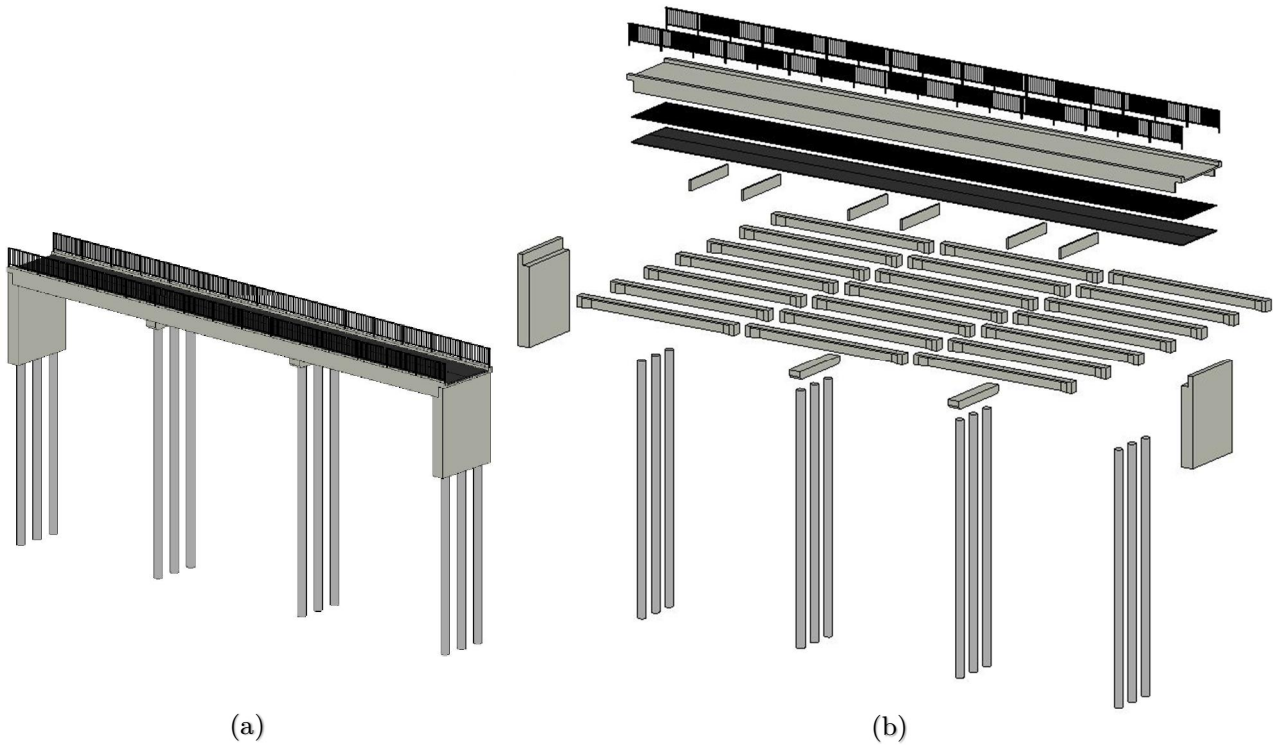


Figure 3.14. Geometric representation of the bridge: (a) integral view, (b) exploded view.

As thoroughly explained by Borrmann et al. [56], IFC schema offers several different ways for the geometric description of 3D solids. Three basic methods for modeling of solids are Boundary Representation (BRep), Rotation, Extrusion, and Swept Solids, and Constructive Solid Geometry (CSG). In the following subsections, a brief description of these methods will be presented. Further explanation of these methods in general is provided by Borrmann and Berkhahn [74], while its implementation in IFC is thoroughly described by Borrmann et al. [56], as well as in the IFC 4.1 documentation [60].

3.2.2.2 Boundary representation

This method is based on the hierarchy of following boundary elements: *body*, *face*, *edge*, and *vertex,* in that order. Lower elements in the hierarchy describe the upper ones, thus two vertices define an edge, edges define the face, and faces define the body.

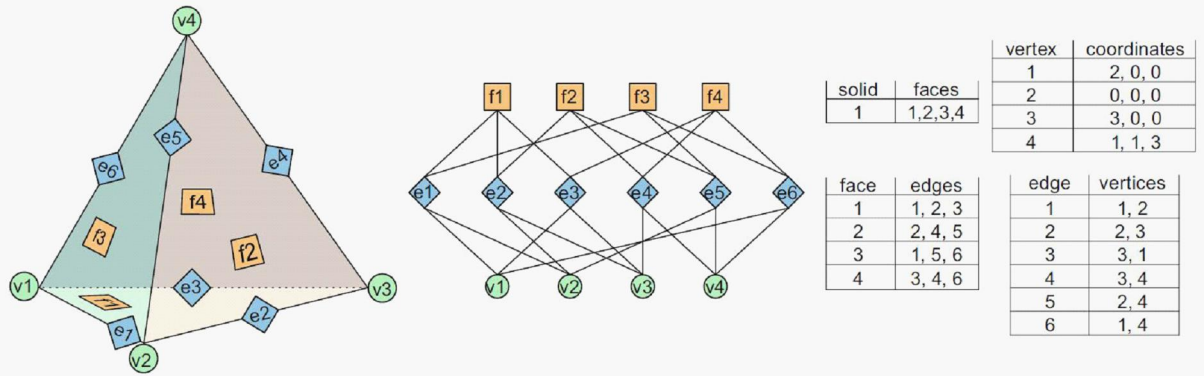


Figure 3.15. BRep description of a pyramid (adapted from [74]).

Whereas straight-edged bodies are described by coordinates of vertices, for the description of curve-edged bodies, the additional information on splines used for the edge description, so-called discontinuities, need to be provided. This data structure is, however, sufficient only for the description of bodies without voids. On the other hand, the BRep extension, named ACIS, is capable of modeling more complex bodies.

3.2.2.3 Extrusion and rotation

The principle of this method is shifting of a closed planar boundary along a line, that can be either straight or spatially curved. The former guide-line produces an extrusion, whereas the later one produces a sweep. An additional variant of this method is called lofting, and it refers to the interpolation of several parallel cross-sections, possibly of different shapes and sizes. Figure 3.16 shows all described variants of this method.

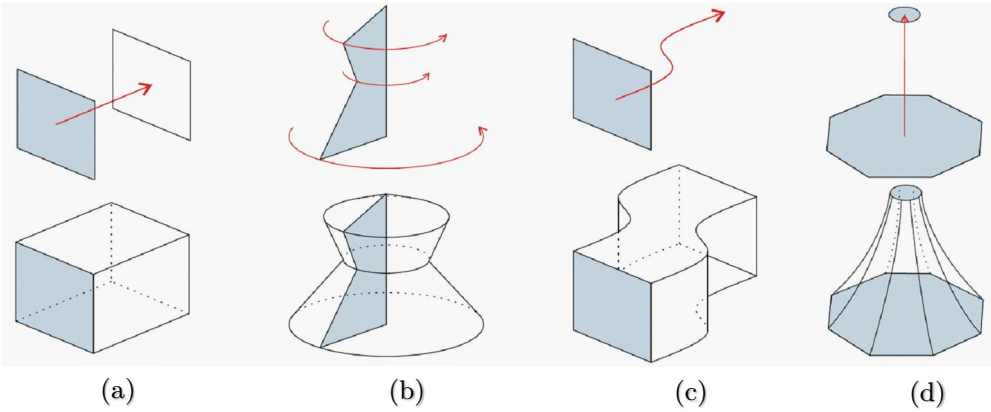


Figure 3.16. (a) Extrusion, (b) Rotation, (c) Sweep, (d) Lofting (adapted from [74]).

3.2.2.4 Constructive Solid Geometry

CSG method generates a new object combining predefined objects named primitives (e.g. cubes, spheres, pyramids, cones) by applying Boolean operations (i.e. difference,

intersection, or union). Although powerful, using only primitives as Boolean operands, significantly narrows the range of possible resulting shapes. Therefore, IFC, as well as many other data formats enable applying CSG operations on objects of any shape. Figure 3.17 shows products of CSG operations between a cube and a sphere.

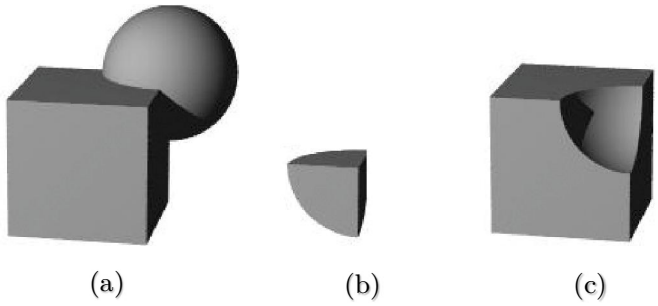


Figure 3.17. CSG operations on a cube and a sphere: (a) union, (b) intersection, and (c) difference (retrieved from [75]).

3.2.2.5 Damage

The output of a UAV photogrammetry-based bridge inspection is a 3D point cloud of the bridge. By further processing, point cloud is turned into a triangular mesh. A detail description of available algorithms for a mesh reconstruction is provided by Isailović et al. [76]. In Figure 3.18, parts of a point cloud and triangular mesh of a damaged cylindrical column are shown as an example of described photogrammetric outputs.

According to the previous description, the current schema, IFC 4.1 [60], is thoroughly analyzed in order to find the most appropriate geometric representation entity type complying with the triangular mesh data structure. The general idea for introducing damage to BrIM, in terms of geometry, is to treat it as a material void and to subtract its volume from the intact element.

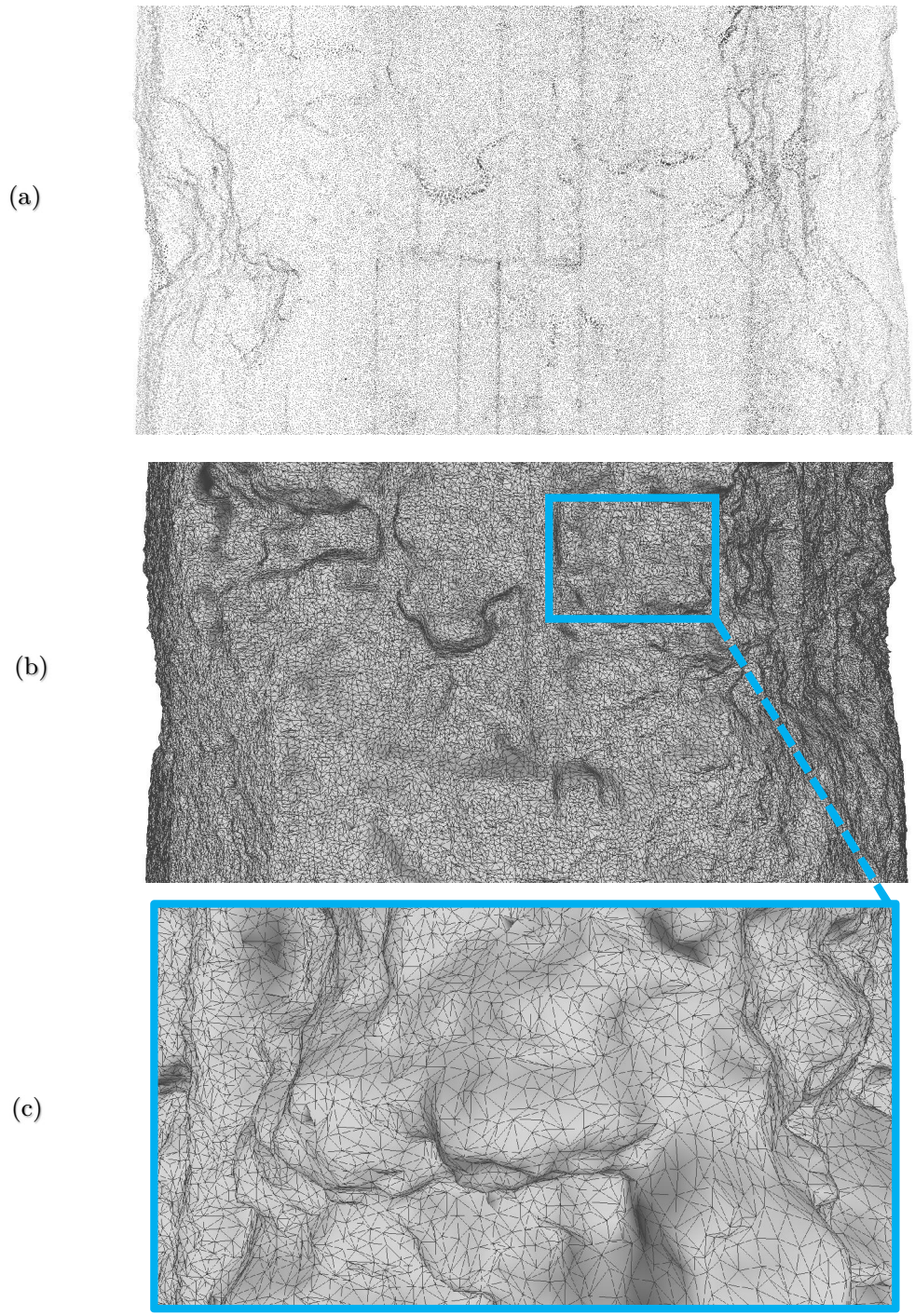


Figure 3.18. Outputs of a photogrammetry-based inspection: (a) 3D point cloud, (b) reconstructed triangular mesh, (c) close view to the triangular mesh.

A detail description of the actual CSG subtraction is given in Section 4.3. Here, just a resulting mesh and its IFC implementation as a concrete damage will be considered.

The *Representation* attribute of *IfcSurfaceFeature* points to an instance of a certain subtype of *IfcProductDefinitionShape.* The chosen IFC entity for the geometry representation of damage is *IfcTriangulatedFaceSet,* a tessellated set of triangular faces. Table 3.1 shows attribute definitions of this entity type. *CoordIndex* is a two-dimensional list, of which the first dimension represents the list of triangular faces, whereas the second

one provides the indices of three Cartesian points, triangle vertices. The *Coordinates* attribute points to the *IfcCartesianPointList3D,* an IFC entity type representing an ordered collection of three-dimensional Cartesian points, referred by a *CoordIndex.* Although an indirect vertex indexing is possible using a *PnIndex,* which is an integer value providing the coordinate values location in the *IfcCartesianPointList3D,* here the indices will point directly into the *IfcCartesianPointList3D.* A reason for this is the structure of Wavefront OBJ, mesh data format used for generating the geometric representation of damage in IFC, where face vertices are also directly addressed. The *Normals* attribute, intended to provide the surface orientation, is also left valueless since the orientation is completed during the mesh pre-processing.

Table 3.1. Definitions of IfcTriangulatedFaceSet attributes (adapted from [60]).

#	Attribute	Type	Cardinality	Description
1	Coordinates	IfcCartesianPointList3D		An ordered list of Cartesian points used by the coordinate index defined at the subtypes of IfcTessellatedFaceSet.
2	Normals	IfcParameterValue	? L[1:?] L[3:3]	An ordered list of three directions for normals. It is a two-dimensional list of directions provided by three parameter values. • The first dimension corresponds to the vertex indices of the *Coordindex (i.e. triangle).* • The second dimension has exactly three values, [1] the x-direction, [2] the y-direction and [3] the z-directions
3	Closed	IfcBoolean	?	Indication whether the IfcTriangulatedFaceSet is a closed shell or not. If omitted no such information can be provided.
4	CoordIndex	IfcPositiveInteger	L[1:?] L[3:3]	Two-dimensional list for the indexed-based triangles, where • The first dimension represents the triangles (from 1 to N) • The second dimension has exactly three values representing the indices to three vertex points (from 1 to 3). NOTE: The coordinates of the vertices are provided by the indexed list of *SELF\IfcTessellatedFaceSet.Coordinates.CoordList.*
5	PnIndex	IfcPositiveInteger	? L[1:?]	The list of integers defining the locations in the IfcCartesianPointList3D to obtain the point coordinates for the indices within the *CoordIndex*. If the *PnIndex* is not provided the indices point directly into the IfcCartesianPointList3D.
	NumberOfTriangles: =SIZEOF(CoordIndex)	IfcInteger		Derived number of triangles used for this triangulation.

Defined this way, geometric representation of the defect is self-sufficient for any kind of further structural analysis or condition assessment. An example of several instances of the *IfcTriangulatedFaceSet* representing the concrete spalling is shown in Figure 3.19.

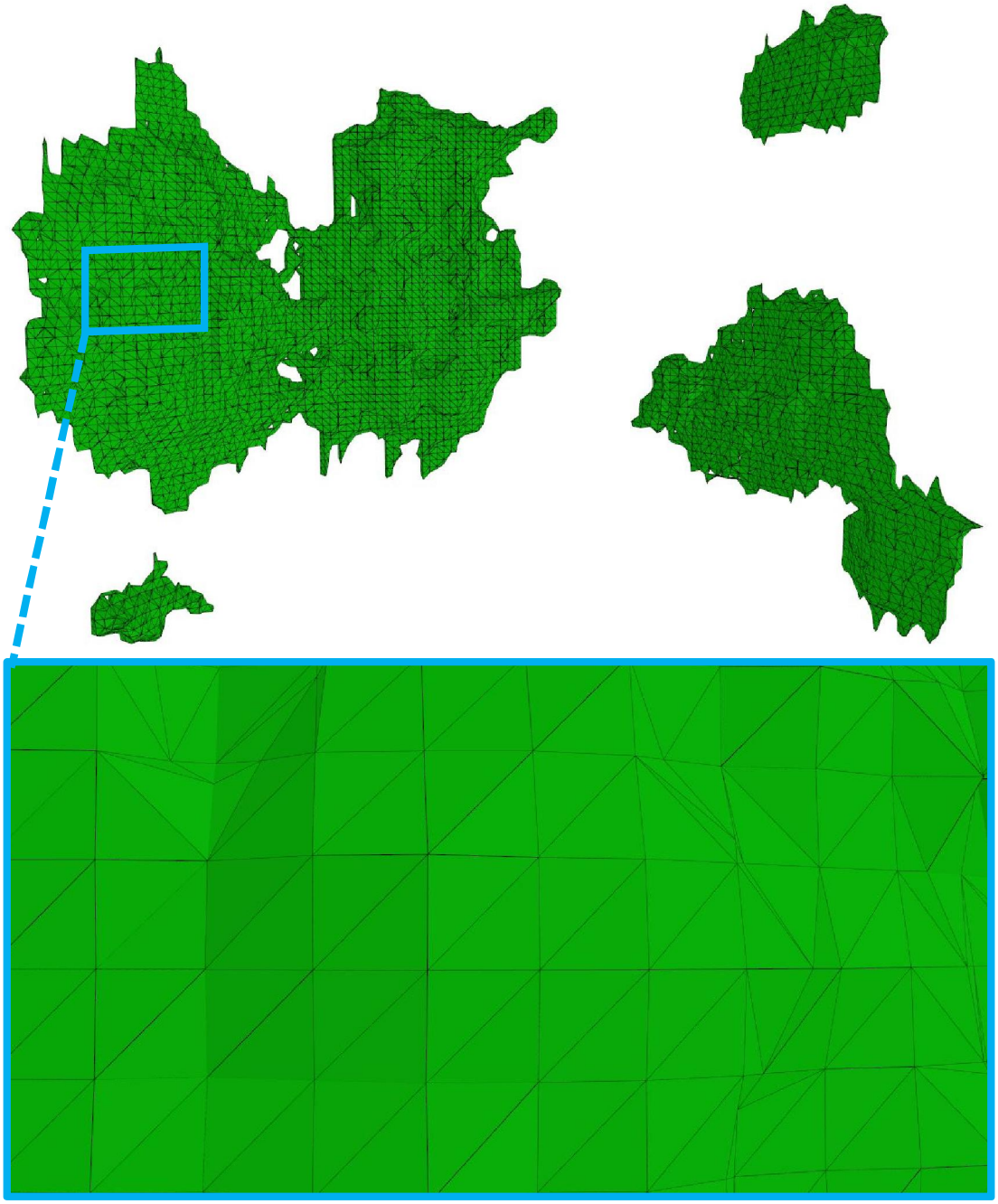


Figure 3.19. Geometric representation of damage: IfcTriangulatedFaceSet.

Once defined, damage geometry will be realized as a void in a volume of the damaged element only if properly positioned, and the appropriate relationship with the element is established. Figure 3.20 illustrates the IFC representation of concrete spalling on the bridge column. The IFC objects describing spalling are colored green, whereas the objects describing column are colored grey. The IFC objects defining the relationship between the spalling and the column are colored red.

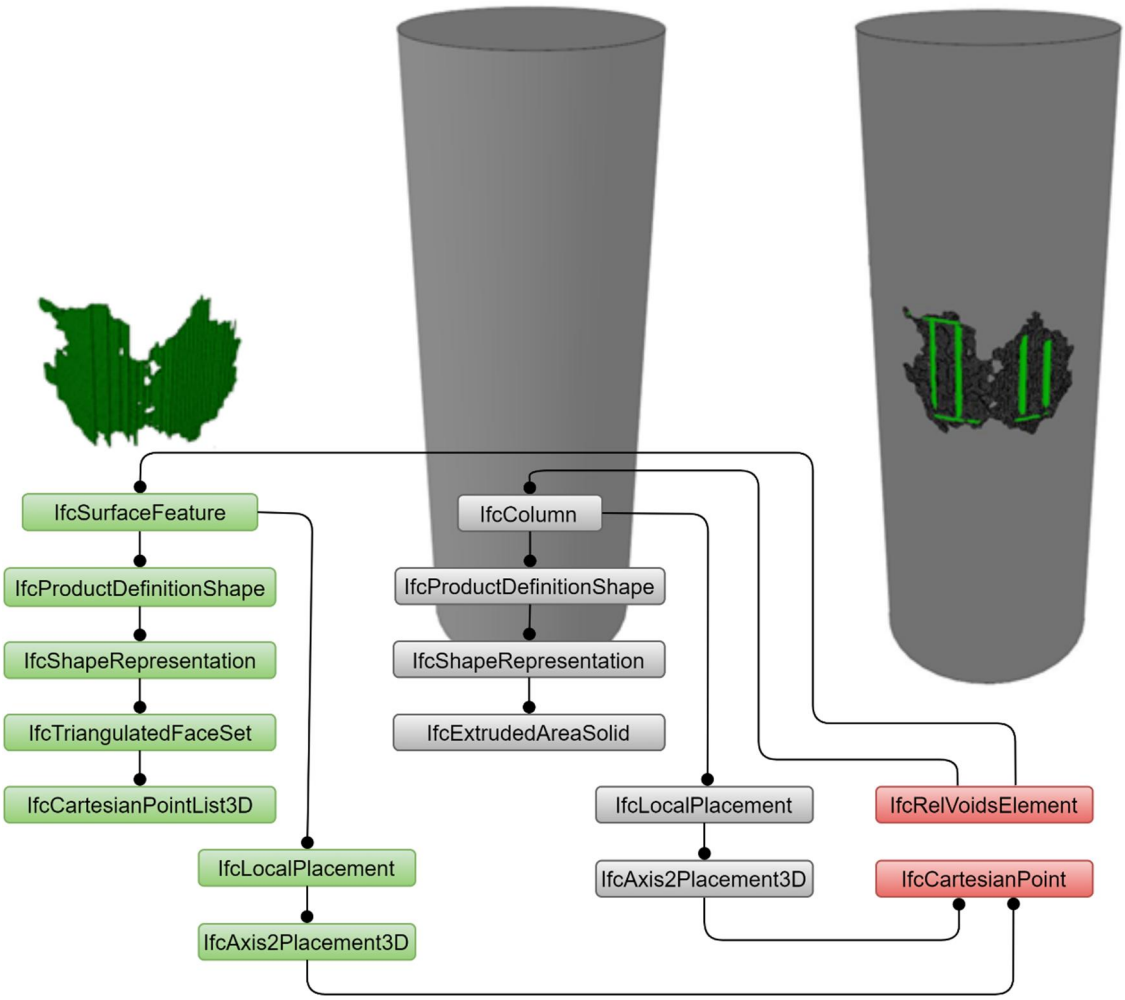


Figure 3.20. IFC structure for geometric representation of damage.

The *IfcSurfaceFeature* instance must be hosted by an instance of a child of an *IfcElement* (in this case *IfcColumn*). This is implemented by the *IfcRelVoidsElement* relationship between the *IfcSurfaceFeature* and *IfcColumn*. This relationship ensures an automatic computation of the result of CSG difference between geometric representations of those two objects every time the model is to be rendered in IFC viewer. The local placement (*IfcLocalPlacement*) of both objects refers to the same instance of *IfcCartesianPoint*, so the previously performed alignment between IFC and point cloud model representation is preserved (e.g., the volumes of the objects overlap). Although nested in the geometric representation of a column (*IfcColumn*), reinforcement bars (*IfcReinforcingBar*) are not voided. Instead, they stick out of damaged elements. They also mostly correspond to real spalling geometry and therefore this representation allows computation of the extent of exposed reinforcement. Figure 3.21 shows the final geometric representation of damage in BrIM, along with the corresponding photogrammetric mesh.

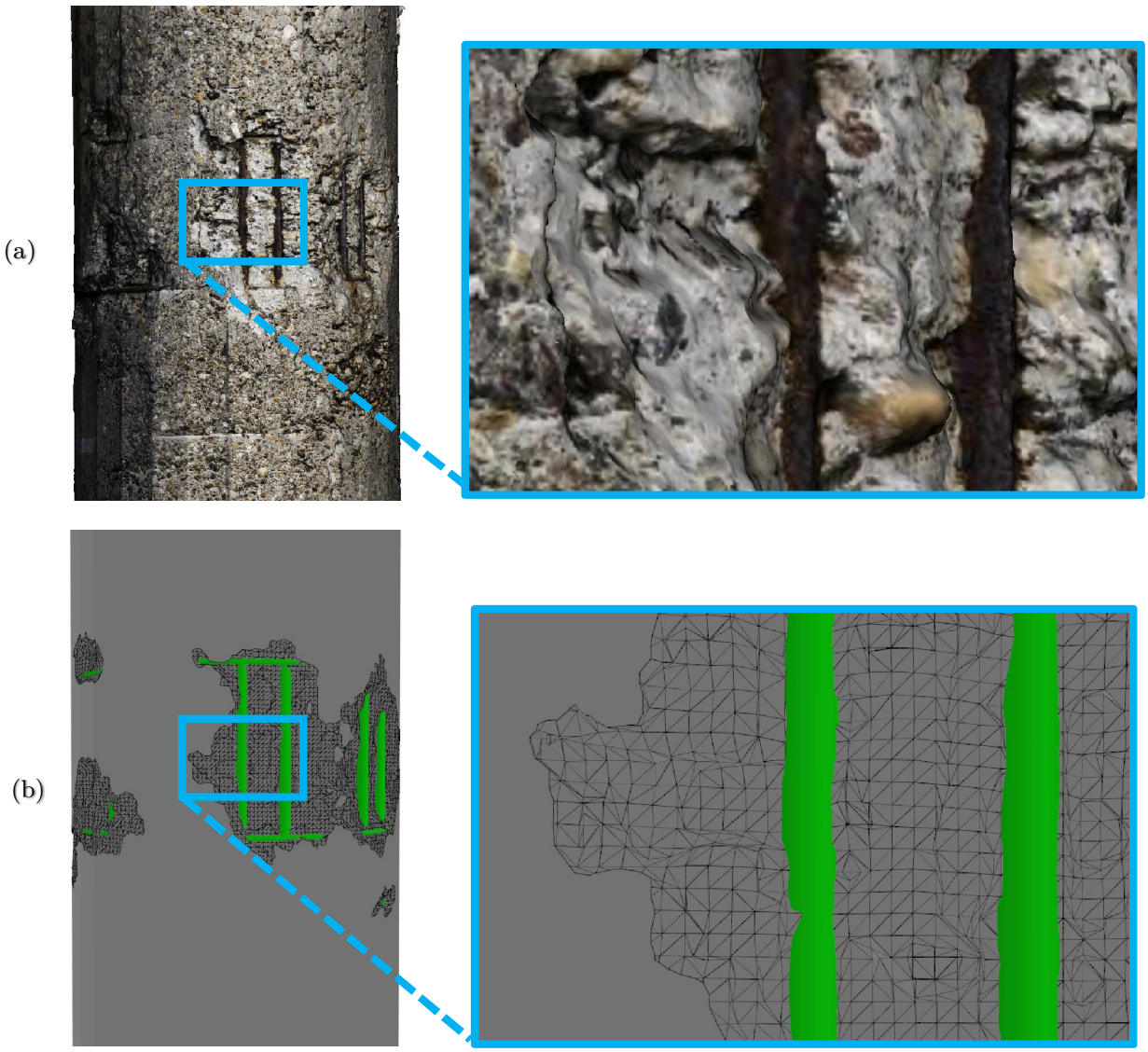


Figure 3.21. Comparative concrete spalling representation: (a) photogrammetric mesh, and (b) proposed IFC geometric representation.

4 GENERATION OF AS-IS BRIDGE INFORMATION MODEL

4.1 Overview

Once defined, digital model of as-damaged bridge is instantiated by generation of BrIM, compliant to the proposed model. The essential input for this procedure is an as-designed BrIM, which will be further enriched by inspection findings to produce the as-is BrIM. This chapter thoroughly describes the process of generating the as-is BrIM compliant to the data model proposed in Chapter 3.

The method for generating the as-is BrIM is described using the Business Process Modeling and Notation (BPMN) [77] (Figure 4.1). In the following sections, the process will be explained step by step.

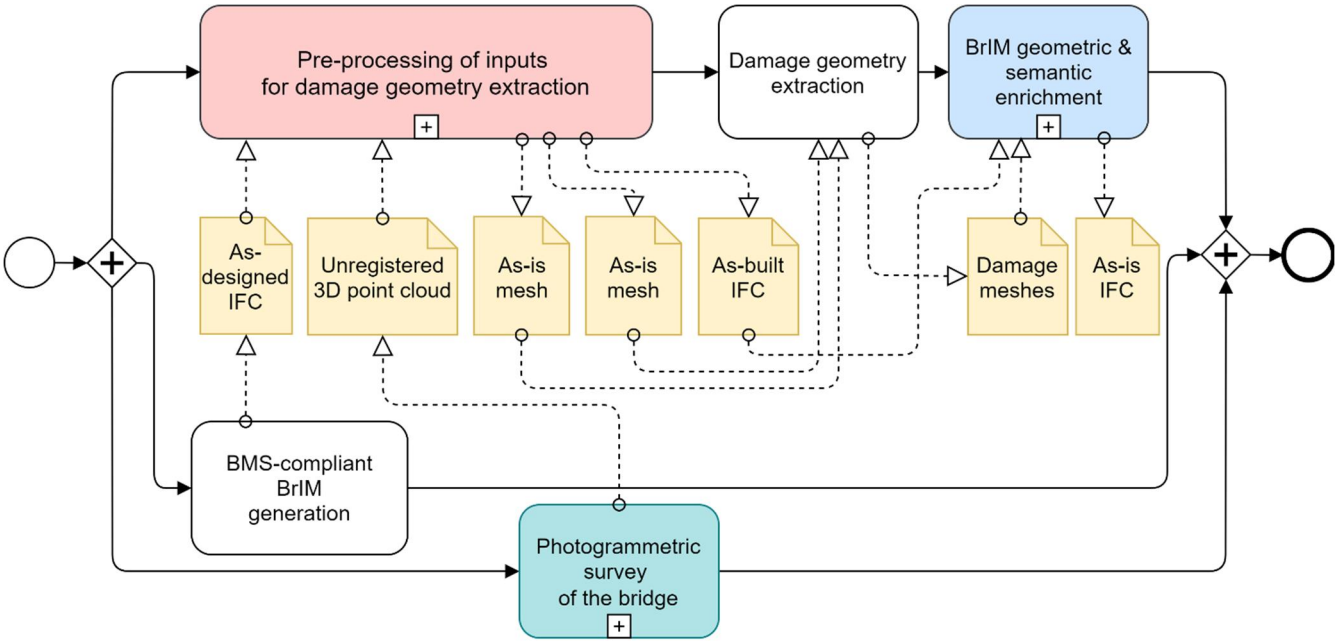


Figure 4.1. Process map for as-is BrIM generation.

4.2 Photogrammetric survey of a bridge

Photogrammetry, a technology used in this step, is highly developed. However, until recently, photogrammetry has been used mainly for mapping terrain, which is significantly different than generating 3D point clouds of solid objects. Capturing imagery of building structures and generating corresponding 3D point clouds has been investigated by many authors, such as Tuttas et al. [52], mainly for construction progress monitoring. Generating 3D point clouds of bridges, on the other hand, is a rather new field, which was almost uninvestigated at the start of this research. As opposed to compactly shaped buildings, bridges are much more demanding structures for capturing due to numerous reasons, some of which are:

- Lengthy uncondensed shape,
- Limited approachability by UAV,
- Unfavorable wind conditions in the bridge proximity,
- and Loss of GPS signal under the bridge.

The sub-process of generating 3D point cloud of the bridge, in the context of the proposed method for generating the as-is BrIM is shown in Figure 4.2.

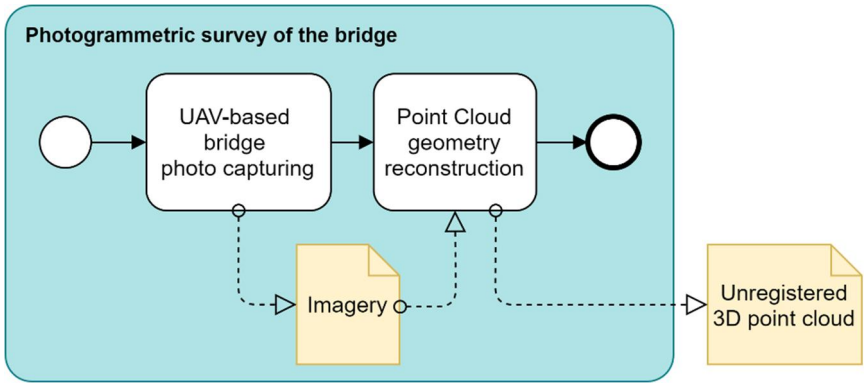


Figure 4.2. Map of the photogrammetric survey sub-process.

For the purpose of this research, an exhaustive investigation of all requirements for the photogrammetry application in the bridge inspection is performed, including series of field testing. In the following subsections, the results of this investigation are presented.

4.2.1 Bridge imagery capturing technique

Perhaps the essential prerequisite for the generation of a high quality 3D point clouds is an appropriate photo capturing. Based on recommendations from [78], and the experience gained through various tests, the basics of the capturing technique will be presented.

Photo capturing in the context of bridge inspection can be done using a digital camera with reasonably high resolution (5 MP or more), preferably with fixed lenses. If zoom lens is unavoidable, a focal length should be set either to maximal or to minimal value for more stable results. Using ultra-wide angle and fish-eye lenses is not desirable.

Images should be taken at maximal possible resolution. For sharp photos, an aperture should be reasonably high. Furthermore, a preferable shooting method is manual, otherwise the UAV speed should be slow enough to avoid blurriness.

UAVs are usually equipped with a GPS device. Therefore, each photo carries geo-information. This is important for completing the optional georeferencing task. However, when capturing without GPS information related to photos, as well as for achieving a more precise georeferencing and more accurate geometry of a point cloud, marker points should be placed on the object of interest (Figure 4.3). To be able to determine a highly precise scale of the generated model, marker points (at least two), with a known relative

distance to each other, should be placed on the object before shooting, so the points are later visible on images.

Figure 4.3. Marker point with precisely measured coordinates placed on a concrete column.

When shooting, each part of the bridge should be captured many times from all possible angles, thus eliminating, or at least minimizing so-called „blind zones“, which prevent the exact geometry reconstruction. Figure 4.4 shows a point cloud of the bridge, without and with camera positions.

(a)

(b)

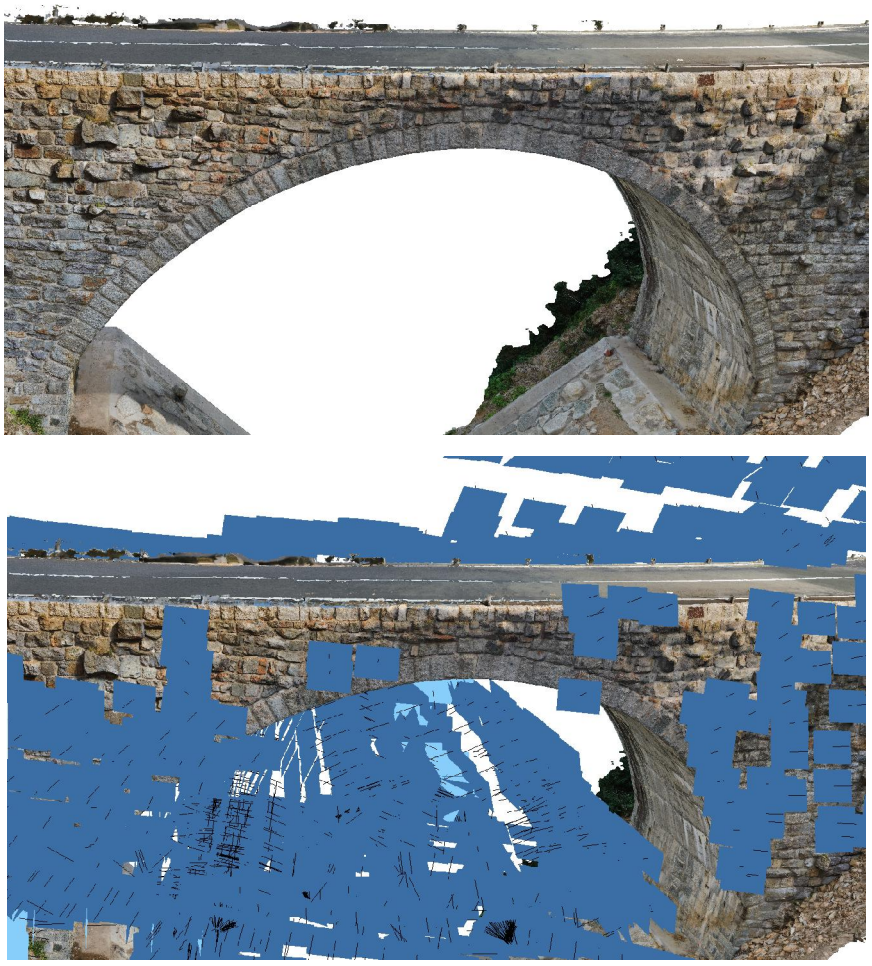


Figure 4.4. 3D point cloud of the bridge: (a) without camera positions, (b) with camera position and orientation (blue rectangle represents the camera sensor, whereas the sticking-out lines represent camera orientation).

All surfaces of the object should be covered by mutually overlapping photos. This overlap should be between 60% and 80% of the photo dimension. The biggest part of a photo

should be taken by the object of interest. If possible, a shooting should be done on cloudy days, without using flash, thus providing a uniform brightness.

4.2.2 Photogrammetric software solutions tests

In order to make a highly precise georeferenced triangular mesh of the bridge, three commercial solutions are tested: Bentley ContextCapture [79], Autodesk ReCap [80], and Agisoft Photoscan [81]. Although there are some differences in workflows of those three software, the basic principle is the same. After the imagery is imported, the software estimates camera position for each photo and generates a point cloud. Afterwards, a triangular mesh is generated, based on the point cloud data. The final step is adding texture to the mesh. As a test case, a cylindrical concrete column of the approach structure of the Pančevački bridge.

Figure 4.5. Test case: Pančevački bridge column.

4.2.2.1 Bentley ContextCapture

This is a photogrammetry tool developed by Bentley Systems. As one of the first reality capturing software solutions developed by a CAD company, there are numerous advantages regarding interoperability with other CAD programs. However, the ContextCapture does not lack space for improvements. In the following subsections, a general workflow of this software is presented, as well as its advantages and disadvantages.

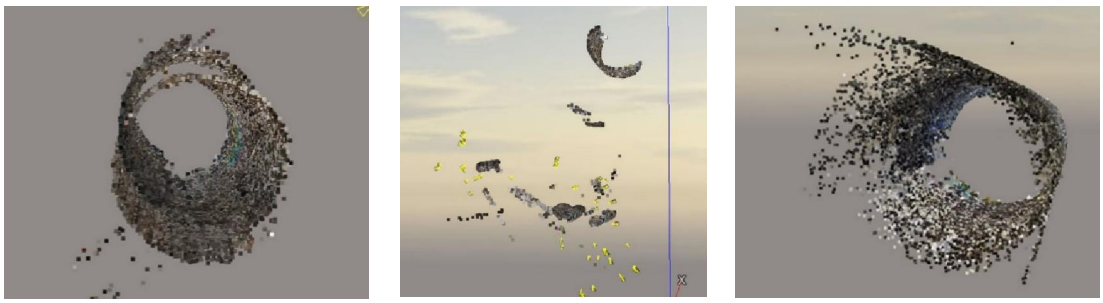
4.2.2.1.1 Loading imagery

Although this is a common step in every reality capturing software, the ContextCapture is special for the way it loads photos. Namely, the photos are automatically grouped in so called photogroups. One photogroup contains all photos with the same size of the camera

sensor, focal length of the lens, position of the principal point in the image plane and the distortion of the lens. All these characteristics of the photo are the input data for the camera position estimation step (Aerotriangulation) and they are automatically obtained from the camera. Although the idea to group photos with the same parameters is good, for some reason it does not always work. However, if photos are manually ungrouped, aerotriangulation is executed in good quality. The model can be georeferenced in this step. Points with the known coordinates (Control points) are selected on each photo. Apart from being a prerequisite for the georeferencing, control points carry a valuable information about the model geometry and, therefore, speed up the aerotriangulation step. Nevertheless, ContextCapture is very demanding regarding georeferencing. The software doesn't even notify about georeferencing success. The only way to find out about the unsuccessful georeferencing is to notice the software's inability to export the mesh to ".kml", a standard format for geographic data representation.

4.2.2.1.2 Aerotriangulation

This step is critical for model geometry assessment. Here, the camera position in the moment of photo capturing is calculated. The software calculates camera position with or without initial information such as GPS record or control points for georeferencing. Aerotriangulation is a well-known photogrammetric method, thoroughly explained by Tang et al. [82]. In ContextCapture, aerotriangulation doesn't always give satisfying results. Several settings were tested both on photogroups and ungrouped photos. Better results were achieved with ungrouped photos, so a few point clouds adhering to this approach are generated (Figure 4.6). Key point density for all models was set to high, whereas values of the rest of parameters remained default.



(a) (b) (c)

Figure 4.6. Point clouds for different aerotriangulation settings: (a) Positioning mode: automatic vertical (Points lie on two different cylinders), (b) Positioning mode: arbitrary (Points lie on randomly distributed cylinder fragments), (c) Position mode: Arbitrary, Georeferenced (All points lie on the same cylindrical surface).

4.2.2.1.3 Production

This step includes mesh generation and adding texture to it. According to the previous tests, outside of this research, for the quality of the mesh, a good quality of the point cloud is critical. When it comes to texture, resulting model quality depends on several factors, such as surface roughness and illumination. Namely, the rougher the surface, the

blurrier the texture (Figure 4.7a). Artifacts, such as the "border line" between two texture regions are noticed on the generated model (Figure 4.7b). This happens on spots where textures from two different photos converge. The reason might be improper mesh calculation or illuminance difference in two source photos. This is found inconvenient in terms of the concrete element condition assessment, i.e. the "border line" may be interpreted as a concrete crack.

(a) (b)

Figure 4.7. Texture errors: (a) Rough surface's blurry texture, (b) "border line".

Besides the listed problems, ContextCapture is fully interoperable with MicroStation [83], BIM software developed by Bentley Systems. Microstation offers a possibility to import a complete reality mesh in Bentley's ".3mx" format. As about interoperability with GIS software, reportedly it is possible to export reality mesh in ".kml" format. Having said that, the case model (Figure 4.8) export to ".kml" was not successful. The assumed reason is improper georeferencing.

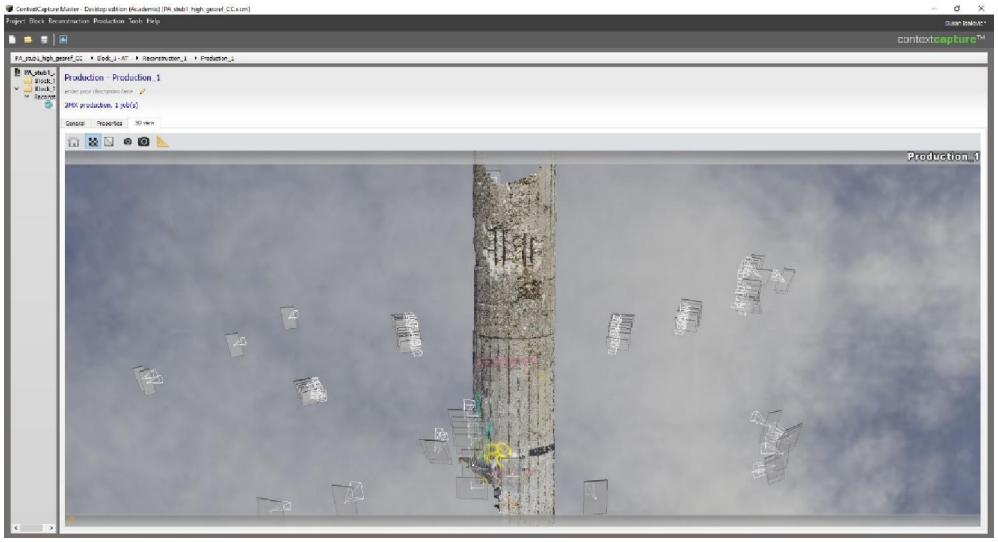


Figure 4.8. Concrete column georeferenced reality mesh generated by Bentley ContextCapture.

4.2.2.2 Autodesk ReCap

This is another solution developed by a CAD company and therefore it is interesting for this research, in terms of potential interoperability between the point cloud and the corresponding BIM. As opposed to ContextCapture, this software is cloud-based, offering none of the input parameters, so workflow analysis was impossible. Actually, the only choice ReCap offers is a mesh density. Available mesh generating modes are Preview and

Ultra. Since we have used an academic license, Ultra density of the mesh was not an option. Resulting reality mesh is shown in Figure 4.9. Since this is not even a complete model, texture quality is not further discussed.

Figure 4.9. Concrete column reality mesh generated by Autodesk ReCap.

Although ReCap is developed by Autodesk, its interoperability with Revit, Autodesk's BIM solution is not fully accomplished. The ReCap generated mesh cannot be imported into Revit. In fact, Revit accepts nothing but point cloud to be imported. ReCap does not support georeferencing of the model.

4.2.2.3 Agisoft PhotoScan

At the moment of the test, this was a software solution developed by Agisoft, company specialized for computer vision. Meanwhile, the same company developed a new software, MetaShape, heavily relying on the PhotoScan legacy. PhotoScan's robust workflow includes following steps:

4.2.2.3.1 Loading Imagery

This is a trivial step where all the photos of the model subject are being loaded into the program. As opposed to ContextCapture, PhotoScan doesn't make photogroups. This way, it is much easier to load imagery, without worries about photos characteristics such as focal length or orientation. When loaded, source images can be marked for georeferencing. Similar as in ContextCapture, points with previously determined coordinates need to be tagged on each photo where appeared.

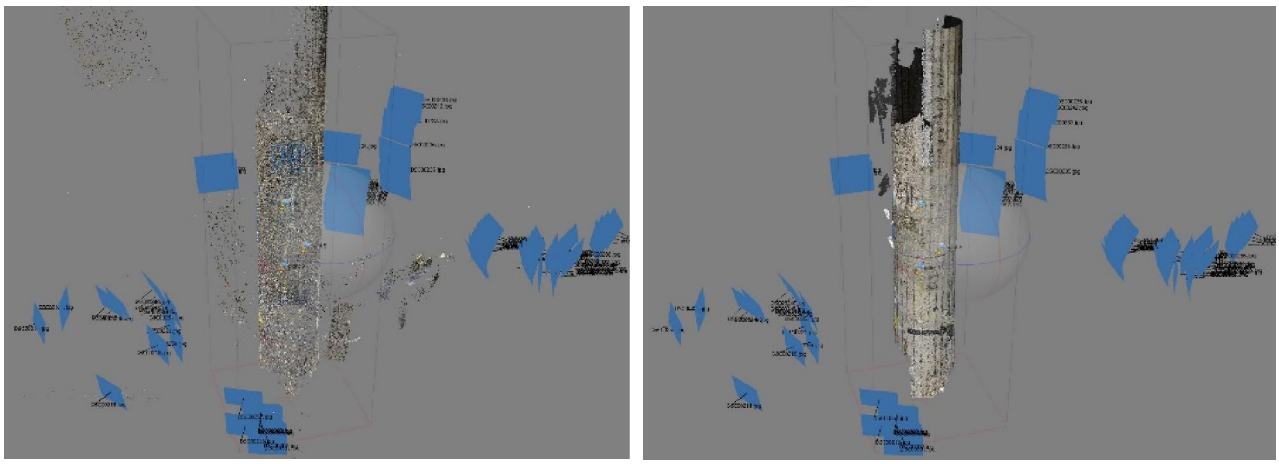
4.2.2.3.2 Photos Alignment

After loading source images, in this step, camera positions are being calculated. Additional tasks in this step are finding and matching the common points on photos. Moreover, the program refines camera calibration parameters. Outputs of this step are sparse point cloud and camera positions for each photo (Figure 5a). According to the previous generation of about ten models, outside of this research, this step runs perfectly with accuracy set to

high, which is found to give outputs with the optimal point cloud density/computation time ratio.

4.2.2.3.3 Dense point cloud generation

For dense point cloud generation PhotoScan use previously estimated camera positions. Sparse cloud is not used in this step. By calculating every point depth, this step is crucial for the model's final geometry. With the quality set to high, dense cloud is generated successfully in all of about ten different photo sets (Figure 5b).



(a) (b)

Figure 4.10. (a) Sparse point cloud, (b) dense point cloud. Camera positions are colored blue.

4.2.2.3.4 Generating mesh

In this step, software reconstructs a 3D triangular mesh. Starting point for mesh reconstruction can be sparse or dense point cloud, depending on user's choice. Additionally, PhotoScan offers two methods for mesh reconstruction based on surface type: Height Field - for planar surfaces, and Arbitrary - for any kind of object. Reconstructed mesh can be edited in the PhotoScan, or in a third party program. For the test case, a dense cloud is set as a source data and surface type is set to arbitrary.

4.2.2.3.5 Adding texture

After the mesh is generated, the texture can be added to the model (Figure 4.11). There are several texture mapping modes (Generic, Adaptive orthophoto, Orthophoto, Spherical, Single photo, and Keep uv) and several blending modes (Mosaic, Average, Max Intensity, Min Intensity, and Disabled). The test case settings were generic and mosaic. Although not as detailed as ContextCapture's, the texture achieved homogenous quality, without blurry places or "border lines". Interoperability with BIM software is not a PhotoScan's asset. The main reason are proprietary import formats. Nevertheless, point cloud created by this program can be imported in Autodesk Revit or Bentley Microstation.

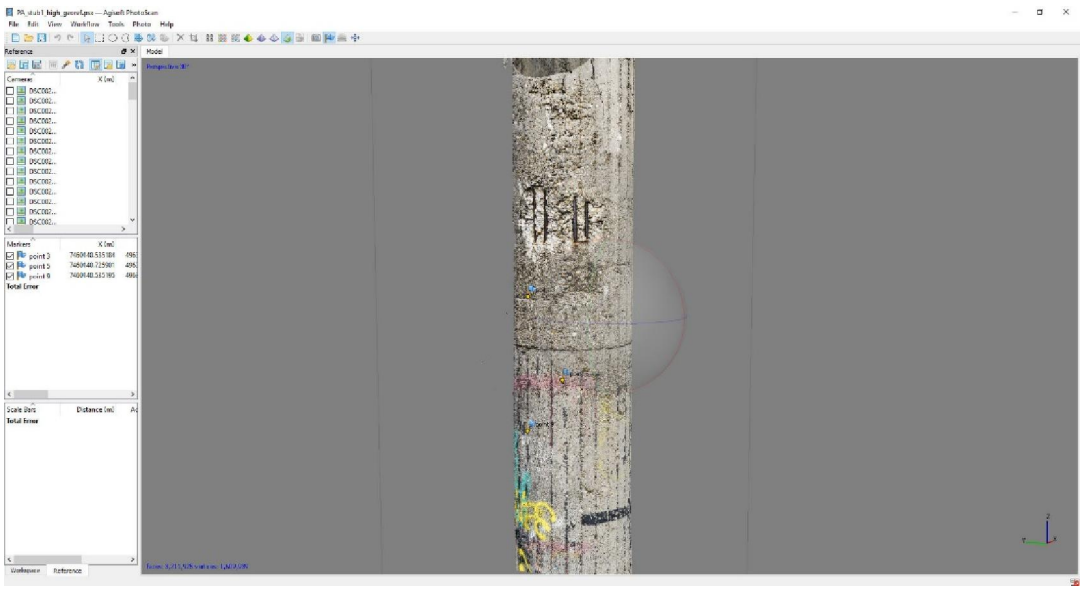


Figure 4.11. Concrete column georeferenced triangular mesh with texture, generated by Agisoft PhotoScan.

4.2.2.4 Tests conclusion

Among the tested solutions, two out of three are considerable. ReCap's not transparent workflow and low-quality outputs disqualify it for the use in the dissertation research. ContextCapture does not always run smoothly, which is mostly settings related. Besides, it gives a highly detailed texture which is blurry in some places and georeferencing can sometimes be a problem. ContextCapture's triangular mesh is fully compatible with Microstation. On the other hand, PhotoScan is a robust software solution running almost always flawlessly. It gives homogenous quality, but medium detailed texture. PhotoScan's point cloud is compatible both with Microstation and Revit.

4.3 Pre-processing of inputs for damage geometry extraction

Once the as-designed BrIM and a 3D point cloud of the inspected bridge are generated, they need to be further processed in terms of triangulation, alignment, and geometric adjustment to the as-built condition. The goal of this step is to fully prepare geometric inputs for the flawless extraction of geometric representations of damages. Figure 4.12 shows the map of this sub-process.

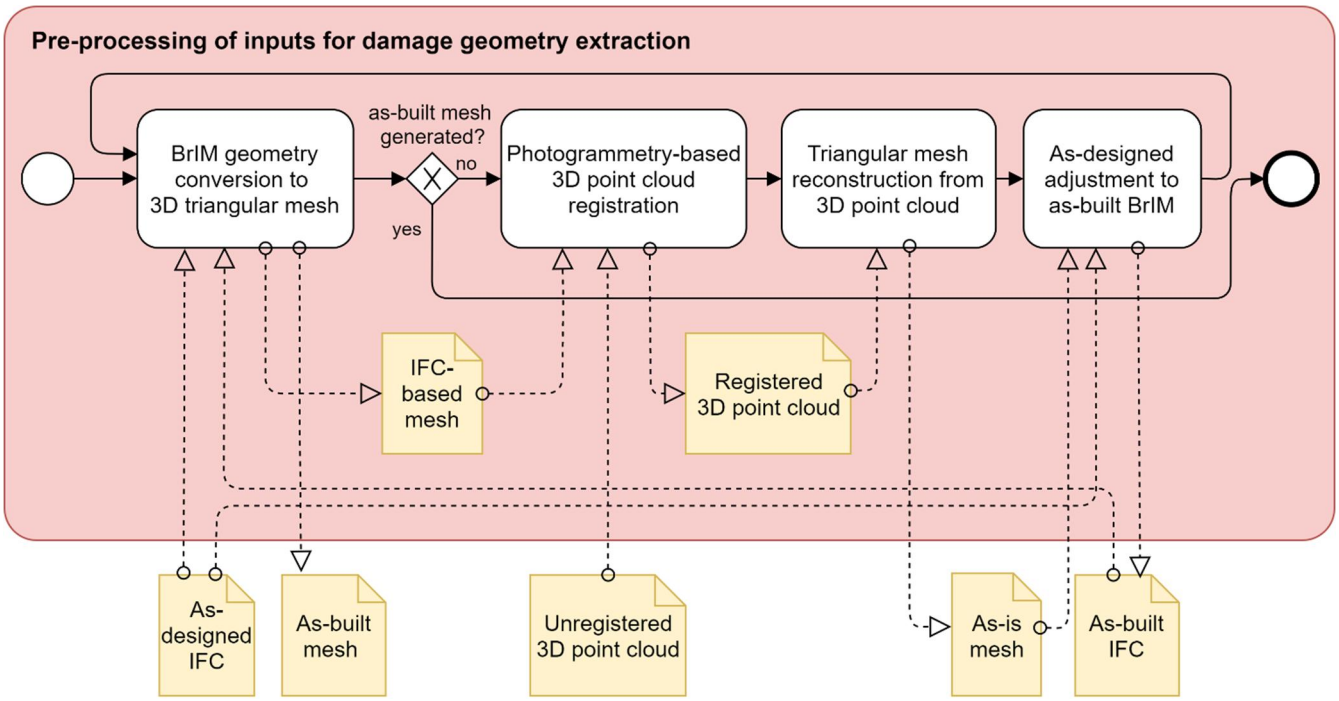


Figure 4.12. Map of the sub-process describing pre-processing of inputs for damage geometry extraction.

The first and the last step of this sub-process is BrIM geometry conversion to 3D triangular mesh. Namely, the as-designed IFC and as-built IFC are subjected to this step. IFCOpenShell library [84] is used to convert IFC geometry into triangular mesh. The proposed file format of both point cloud-based and IFC-based mesh is Wavefront OBJ. This format was chosen as it is text-based, easy to debug, supports indexed triangular meshes and is one of the most widely supported 3D file formats (due it its legacy use). In the following, the rest three steps of this sub-process are described in detail.

4.3.1 Photogrammetry-based 3D point cloud registration

As a prerequisite for all subsequent steps, the output of a photogrammetric survey of the bridge, 3D point cloud, should be registered to the geometry of previously generated BrIM. Registration is a computer graphics term and it stands for aligning two graphical objects (e.g. 3D point cloud, triangular mesh) by means of geometric transformations (i.e. translations and rotations) and scale adjustment. More precisely, registration aligns one object to another.

In this research, 3D point clouds are manually registered to the BrIM-based meshes using CloudCompare software [85]. CloudCompare offers several ways for the registration of two objects (i.e. two point clouds, point cloud and mesh, or two meshes), such as matching bounding-box centers, manual transformation, point pairs picking, or fine registration. Depending on the quality and shape of meshes, any of the listed alignment methods, as well as a combination of few of them, can be appropriate for the alignment task. Based on the experience aligning several bridge meshes to the corresponding BrIM geometries, combining the point pairs picking and fine registration is the most efficient approach.

Figure 4.13 shows the registration of a cylindrical column 3D point cloud to the swept solid representation of an *IfcColumn*. To achieve a clean depiction of the described step, the point cloud is cropped and the *IfcColumn* is shortened. In the first step, it is necessary to assign roles of Reference (static) and Aligned (dynamic) to each of two objects. Afterwards, at least three equivalent points need to be picked in both objects. These points are shown in Figure 4.13a (R and A in point labels refer to Reference and Aligned object respectively). At this step, CloudCompare computes the transformation matrix for the rigid transformation of Aligned object and the user executes it (Figure 4.13b).

Second step is the fine registration, applying the Iterative Closest Point (ICP) algorithm [86] on the two objects. Like in point pairs picking, firstly the roles to each object are assigned. Afterwards, the ICP algorithm iteratively re-computes the necessary transformation for the minimization of a distance from the Aligned to the Reference object (i.e. root mean square (RMS) of differences between the coordinates of the matched pairs). In this step, the scale of Aligned object is automatically adjusted. The result of a fine registration is shown in Figure 4.13c.

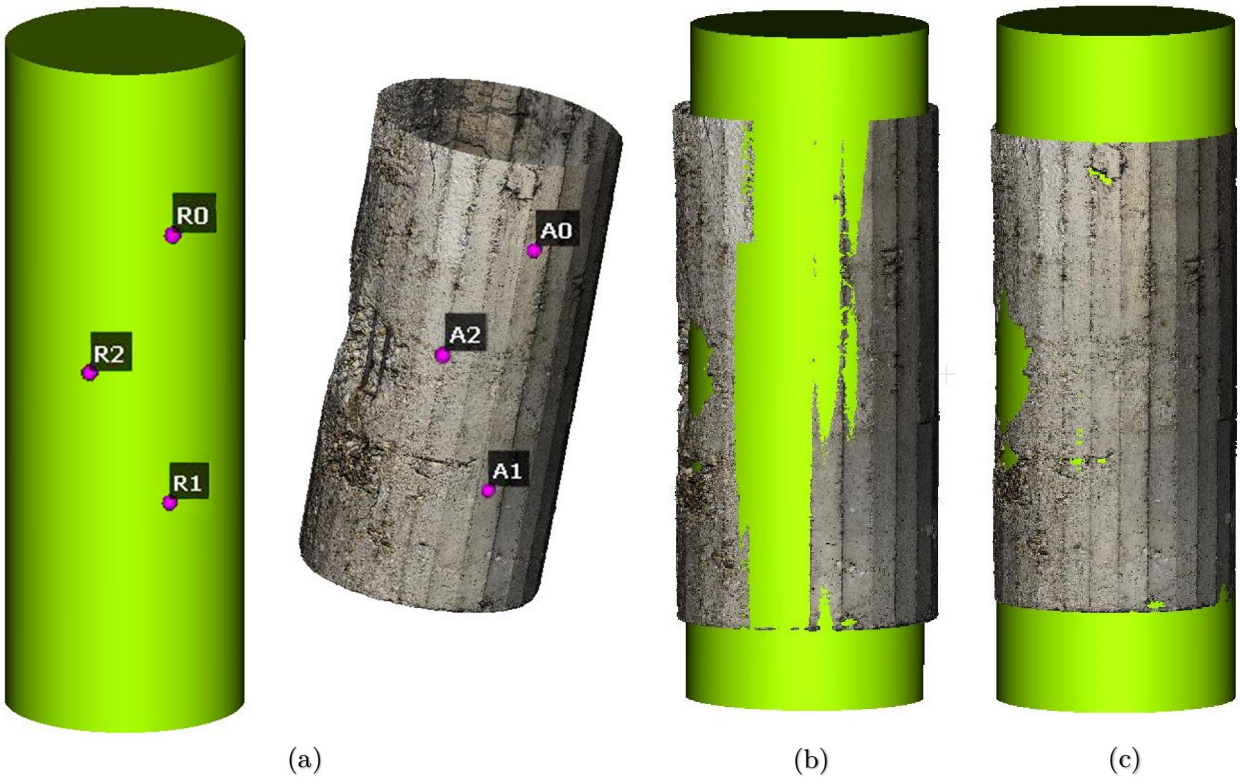


Figure 4.13. Swept solid representation of an IfcColumn and a cylindrical column 3D point cloud: (a) original state, (b) after point pair picking alignment, and (c) after a fine registration.

The upcoming step *As-designed adjustment to as-built BrIM* should sort out all major and minor deviations of the as-built geometry from the as-designed one. To be able to adjust geometry, the as-built point cloud needs to be registered to the as-designed BrIM geometry representation. Sometimes, however, the deviations (caused by damage, formwork imperfections, etc.) between these two are so significant that the registration is hardly possible. Although the exhausting and time-consuming manual registration of

separate coinciding parts of these two is possible, for the efficient use of the proposed method, a systematic solution for this problem needs to be developed.

4.3.2 Triangular mesh reconstruction from 3D point cloud

For damage detection both the as-built BrIM representation and the as-is 3D point cloud are required. Whereas the photogrammetric survey made a bridge 3D point cloud available, the as-built BrIM is still to be generated. The as-built BrIM represents the bridge at the time of completion. Although the geometry of the constructed bridge should comply with the designed one, this rarely happens in reality. Newly constructed concrete bridges contain various imperfections, mainly caused by the construction inaccuracy either due to slightly misplaced formwork, or the formwork deformation due to the weight of the fresh concrete. The settlement of the foundation can be also a less common cause. Before the as-designed IFC is adjusted to the as-built one, a triangular mesh is reconstructed based on the registered 3D point cloud.

Although all of the discussed software in Section 4.2.2 are capable of mesh reconstruction, due to the more transparent control of the process (i.e. choice of reconstruction algorithm), the MeshLab software tool [87] is used in this step. Among several geometric reconstruction algorithms, the following two are considered for application in this research: Poisson algorithm [88] and Ball Pivoting Algorithm (BPA) [89]. Whereas BPA can preserve the sharper edge features in the reconstructed geometry result, Poisson geometric reconstruction is suited for reconstructing organic shapes due to its nature to smoothen hard edges in its 3D shape approximation. Thus, both algorithms are used in reconstruction of differently shaped bridge elements.

To be able to perform CSG Boolean operations, both the as-built and the as-is mesh need to meet the following criteria:

- Each mesh needs to be watertight, i.e. is must consist of one closed surface,
- Each mesh must not self-intersect, i.e. it must have neither non-manifold edges, or non-manifold vertices.

Repairing a mesh to fulfil the listed conditions can be easily done by using any of the open-source computer vision software (e.g. MeshLab [87], or CloudCompare [85]). However, much more powerful software for the mesh repair is Geomagic Wrap [90], capable of detecting and repairing non-manifold edges, self-intersections, highly creased edges, spikes, holes, and tunnels. Since the test case is a cylindrical concrete column, besides small holes (i.e. mesh reconstruction side effects), two large cylinder needed to be filled. Figure 4.14 shows the result of automatic filling of holes by generating cover meshes.

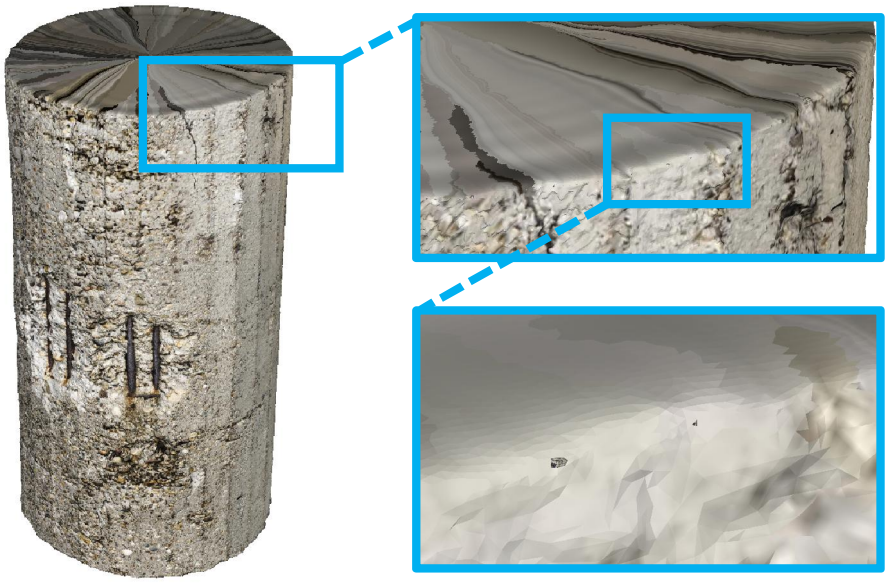


Figure 4.14. Automatically filled holes of a cylinder column by generating triangular cover meshes.

4.3.3 As-designed adjustment to as-built BrIM

The as-built geometry of the bridge is assessed by slicing the reconstructed photogrammetry-based mesh equidistantly in two orthogonal directions using MeshLab. Exported in the DXF format, the cross-sections are overlapped, and the centerline is manually fitted using Autodesk AutoCad. The cross-sections centerline represents the actual bridge contour in two orthogonal directions. The actual bridge dimensions are measured, and the BrIM is remodeled using Autodesk Revit [40]. Finally, the BrIM is exported to IFC format (as-built IFC). Figure 4.15 illustrate this step by showing the transversally sliced mesh of a cylindrical column, and a fitted centerline of the assessed as-built column contour.

When the centerline is fitted, the actual bridge dimensions can be measured. Finally, the BrIM needs to be remodeled according to updated bridge dimensions. Remodeling can be done manually, by any BIM modeling software. Nevertheless, only small number of BrIM elements usually need the dimension adjustment. Therefore, in order to save time of modeling from scratch, it is highly recommended to use the same software used for the initial model creation and only adjust the existing model. The use of parametric modeling can facilitate the adjustments. Whereas for solids with non-changing cross-sections, such as the presented cylindrical column, use of parametric modeling is almost needless, its full potential is unfolded when adjusting complex geometries. The remodeled BrIM is finally exported to the as-built IFC.

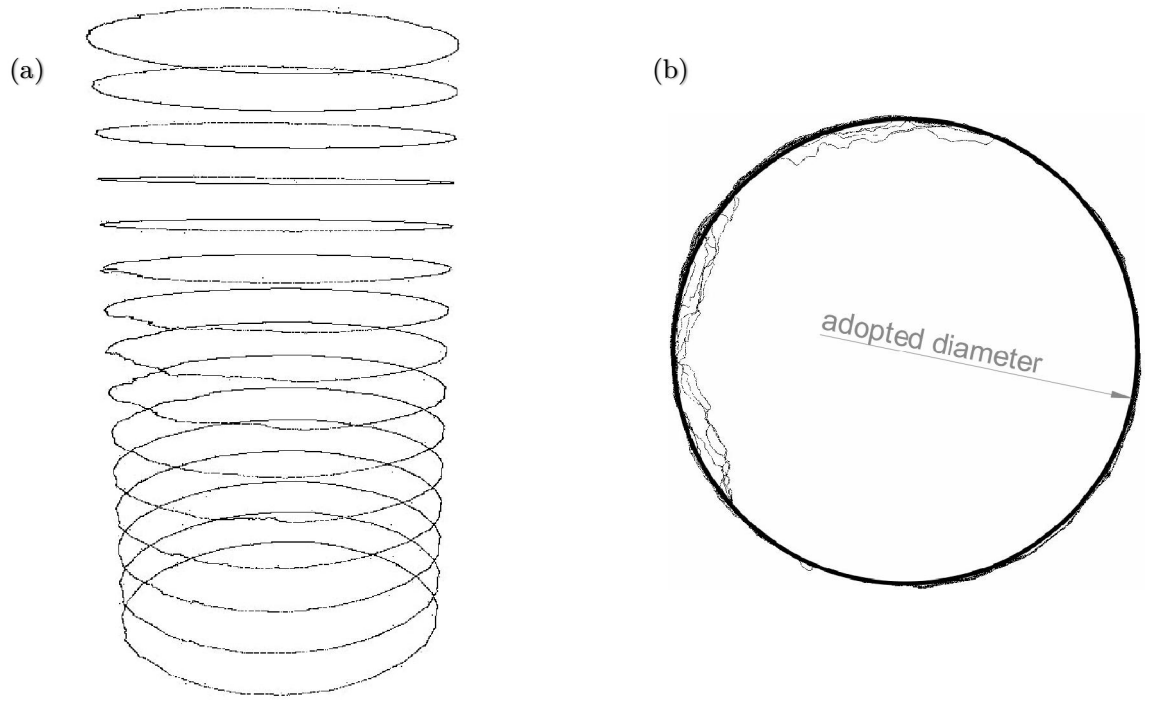


Figure 4.15. Transversal slices of a cylindrical column: (a) originally positioned slices, and (b) overlapped (continuous line represents the fitted centerline of an as-built column contour, and the dotted ones are overlapped sliced cross-sections of the as-is mesh).

4.4 Damage geometry extraction

After both the as-built and the as-is triangular mesh are fully aligned and post-processed to meet the criteria listed in Section 4.3.2, the CSG difference between these two can be finally computed. For this task, the MeshLab software [87] is used. MeshLab uses a Marching Intersections algorithm [91]. The algorithm re-computes meshes of two CSG objects, by discretizing their surfaces, according to the given unique discretization step. Afterwards, the CSG Boolean difference is computed with regard to the same discretization step. For this reason, the resulting mesh may have a "rasterized" look. However, it can be re-computed later again, in order to decrease the number of vertices and faces, due to the rendering optimization. In this second re-computation, the mesh can get the more "organic" look, since the only parameter for the repeated computation is the node number decrease ratio. The output of this process is a triangular mesh representing damage, in a Wavefront OBJ data format. Figure 4.16 shows the starting and ending point of the computation of CSG Boolean difference between the as-is photogrammetry-based and the as-built BrIM-based mesh.

(a)

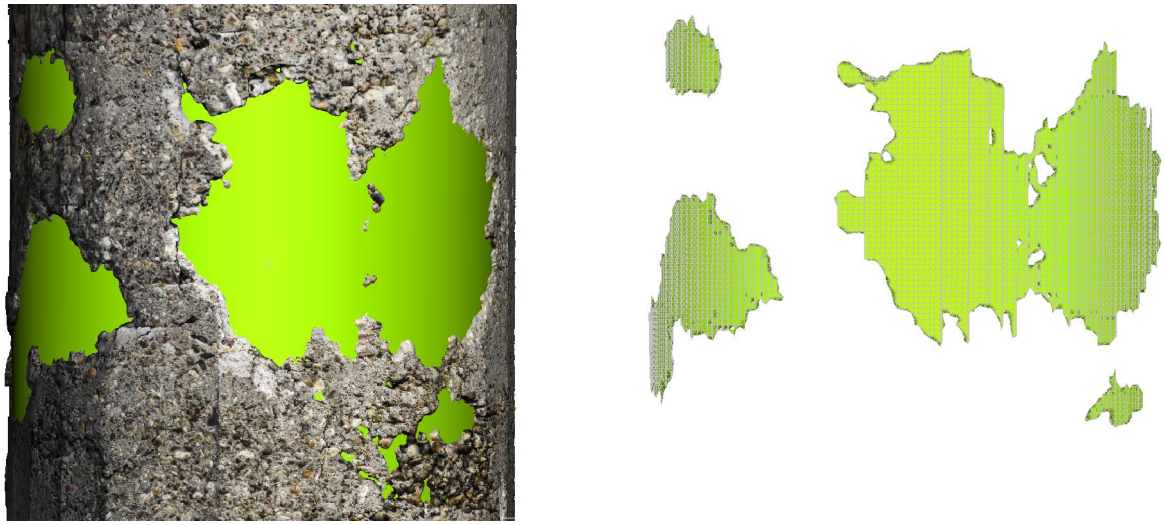


Figure 4.16. CSG Boolean difference operation: (a) the as-built BrIM-based mesh (yellow) volume subtraction from the as-is photogrammetry-based mesh (textured concrete), and (b) the resulting damage mesh.

4.5 BrIM geometric & semantic enrichment

Each extracted mesh corresponds to the identified damage. In this step the BMS catalog is consulted in order to associate each detected damage with a corresponding damage type from the catalog. Once each mesh is associated with a single or multiple catalog types, a prototypical software tool was developed and used to enrich the as-built IFC with damage data. Figure 4.17 shows the process map of this step.

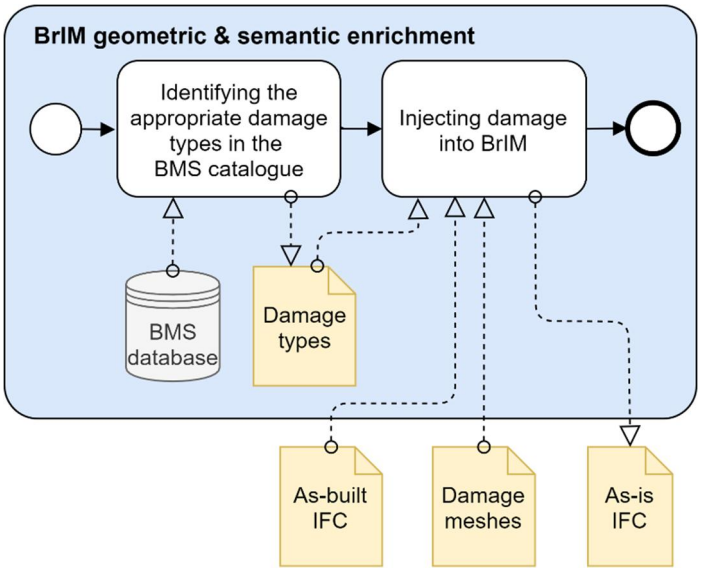


Figure 4.17. Map of the BrIM geometric & semantic enrichment sub-process.

The software tool is implemented using the IFCEngine [92]. This is a STEP toolbox written in C++, and distributed as a Dynamic Link Library (DLL), and is interfaced using the late binding technique. Late binding uses the standard data access interface

(SDAI), an application programming interface (API) with a set of functions and methods for reading and writing STEP files.

To insert the damage geometry into IFC, an OBJ parser is coded. The parser reads vertices and faces data from an OBJ file and writes it into the IFC, as *IfcCatresianPointList3D* and *IfcTriangulatedFaceSet*. The final bridge geometry is a result of CSG difference between the as-built IFC and damage geometry, implemented through the *IfcRelVoidsElement* relationship between damage object (*IfcSurfaceFeature*) and damaged bridge element (*IfcElement*). For the model visualization, the existing IFC viewer developed by RDF is embedded into the prototype software. The software is written in C++, using the Microsoft Foundation Class library (MFC).

Additional features are added to the viewer, such as adding damages interactively. Two ways of adding damage objects are implemented. In the first one, the main damage object (*IfcSurfaceFeature*) in the damage group is associated with the bridge element (*IfcElement*) and the source of its geometric representation (OBJ mesh file) is selected through the browse dialog box (Figure 4.18b). The second way to add damage is by selecting Add sample damage. In this case, the damage object (*IfcSurfaceFeature*) without geometric representation is created. It represents other identified damage types related to the main damage type whose geometric representation is added. Due to the large size of BMS catalog, the damage hierarchyCode and type name are entered through text boxes.

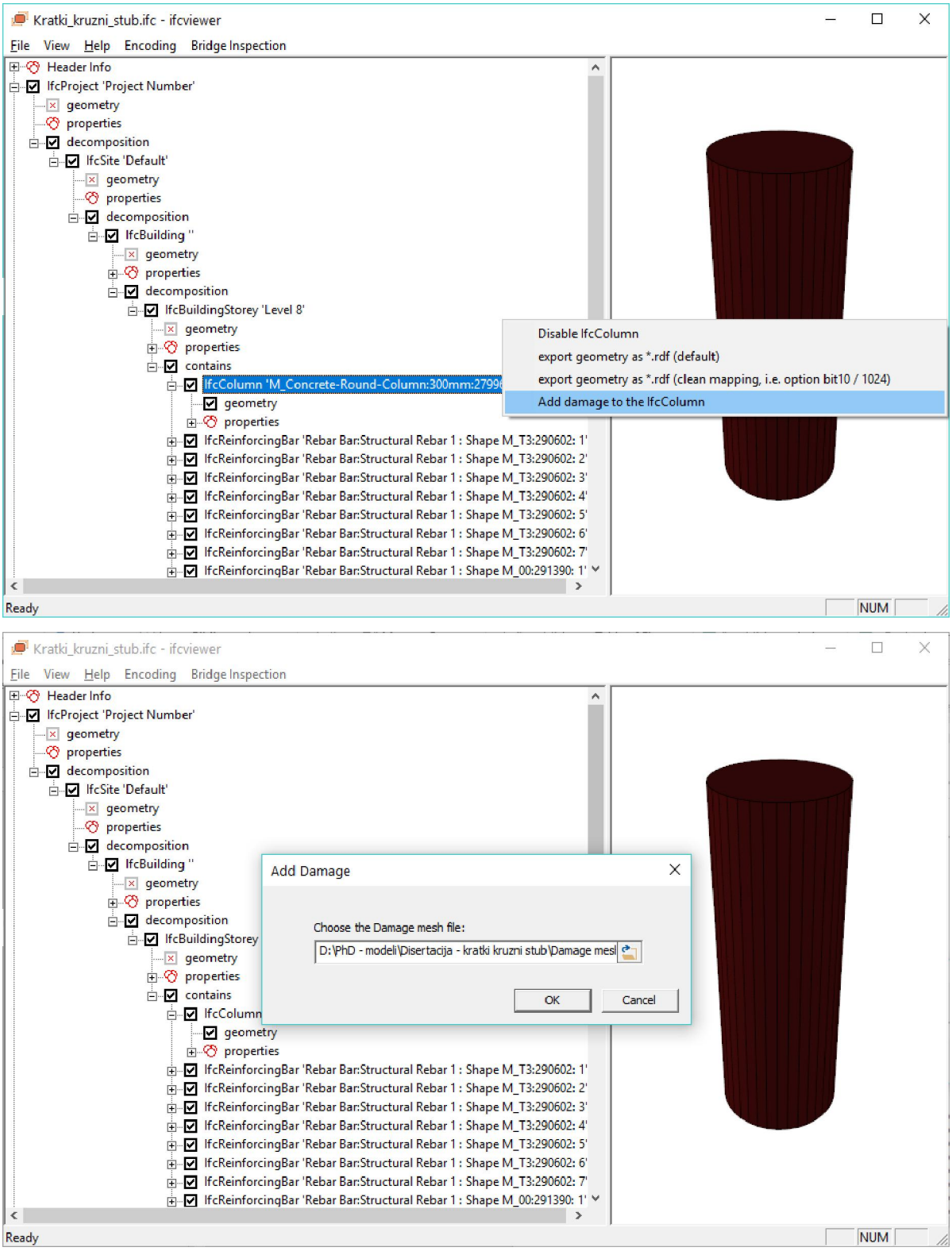
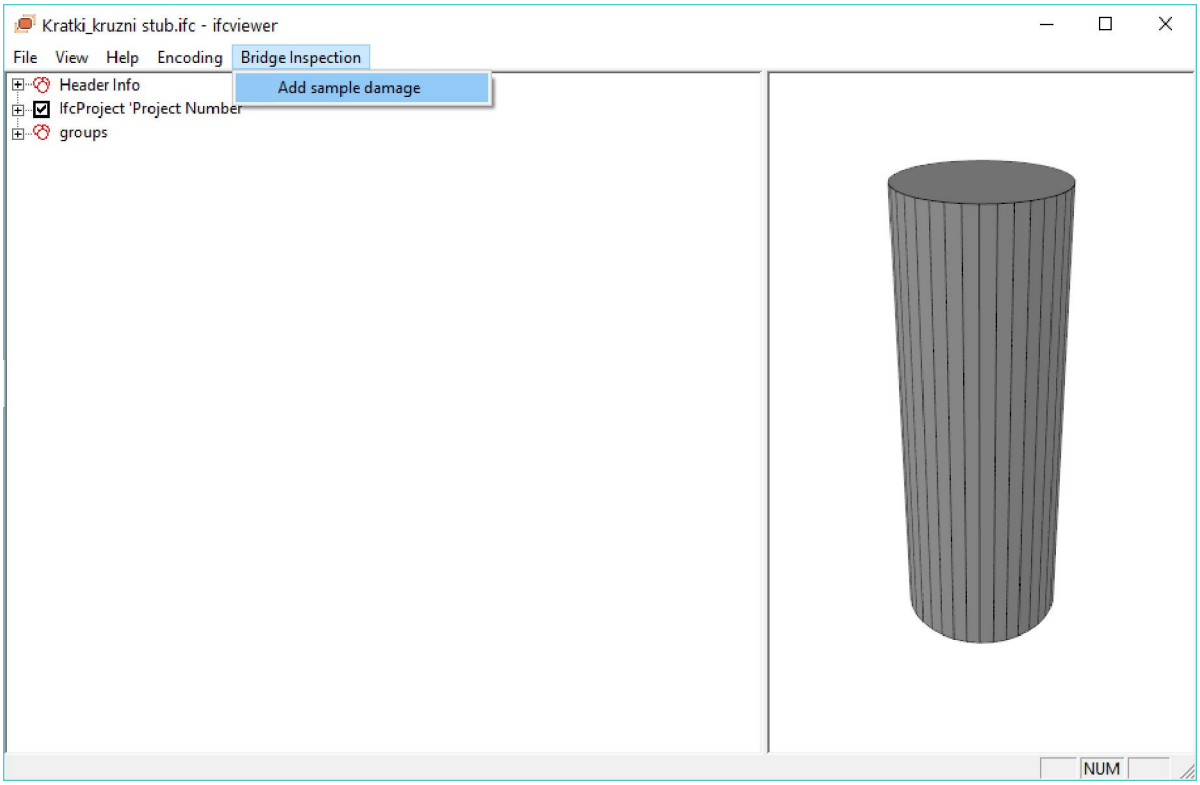


Figure 4.18. Prototype software with embedded viewer: (a) selection of an element "to be damaged", (b) browsing the damage mesh file.

(a)



(b)

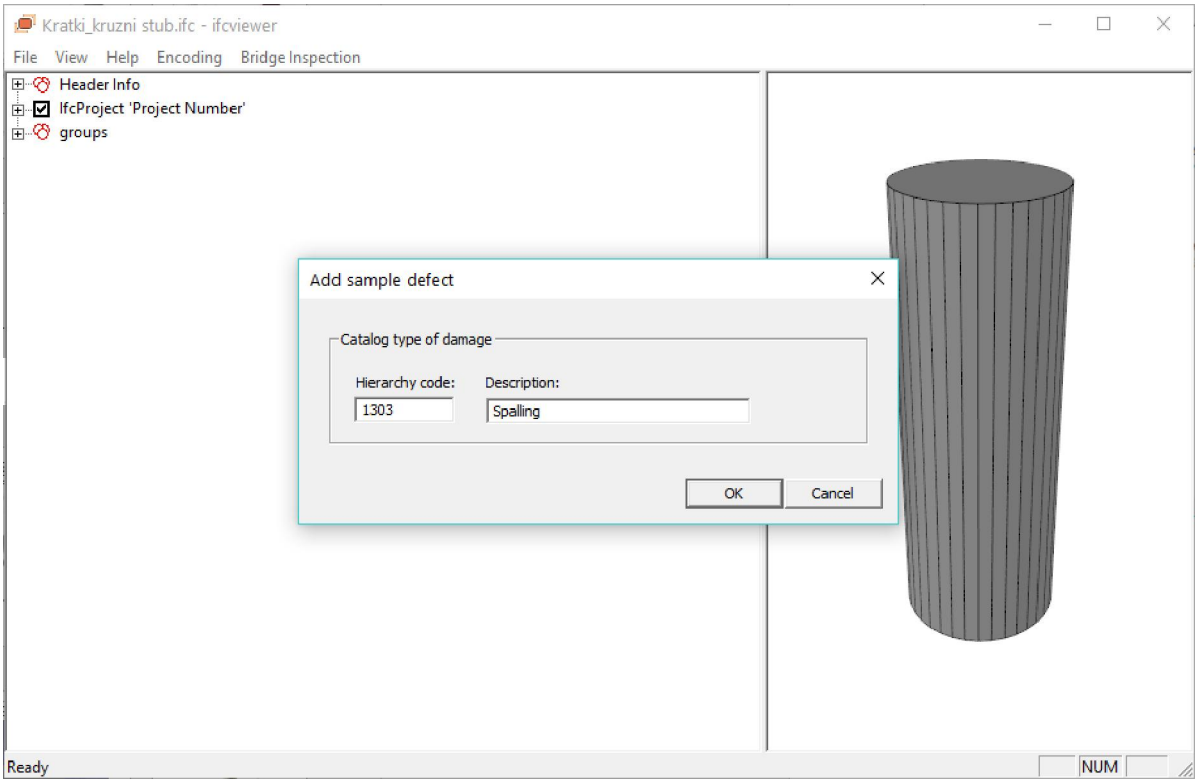


Figure 4.19. Prototype software with embedded viewer: (a) drop-down menu for adding sample damage, (b) dialog box for associating damage instance with the corresponding catalog type.

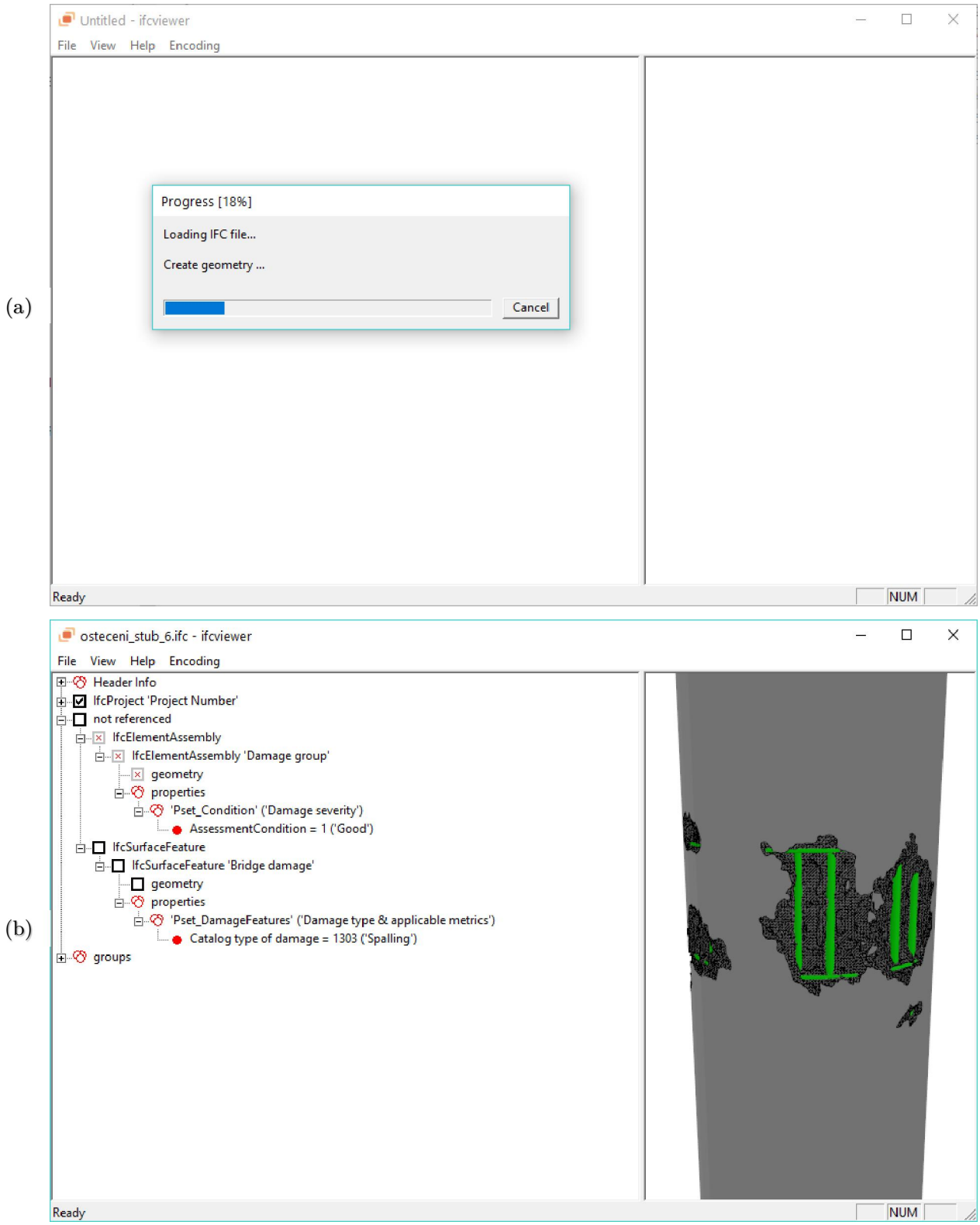


Figure 4.20. Prototype software with embedded viewer: (a) generating BrIM geometry (i.e. computing CSG operation result), (b) final as-is BrIM.

5 CASE STUDY

5.1 Bridge description

The proposed method is applied for a photogrammetrically acquired point cloud of a bridge over the river Gročica, located in the Grocka municipality of the city of Belgrade, Serbia (Figure 5.1). It is a 12.5 meter spanned simply supported double girder bridge built in the 1930s.

Figure 5.1. Case study: Bridge over river Gročica, located in Grocka municipality of the city of Belgrade, Serbia. (a) Aerial photograph showing the location of the bridge, and (b) Perspective view of the bridge.

Neglect in addition to an inappropriate designed and poorly maintained drainage system has caused large spallings, accompanied by extensively corroded reinforcement on girders, abutments, and curbs. Detected damage corresponds to the following damage types from KUBA catalog [14]:

- Cracks due to reinforcement corrosion
- Spalling
- Chipped off patched spots
- Fractured reinforcement
- Chipping-missing pieces
- Loss of chippings
- Slightly corroded Reinforcement
- Strongly rusted reinforcement

General information about the road section where the bridge is located:

- Road Section ID: 0337
- Road Section Name: 0404 Becarevo Brdo – 0405 Grocka
- Road Section Length: 9.557 km

General information about the bridge:

- Chainage at the middle of the bridge: 5+124 km

- Spanned structure length: 12,50 m
- Area of the bridge with access roads: 284,41 m2
- Area of the bridge: 133,75 m²
- Bridge ID: 14818

5.1.1 Bridge management data

The last regular inspection report from 2006 labeled bridge condition as unfavorable. In 2013, the load carrying capacity analysis of the bridge is conducted. Although the analysis recommended comprehensive reconstruction, the preliminary visit in the beginning of August 2018 showed an extremely bad condition of the structural elements. The conducted reconstruction mainly focused on the roadway, neglecting both the substructure and the superstructure of the bridge. This is a relatively short bridge with two traffic lanes, located close to Belgrade. With an extensive, visually noticeable damages, this bridge seemed perfect for this case study.

This bridge is publicly owned and maintained by the Serbian NRA: Roads of Serbia (RoS). RoS manages bridges using the inventory and inspection database. As the first step of establishing the comprehensive BMS, back in 1990, the RoS's predecessor - Road Directorate of Serbia, created a bridge inventory and inspection database named BPM (in Serbian: Baza Podataka o Mostovima) [93]. Unfortunately, the BMS has not been created ever since.

The inspection data is addressing four groups of elements:

- Load-carrying elements
- Pavement, waterproofing and expansion joints
- Traffic indicators
- Bridge equipment

Condition rating of the entire structure is obtained as a weighted sum of the condition ratings of all the bridge elements. During the inspection, each group is evaluated. The evaluation scale depends on the group. Usually, the scale is consisted of six condition ratings (Good, Satisfactory, Unfavorable, Poor, Serious and Critical). The inventory of the Bridge over river Grocka is presented in Appendix 1. Whereas A1.2 contains information about a bridge elements used in the description and elements to be inspected, in the Appendix 2, the data from the inspection report is shown.

5.2 As-designed BrIM Modeling

Following the proposed process map, scanned paper drawings of the bridge (Figure 5.3), provided by RoS, are used as geometry source to create the as-designed BrIM. The bridge

is modeled using Autodesk Revit [94]. The model complies with the LOD 350, according to [19]. Girders, railings, roadsides, and asphalt cover are modeled using existing Revit families, whereas abutments and deck are modeled as in-place structural framing components. Finally, the model is exported in IFC format. Due to the lack of bridge-related IFC entities, abutments are exported as instances of *IfcWall*, deck is represented by *IfcSlab*, whereas girders are exported as instances of *IfcBeam*. The resulting as-designed IFC is shown in Figure 5.3b.

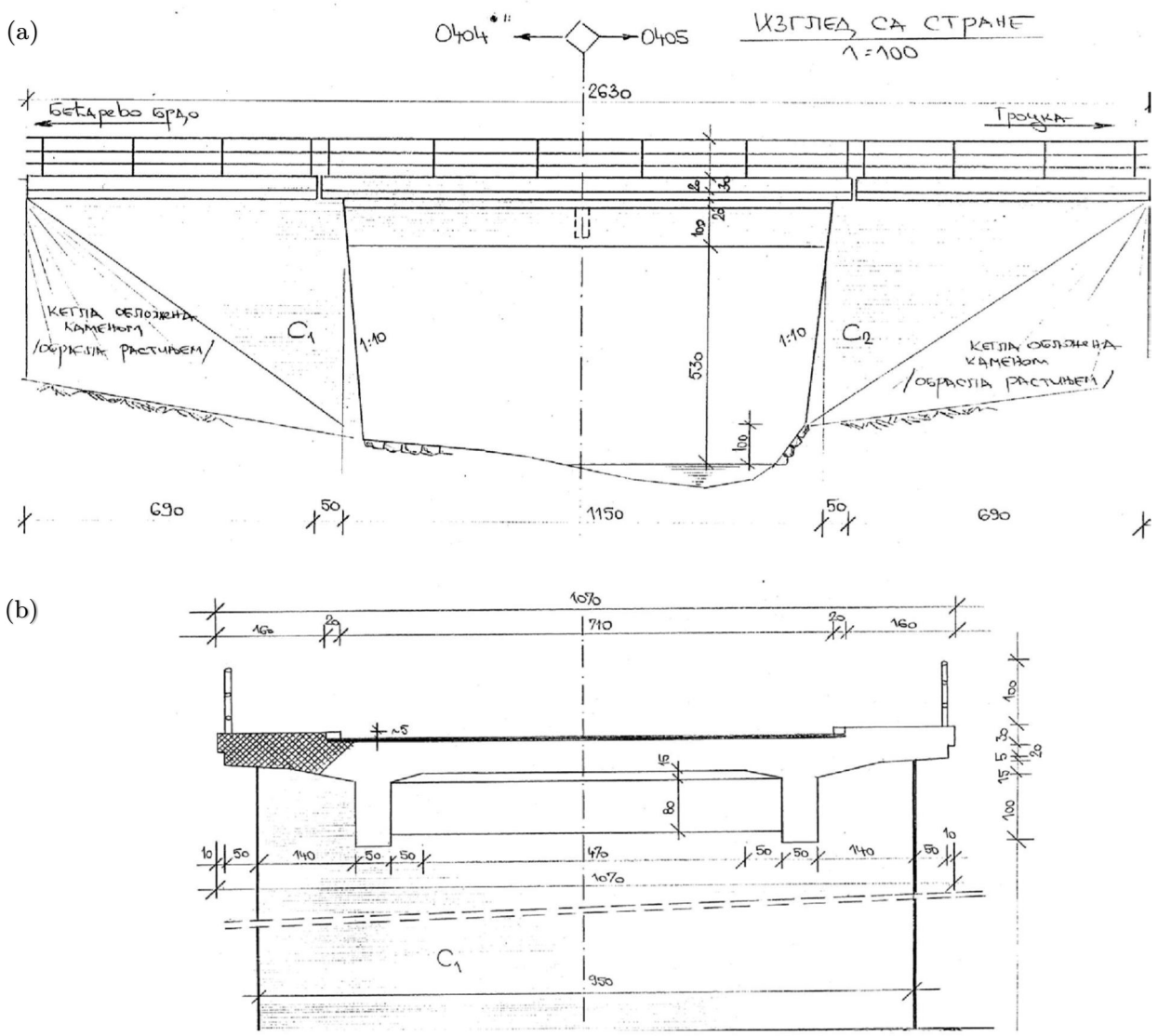


Figure 5.2. Scanned paper drawings of the bridge: (a) side view, and (b) cross-section.

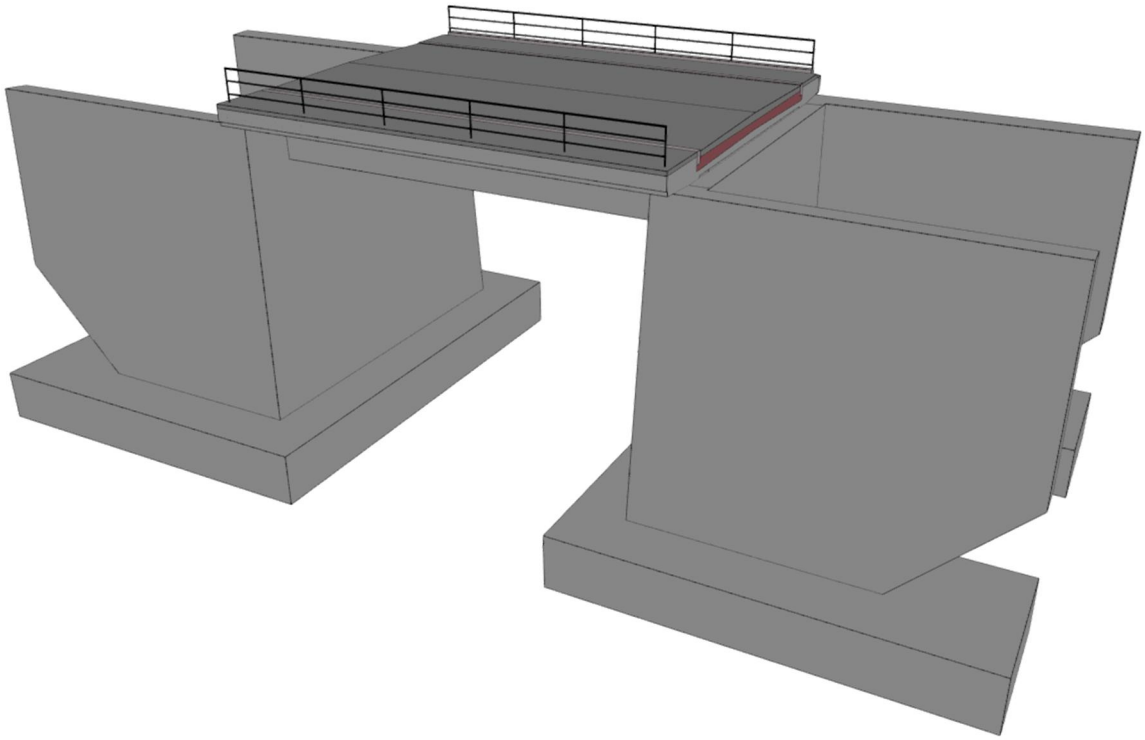


Figure 5.3. As-designed BrIM.

5.3 Photogrammetric survey

5.3.1 UAV-based bridge photo capturing

The survey has been performed on August 10th and August 16th, 2018. On August 10th, in the period between 10:00 and 15:00, the bridge is photographed by a UAV equipped with camera. Weather condition parameters at the bridge location on the day of survey are listed in Table 5.1. Detail solar data (e.g. Sun azimuth and altitude, shadow disposition, etc.) important for the quality of the imagery is shown in Appendix 3.

Table 5.1. Weather in Grocka (retrieved from [95])

	August 10th (10:00 ÷ 15:00)	**August 16th (7:00 ÷ 8:00)**	**Unit**
Average air temperature	30	19	°C
Precipitation	0	0	mm
Average wind speed	26	7	km/h
Wind direction	SE, ESE	W	
Average air pressure	1008	1004	mbar
Average air humidity	42	94	%

The chosen UAV was DJI's Phantom 4 Pro (Figure 5.4). Thanks to its ability to turn camera up to 30º relative to horizon and a high-resolution camera, which allows taking

photos from under the bridge, this drone happened to have the highest performance/cost ratio among other commercial UAVs. Some basic characteristics of the drone, important for the photogrammetric survey are listed in the following, while the detail specification of Phantom 4 Pro is available at [96].

- Camera sensor: 1" CMOS, 20 megapixels
- Camera lens: FOV 84° 8.8 mm/24 mm (35 mm format equivalent) f/2.8 - f/11 auto focus at 1 m - ∞
- Gimbal Controllable Range: Pitch from -90° to +30°

Figure 5.4. Phantom 4 Pro (retrieved from [97])

In order to avoid blurriness and provide sufficient photo overlapping, shooting was manual. Except from driveway, all the bridge elements were photographed from the approximate distance of 3 meters. Because of the active traffic and the presence of electric cables above the bridge, photos of the driveway are captured from the 20-meter distance. In total, 590 photos (5472 x 3648 pixels) are captured. Focal length has been fixed to 9 mm, while the exposure has been manually adjusted for each photo. No flash is used. The preliminary checkup pointed to the unfavorable sun altitude (shown in Appendix 3), which caused oversaturation on 32 photos (Figure 5.5a). After the removal of the oversaturated ones, 558 photos of satisfying quality left for 3D reconstruction.

(a) (b)

Figure 5.5. (a) Oversaturated photo of the bridge, (b) water reflection on the bridge deck.

For 3D point cloud reconstruction, the Agisoft PhotoScan is used. During the generation of a dense point cloud, particular inconsistencies of the model occurred. The water reflection under the bridge (Figure 5.5b) and the insufficient imagery of the lateral sides of the cross girder resulted in an inaccurate geometry of the cross girder, as well as the uneven surfaces of the deck and the main girder (Figure 5.6). Since the cause of the model inconsistency was the lack of geometric information in the source imagery, the second survey was inevitable.

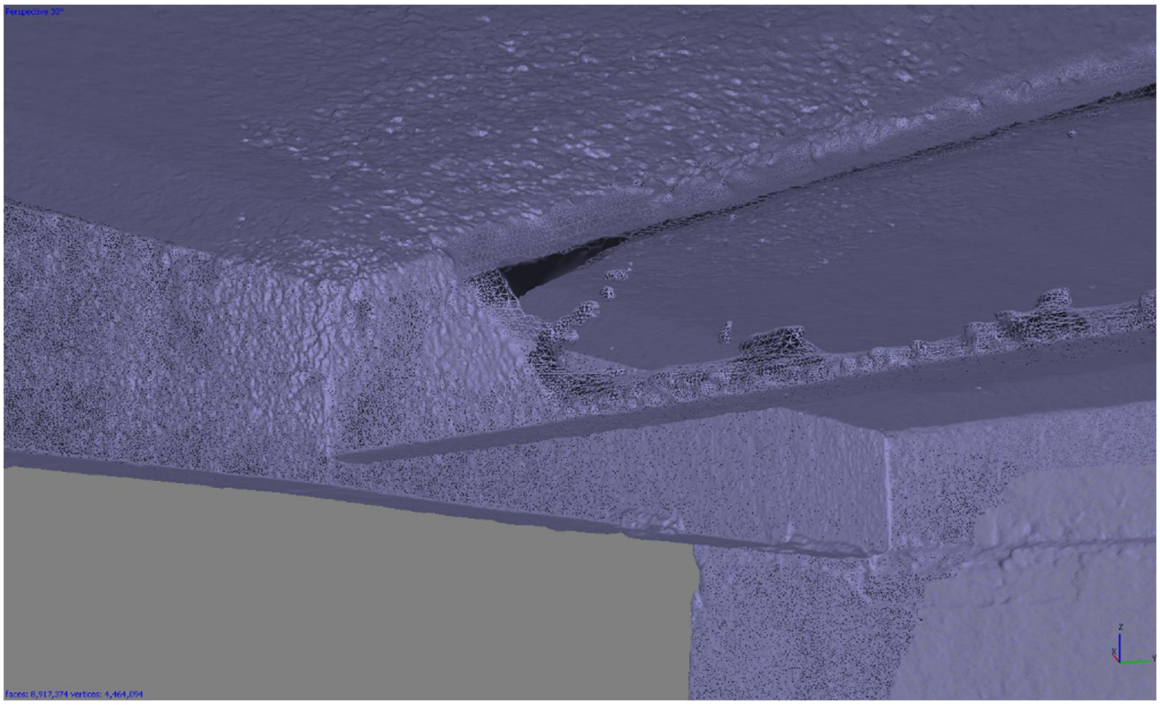


Figure 5.6. Misinterpreted cross girder geometry by Agisoft PhotoScan.

On the August 16th, 2018, a repeated survey is performed. The goal of this survey was to extensively capture the elements under the bridge. To avoid the water reflection, the bridge is photographed early in the morning, between 7:00 and 8:00, when the entire area under the bridge is in the shadow (Appendix 3). This time, a manual camera with a 35 mm lens is used. The reason for using manual camera is low UAV approachability to the area under the bridge. Basic characteristics of the equipment is listed below. The total of 876 (6000 x 4000 pixels) photos is captured. The exposure has been adjusted both manually and by auto mode, depending on the lighting conditions for each image. No flash is used.

- Camera: Nikon D5300
- Sensor: 24 megapixels
- Lens: AF-S DX Nikkor 35mm f/1.8G

5.3.2 3D point cloud and triangular mesh reconstruction

This task is accomplished by using Agisoft Photoscan 1.4.3 [81]. Since the source imagery is acquired by combining two different cameras, two independent models are generated and later merged into one. It took about two days to complete the reconstruction using PC with the characteristics listed below.

- CPU: AMD FX 8350
- GPU: ASUS STRIX-GTX980
- Memory: 32GB

The final outputs of this task are:

- Point Cloud (30.708.690 points)
- Triangulated mesh (14.999.999 faces, 7.504.065 vertices) + Texture

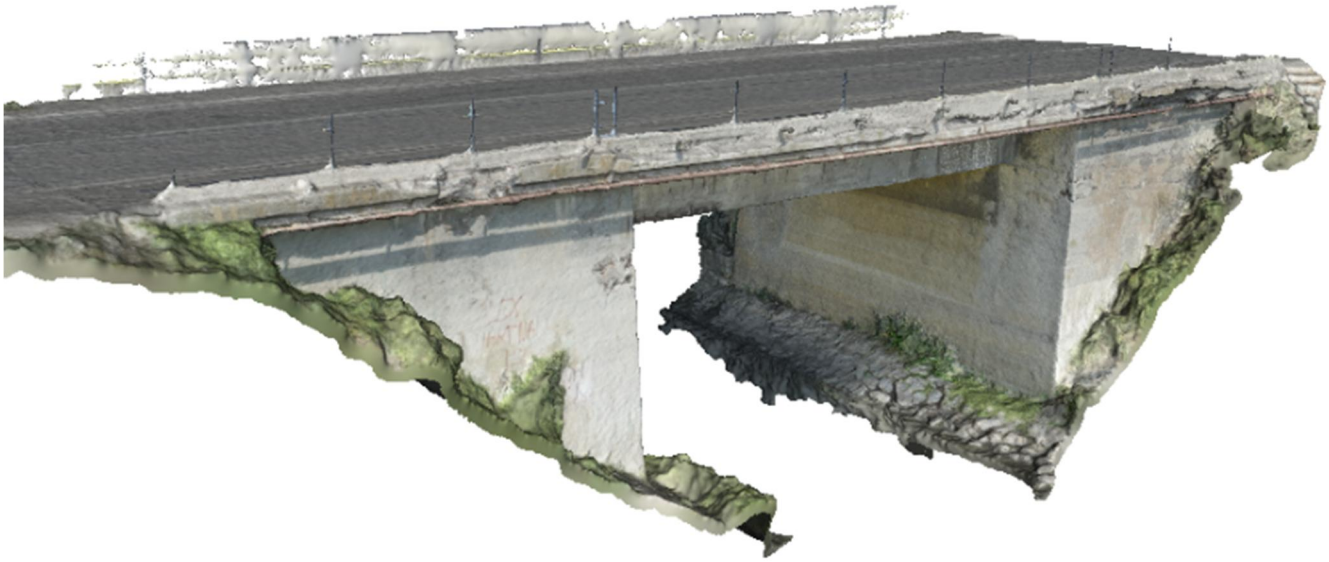


Figure 5.7. The result of 3D Scene Reconstruction: triangulated mesh with texture.

In the following, a short description of the Photoscan workflow and used settings are presented.

5.3.2.1 Photo Alignment

After loading the source imagery, in this step the common points on different photos are recognized and matched. In case of using UAV, each photo contains information on GPS positions of the drone and gimbal azimuth and altitude. Otherwise, the common points on different photos are recognized and matched. The program then refines camera calibration parameters. The outputs are sparse Point Cloud and camera position for each photo. Figure 5.8a depicts the result of this step.

Used settings: Highest accuracy, Pair preselection Disabled

5.3.2.2 Building dense point cloud

To generate dense Point Cloud, Photoscan uses previously estimated camera positions. Sparse Point Cloud is not used in this step. By calculating every point depth, this step is crucial for the model's final geometry. Figure 5.8b shows the dense cloud generated for this case study.

Used settings: High quality

5.3.2.3 Building mesh

In this step, program reconstructs a 3D triangulated mesh using the points from the point cloud as the mesh vertices. Starting point for mesh reconstruction can be sparse or dense point cloud, depending on user's choice. Additionally, Photoscan offers two methods for mesh reconstruction based on surface type: Height Field (for planar surfaces) and Arbitrary (for any shape). The user can edit the mesh after reconstruction in the Photoscan environment. Figure 5.8c shows the mesh of the case bridge.

Used settings: Arbitrary surface type, Dense Point Cloud as a source, Medium face count

5.3.2.4 Adding texture

After the mesh is generated, the texture can be added to the model. There are several texture mapping modes (Generic, Adaptive Orthophoto, Orthophoto, Spherical, Single photo, Keep uv) and several blending modes (Mosaic, Average, Max Intensity, Min Intensity, Disabled). Settings for our case study were generic and mosaic. Figure 5.8d depicts the textured mesh of a Bridge over river Grocka.

Used settings: Generic mapping mode, Mosaic blending mode, 4096 x 1 Texture size/count

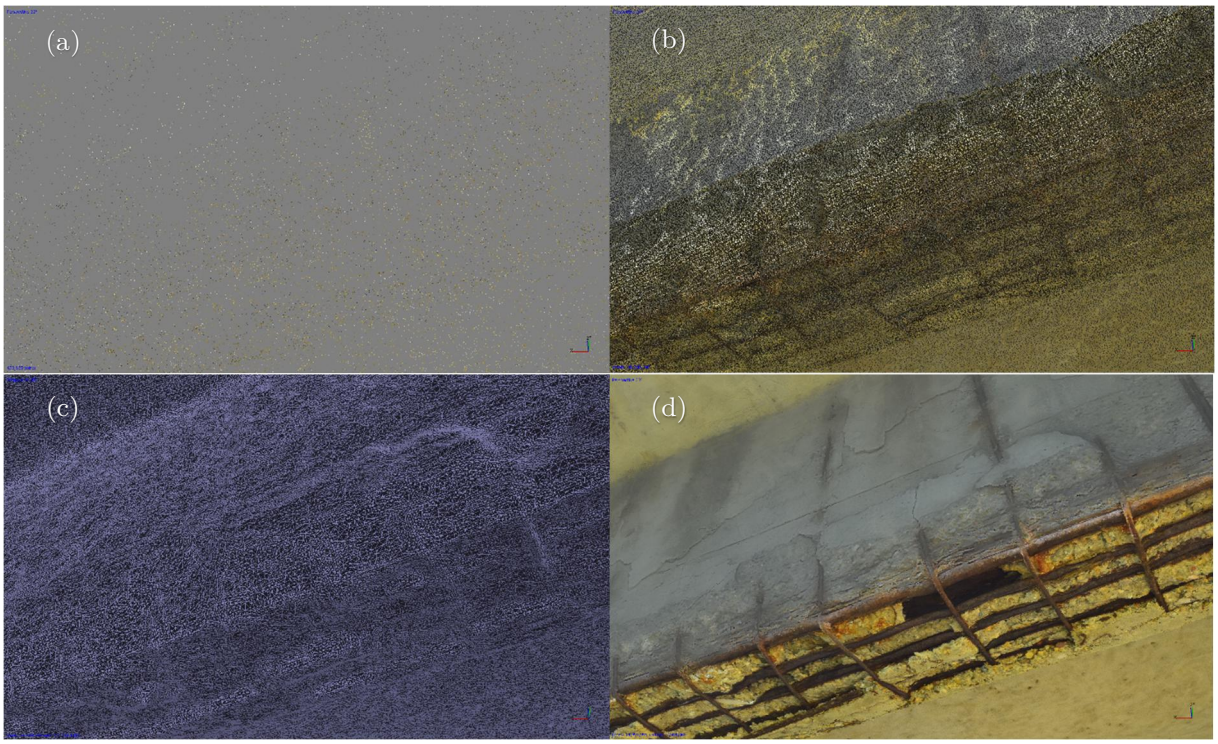


Figure 5.8. (a) Detail of sparse cloud (433.935 points), (b) detail of dense cloud (30.708.690 points), (c) detail of triangular mesh (14.999.999 faces, 7.504.065 vertices), (d) detail of triangulated mesh with texture.

5.4 Pre-processing of inputs for damage geometry extraction

The 3D model geometry from the as-designed BrIM was extracted as an Wavefront OBJ file using the IFCOpenShell command line converter tool [84]. Afterwards, the process described in Chapter 4 is followed to produce the as-is BrIM.

5.4.1 Mesh registration to BrIM geometry

Using the Photoscan software [81], a 3D point cloud registration and triangular mesh reconstruction from the 3D point cloud are performed at once (the second and the third activity in the process map shown in Figure 4.12). Since the as-built IFC version of the bridge did not exist, it had to be generated form the as-designed model. The as-built IFC version of the bridge is what is semantically enriched later on. After the as-designed IFC geometry is converted into triangular mesh, the as-is point cloud and as-designed IFC models are imported into CloudCompare software for a two-step registration (i.e. scaling and alignment) (Figure 5.9a). Transformed and scaled photogrammetry-based triangular mesh overlapped with the as-designed IFC-based mesh is shown in Figure 5.9b.

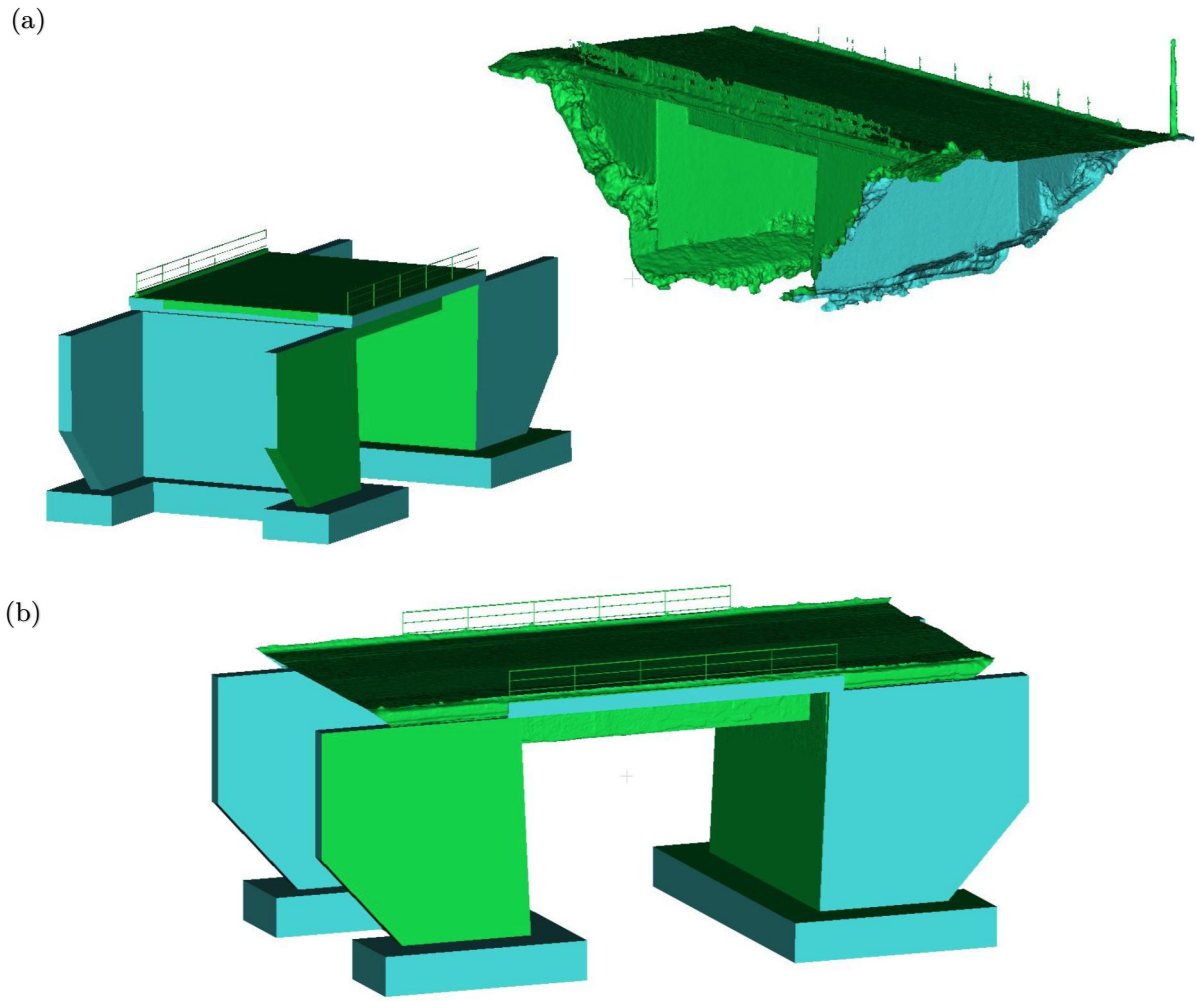


Figure 5.9 (a) Original positions of as-designed IFC-based mesh and photogrammetry-based mesh, and (b) transformed and scaled photogrammetry-based triangular mesh overlapped with as-designed IFC-based mesh.

5.4.2 As-designed adjustment to as-built BrIM

Dimensions of the transformed triangular mesh are analyzed in equidistant cross sections. MeshLab software is used for slicing the mesh. Transversal and longitudinal bridge sections are shown in Figure 5.10a and Figure 5.10b. Exported to DXF format, the cross-sections are overlapped using Autodesk AutoCad. Neglecting the obviously missing volume of the structural elements due to damage, the centerline of both transverse and longitudinal section is manually fitted (Figure 5.10c). The only criteria were to keep the cross section symmetric after the dimension adjustment. The automation of this step is possible, however it was out of the scope of this research.

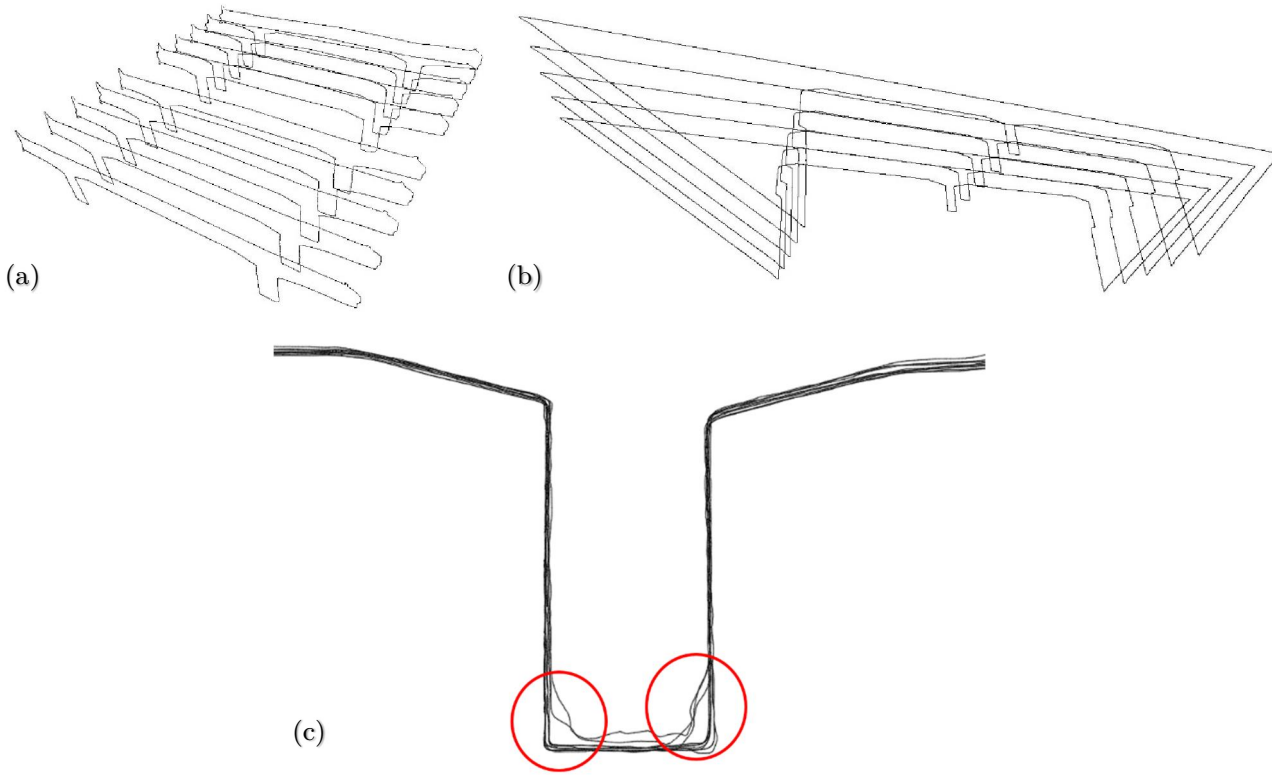


Figure 5.10. (a) Transversal bridge sections. (b) Longitudinal bridge sections. (c) Main girder equidistant cross section overlapped (missing cross section due to damage is circled in red).

After the analysis, the shape of the bottom console contours is slightly changed and translated for approximately 5 cm upwards. The adjustments of girder dimensions are shown in Table 5.2. Finally, the BrIM is remodeled according to updated bridge dimensions using Autodesk Revit. The remodeled BrIM is exported as an updated as-built IFC model.

Table 5.2. Adjustment of girder dimensions from as-designed to as-built.

Element	Main Girders	Middle Cross Girder	End Cross Girders
As-designed cross section dimensions (cm)	50 x 110	30 x 106	50 x 125
As-built cross section dimensions (cm)	52.44 x 110	32 x 110	40 x 125

5.5 Extracting Damage

Once generated, scaled and positioned to perfectly fit the IFC geometry, the photogrammetry-based mesh was used to identify defects as potential damages and extract their geometry. The deviation of the photogrammetry-based mesh from the IFC-based mesh is acquired using a MeshLab [87] software. A CSG Boolean operation tool is used on the meshes. The results were meshes representing damages (Figure 5.11).

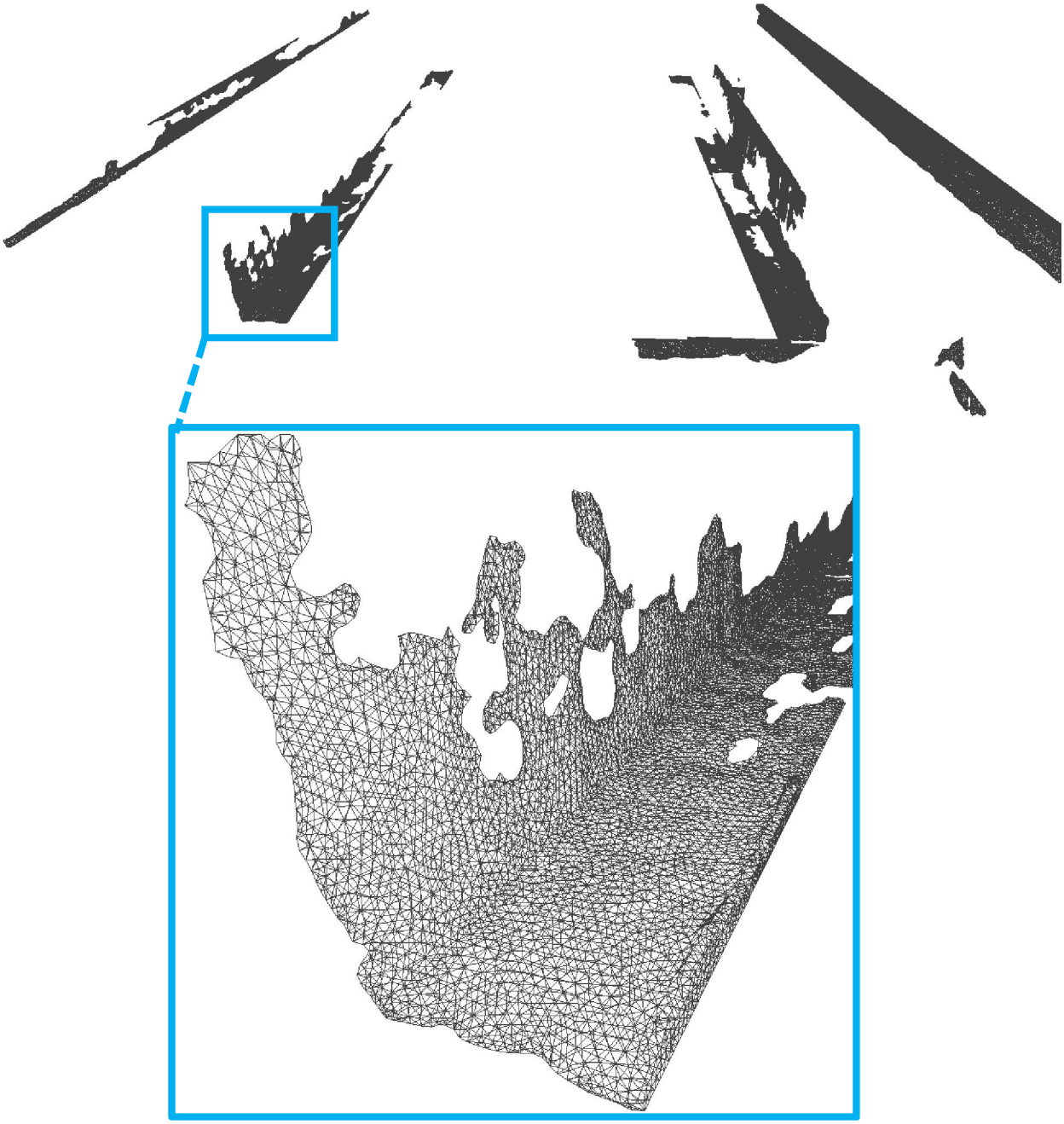


Figure 5.11. Triangulated damage meshes: geometry of the missing concrete as a result of spalling.

5.6 BrIM geometric and semantic enrichment

After the as-designed BrIM is firstly adjusted to the as-built geometry, it is further enriched with geometric and semantic damage information, i.e. inspection findings. The final result, as-is BrIM of the Bridge over river Gročica is shown in Figure 5.12.

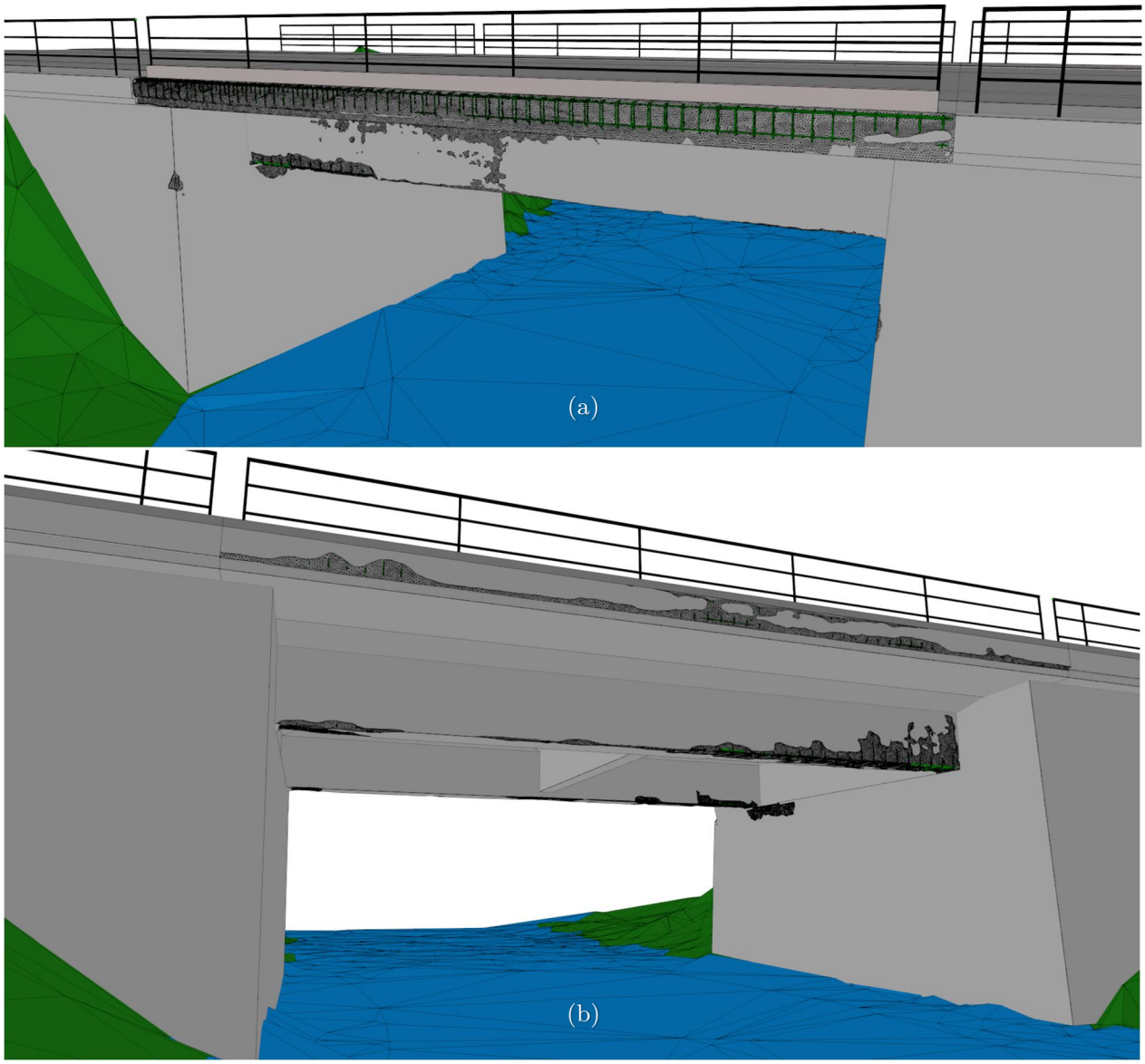
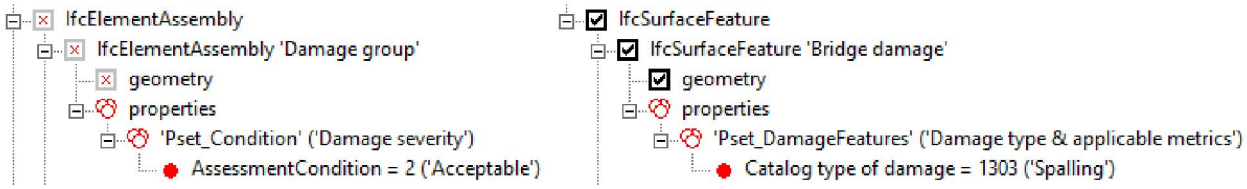


Figure 5.12. As-is IFC geometric representation of Bridge over river Gročica: (a) west side view, (b) east side view.

The as-is IFC file is generated using the proposed method, described in Chapter 4. The as-is photogrammetrically acquired triangular mesh of the bridge IFC was compared against the as-is IFC. The STEP code snippet and the IFC tree of the damage semantics are shown in Figure 5.13, whereas Figure 5.14 shows the STEP code snippet describing bridge inspection results. The IFC STEP code snippet describing the girder damage, its geometric representation and relationship with the girder is shown in Figure 5.15. Both the as-designed, and the as-is IFC file are enclosed on the provided USB flash memory. Visualizations of the baseline IFC and the one with automatically detected damages are shown in Figure 5.16 - Figure 5.18.

```
#121013 = IFCPROPERTYSET('1S5u9YpID6kBOhVrFxchDd', $, 'Pset_DamageFeatures',
          'Damage type & applicable metrics', (#121014));
#121014 = IFCPROPERTYSINGLEVALUE('Catalog type of damage', 'Spalling', '1303', $);
#121015 = IFCRELDEFINESBYPROPERTIES ('1kZOnhb4v4fx$Vt$Yar7v4', $, $, $, #121005, #121013);
#121016 = IFCELEMENTASSEMBLY('1IuystymbElBpwLStuzqYP', $, 'Damage group', $,
          'INSPECTIONFINDING', $, $, $, $, .USERDEFINED.);
#121017 = IFCRELAGGREGATES('2XBTDOJsbDkhXwE82N67ph', $, $, $, #121016, (#121005, #121022,
          #121026, #121030));
#121018 = IFCPROPERTYSET('0XWwlHQT1FQgX3NYFEYPIi', $, 'Pset_Condition', 'Damage severity',
          (#121019));
#121019 = IFCPROPERTYSINGLEVALUE('AssessmentCondition', 'Acceptable', '2', $);
#121020 = IFCRELDEFINESBYPROPERTIES('1v7lckcjf7lgeYumWM_tX_', $, $, $, #121016, #121018);
```

(a)



(b)

Figure 5.13. Damage semantics: (a) IFC STEP code snippet, (b) IFC tree of the embedded viewer.

```
#121059 = IFCTASK('1K9d8_vIX4j86qFmmaKUF3', $, 'Inspection', $, 'Routine inspection',
          $, $, $, $, $, $, #121060, .USERDEFINED.);
#121060 = IFCTASKTIME('Inspection time', 'MEASURED', $, $, $, $, $, $, $, $, $, $, $,
          $, $, $, '2018-08-10T10:00:00', '2018-08-10T15:00:00', $, $);
#121061 = IFCRELASSIGNSTOPROCESS('1KCiqvVunBaBWqdRuHbhWY', $, $, $, (#120829, #120854,
          #120879, #120904, #120933, #120962, #120991, #121016, #121045), $, #121059, $);
```

Figure 5.14. IFC code snippet describing the bridge inspection.

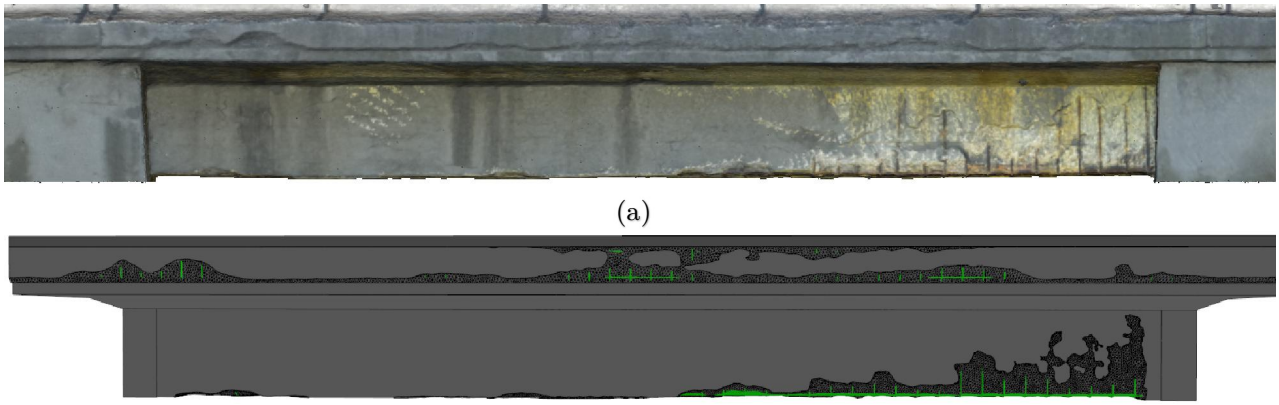
```
/*Bridge girder (IfcBeam)*/
#6 = IFCCARTESIANPOINT((0., 0., 0.));
#67962 = IFCAXIS2PLACEMENT3D(#6, $, $);
#67963 = IFCLOCALPLACEMENT($ , #67962);
#67985 = IFCBEAM('1_T5$UV0vAqBl32$8LDvNv', #41, 'Concrete-Rectangular Beam:52 x 110:
         316571', $, 'Concrete-Rectangular Beam:52 x 110', #67963, #67983, '316571');

/*Damage (IfcSurfaceFeature) geometic representation: IfcTriangulatedFaceSet*/
#121005 = IFCSURFACEFEATURE('3k6jCKxz5BRuSE0w84Zh$V', $, 'Bridge damage', $,'DAMAGE',
          #121006, #121009, $, .USERDEFINED.);
#121006 = IFCLOCALPLACEMENT($, #121007);
#121007 = IFCAXIS2PLACEMENT3D(#6, $, $);
#121009 = IFCPRODUCTDEFINITIONSHAPE($, $, (#121010));
#121010 = IFCSHAPEREPRESENTATION(#101, 'Body', 'Tesselation', (#121011));
#121011 = IFCTRIANGULATEDFACESET(#121012, $, .T., ((6, 8, 9), (12, 13, 366), ...)
#121012 = IFCCARTESIANPOINTLIST3D(((-893.999, -352.476, -186.071), ...)
#121021 = IFCRELVOIDSELEMENT('0sIPpA3CH9ivdmNTQ6k0Gd', $, $, $, #67985, #121005);
```

Figure 5.15. STEP code snippet describing the geometric representation of girder damage.

(a)

(b)

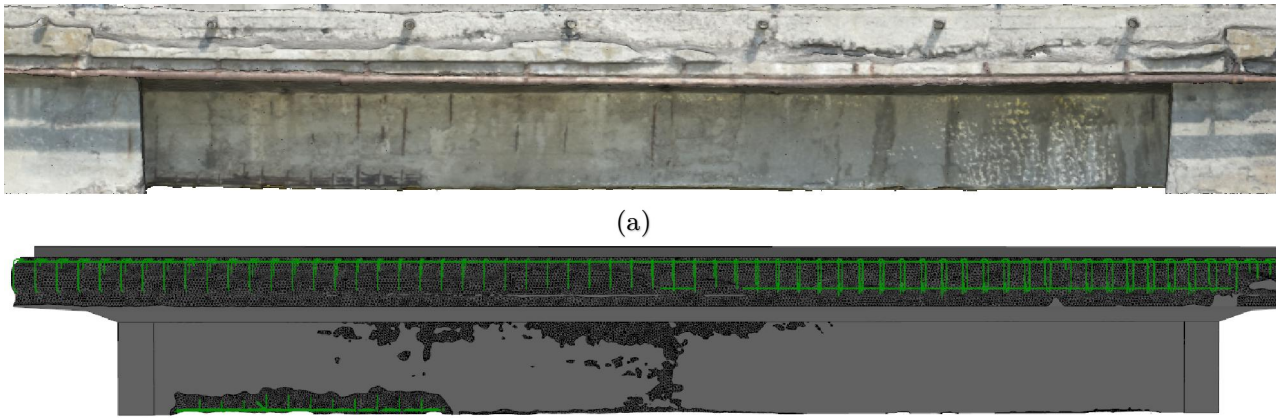
Figure 5.16. Bottom of the bridge superstructure: (a) photogrammetry-based triangular mesh, and (b) the as-is IFC with spalling damage geometry.



(a)

(b)

Figure 5.17. East girder and curb of the bridge: (a) photogrammetry-based triangular mesh, and (b) the as-is IFC with spalling damage geometry.



(a)

(b)

Figure 5.18. West girder and curb of the bridge: (a) photogrammetry-based triangular mesh, and (b) the as-is IFC with spalling damage geometry.

6 DISCUSSION

6.1 Representation of Damage Semantics in IFC Model

After being reconstructed as triangulated geometry, the extracted damage clusters are injected into the BrIM, using the proposed IFC geometric representation and prototype software. To generate a visible void, the damage mesh needs to enclose the damaged element. However, the damage mesh resulting from damage detection sub-process fails to fulfill this criterion for two reasons. Firstly, the outer surfaces of the mesh and damaged element coincide. Secondly, the mesh edges are chamfered as a result of a CSG computation. As the most straightforward way to overcome these issues, the authors chose to enlarge the damage mesh before injecting it into IFC, by scaling it. The scale is determined based on empirical tests and the results of these tests are shown in Figure 6.1. Figure 6.1a shows that the damage mesh in original size does not produce any visible void, whereas Figure 6.1b shows the artifacts due to the insufficient enlargement of the mesh (e.g., damage mesh does not completely enclose the damaged element). The smallest scale which does not produce any artifacts is 100.5 % of original mesh size (Figure 6.1c). Thus, each damage mesh is scaled to this percentage before introducing to the as-built IFC. The change of the scale of damage mesh slightly increases the damage dimensions in the resulting as-is IFC geometry. Most probably, the problem of artifacts can be solved even without the scale change. However, that kind of solution would require changes in CSG Boolean difference algorithm and mesh reconstruction algorithm.

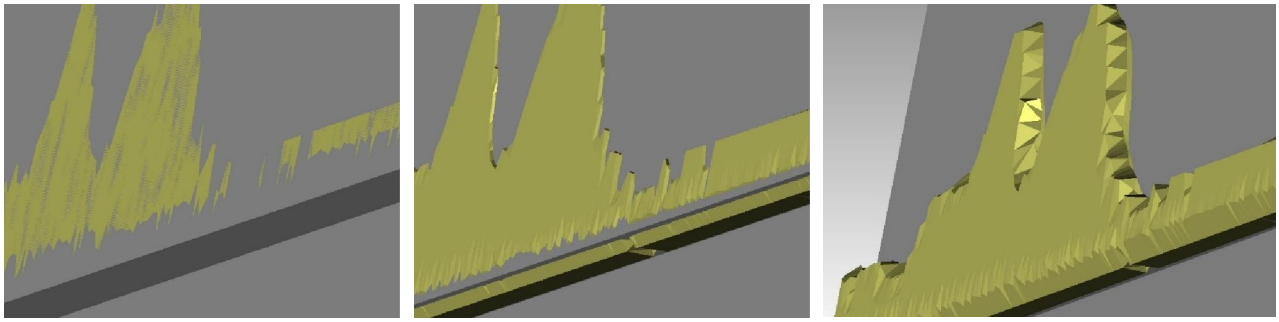


Figure 6.1. Analysis of different damage mesh scales (the last one is used as the damage geometry representation): (a) original size, (b) scaled to 100.2%, (c) scaled to 100.5%.

The proposed IFC structure succeeded in representing all the detected damages. Using the simple tree structure in IFC viewer, the damage structure and semantics can easily be navigated and selected for further inspection by the user. Due to a large amount of added data (e.g., coordinates of mesh vertices, mesh indices), the IFC file size increased considerably after the semantic enrichment. The size of the as-built IFC file is 4.39 MB, whereas the size of the as-is file is 13.1 MB.

6.2 Automation of damage detection

The proposed method for generating the as-is BrIM is described by the process (Figure 4.1, Figure 4.2, and Figure 4.12) which is automated to a certain extent. This automation is however present only on the activity level. Activities such as *BrIM geometry conversion to 3D triangular mesh* and *Triangular mesh reconstruction from 3D point cloud* are completely automatic. The proposed process also includes semi-automated activities, such as *UAV-based bridge photo capturing* and *Injecting damage into BrIM.* Finally, there are completely manual activities. One of these activities is certainly the *Damage geometry extraction.* If understood as damage detection, it is easy to realize that the accuracy of this activity output is critical for the entire process. Moreover, this accuracy of this step is critical for the bridge condition assessment.

As an extension of the method proposed in this research, Isailović et al. [76] proposed the method for automated damage detection. The proposed method uses image-based multiview classification to detect and extract large concrete damages, such as concrete spalling. Image-based, multiview classification is used, where point cluster regions are spatially divided using an octree data structure. Each of the octree nodes is used to generate a cubemap representation of the point cluster inside it. These cubemap images are then classified using a retrained Convolutional Neural Network (CNN), and the damage classified clusters can then be extracted.

Using a 3D point cloud model of a given bridge, the proposed multiview approach can detect spalling damage regions from partitioned 3D point cluster regions of the model - by generating consecutive images of the given point cluster region, classifying them using a retrained CNN, averaging the classification result by the number of multiview images of the given point cluster, and streaming the classification result back to the point cluster as a new semantic. Classified this way, point clusters representing damages are then reconstructed to generate damage meshes, ready for injection into the as-built BrIM.

6.3 Benefits for the BMS

Contributions of the proposed method to BMSs and Bridge Management in general, can be analyzed from two perspectives:

- Improvement of the current BMS in terms of information quality, and
- Providing a basis for a new bridge quality control paradigm.

Since the damage data model is based on KUBA [14], whereas the inspection report of the Bridge over river Gročica (Appendix 2) is written following the rules of RoS [14], a

straightforward comparison of the generated as-is BrIM with the corresponding inspection is not possible. Accordingly, the analysis will be provided in a rather qualitative manner.

The *InspectionFinding* semantics fully comply with the inspection database organization. Therefore, a direct linking between the as-is IFC and the BMS is easy to establish. Once established, each inspection finding from the BMS database, would point to the *InspectionFinding* instance of the as-is IFC. Here, the real benefit of IFC can be realized. Much more reliable element condition rating and making decisions about maintenance can result from conclusions based on the simple 3D damage visualization. However, the full potential of an as-is IFC goes far beyond the navigation of the damaged bridge element in a viewer. A plenty of explicit and accurate geometric information can be easily derived from IFC. An exact volume loss or a cross-section area loss of a spalled concrete element, a reinforcement cross-section reduction due to a corrosion, or various deformations of structural elements are explicitly or implicitly included in the as-is IFC.

Besides a short-term BMS benefits, an as-is IFC provides a solid basis for a fundamental improvement of the current Bridge Management. Providing the exact dimensions of bridge elements directly facilitates the condition rating. However, according to findings of Hajdin et al. [98], the current approach of bridge quality assessment is inadequate since the bridge condition rating is only loosely related to safety and serviceability that are widely accepted as key performance indicators (KPIs) and should govern the decision-making process. Moreover, the valuable information about safety and serviceability for different load situations once evaluated in the design phase is not even stored in the BMS's database. Providing the precise and up-to-date information about the exact geometry, material characteristics, and loads, BrIM can serve as a basis for the bridge safety and serviceability calculation. However, considering the maintenance prioritization as the main task of Bridge Management, and the number of bridges in one road infrastructure network, a direct calculation of KPIs for each bridge would not be viable. To provide a more efficient way of acquiring bridge KPIs, Hajdin et al. [98] introduced the concept of "vulnerable zones". The "vulnerable zones" are the areas of the bridge where the damage impact on safety or serviceability is expected to be more pronounced than outside of it. There are just a few types of vulnerable zones and all of them can be easily defined in BrIM by bounding boxes positioned and sized relative to a BrIM element, a group of elements, or the entire BrIM. KPIs would be assessed based on the reduced resistance of the as-designed bridge, due to the occurred damages, and the possible load increase, due to the changed traffic conditions. Based on this principle, Isailović et al. [99] proposed a method for assessing the bridge KPIs by using Bayesian nets. The information about damages scoped in the established "vulnerable zones" can be derived from BrIM and linked with corresponding nodes of the Bayesian net. This way, after each inspection, i.e. update of the as-is BrIM, node values of the Bayesian net would be updated as well, thus providing a direct bridge condition assessment.

6.4 Economic viability of the proposed method

The inspection duration analysis presented in Table 6.1 includes the photogrammetric survey and point cloud generation. The analysis does not include the as-designed BrIM generation because it is a one-time procedure. To analyze the duration of the proposed method, the most efficient variant of the overall process is taken into account. Therefore, rather than identifying damages manually, the automatic damage detection, proposed by Isailović et al. [76], is applied.

Table 6.1. Inspection duration (proposed method, with automatic damage detection applied)

Process Activity	Software components	Dominantly engaged	Domain expertise	Duration [h]
Photogrammetric survey	None	Human labor	UAV photogrammetry	5.0
3D point cloud generation	Agisoft Photoscan	Computer	Photogrammetry	38.0
BrIM geometry conversion to 3D triangular mesh	IfcOpenShell	Computer	BIM	5.0
3d point cloud registration to BrIM-based mesh	CloudCompare	Human labor	Computer graphics	1.5
Multiview classification	Prototype web-based app	Computer	Computer graphics, machine learning	0.4
Reconstruction of damage clusters	MeshLab	Computer	Computer graphics	1.0
CSG subtraction of reconstructed damage cluster from IFC-based mesh	MeshLab	Computer	Computer graphics	2.0
Triangular mesh reconstruction from 3D point cloud	MeshLab	Computer	Computer graphics	10.0
As-designed adjustment to as-built BrIM	MeshLab, AutoCad, Revit	Human labor	Structural engineering, BIM	4.0
Identifying the appropriate damage types in the BMS catalog	None	Human labor	Structural engineering	0.2
Injecting the damage into BrIM	Prototype app	Human labor	BIM	1.0
			total:	68.1

The total duration of the inspection was approximately 68.1 hours, of which 56.4 hours were spent on activities dominantly performed by computer. All computer processing is performed on a workstation PC with the following hardware specifications:

- CPU: AMD FX 8350
- GPU: ASUS STRIX-GTX980

- Memory: 32GB

The traditional routine inspection significantly varies in duration, depending on the type of bridge structure, the number of spans, deck area, minimum applicable condition rating, time of the year, access equipment and inspector. The average normalized duration of an inspection of reinforced concrete slab bridges is 462 min/1000 ft2 [100]. The inspection of the Bridge over river Gročica (133 m2 = 1440 ft2) therefore, would have taken 665 min, or 11 hours. Besides the fieldwork, this duration refers to the report writing.

To perform the proposed process in this case study took six times longer than the traditional one would. Nevertheless, whereas the traditional process requires permanent human labor engagement, most of the work (83%) in the proposed process is done by a computer. The generation of the photogrammetry-based point cloud is certainly the most time-consuming activity in the process, taking more than half of the total inspection time. Using the terrestrial laser scanning (TLS) instead of photogrammetry to acquire the point cloud is indeed much faster and more accurate. One drawback when using a TLS to capture a point cloud is a lack of encoded RGB values, which is crucial for damage detection using multiview classification. Additionally, the duration of a UAV-based inspection undoubtedly depends on the bridge size. Using UAVs is efficient in reducing the inspection time when inspecting large bridges, whereas small bridges are faster inspected in the traditional manner [44]. In 2012, 16.4 billion USD was spent on rehabilitation or replacement of existing highway bridges in the USA [101]. According to Zulfiqar et al. [102], the USA spends only 2.7 billion USD per year on routine bridge inspections. Rather than reducing the inspection cost, this research intended to reduce the several times higher maintenance cost, by providing the inspection deliverable with enough information for proper maintenance prioritization. The current inspection deliverable is a paper report with condition ratings, loosely related to the bridge KPIs (e.g., safety and serviceability). Even the point cloud with marked damages is a much more reliable basis for making decisions on maintenance interventions or bridge closure [44]. In comparison to a simple point cloud, the output of the proposed process, as-is BrIM is the object-oriented model of both the bridge and damages, with accompanying BMS semantics. Structured this way, the damage information is ready to be utilized in the calculation of damaged bridge KPIs, the base for a more objective maintenance prioritization.

6.5 Legal aspects

Finally, there are certain legal limitations for the application of the proposed inspection process. Most country's aviation authorities, such as the Federal Aviation Administration (FAA), forbid UAV operating over a moving vehicle [103]. Due to this kind of regulation, although technically possible, the utilization of UAV in bridge inspection is a bit limited.

The first limitation is related to the pavement inspection, and the second one refers to the inspection of the underside of an overpass. However, the traffic lane closure during the inspection eliminates all legal issues of this kind.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This doctoral dissertation provides a rational and practice-oriented method to develop a new generation of BMS by incorporating inspection findings into a BrIM model. Updated with as-is information about the bridge, the BrIM would reflect the current bridge condition more accurately. The proposed method managed to successfully enrich the previously generated as-built BrIM of the case study bridge by the information about identified damages. The identified damages, such as spallings, accompanied by extensively corroded reinforcement, abutments, and curbs, are injected into the as-built IFC geometry via CSG Boolean operations, and related textual semantics following the requirements of a damage classification from the Swiss BMS KUBA - including the requirements of the IFC 2x4 schema. Rather than proposing a schema extension, the existing schema definitions are used to describe the damage type, extent, and severity — as well to group them according to the location and causing deterioration process.

To enable identification of damages other than spalling, the resolution of the bridge point cloud should be reasonably higher than the one used in the case study. Such a point cloud could be generated using a combination of laser scanning and photogrammetry. Damages detected this way could then be geometrically represented in a BrIM, by mapping image segments depicting damages onto damaged elements as textures [67]. Since the damage data structure and semantics definitions proposed in the presented approach comply with the selected BMS structure, there are two ways to utilize it in the existing BMS: (1) It is possible to either apply the proposed approach on an external IFC file and simply link it with BMS, and (2): One can insert IFC representation of every specific bridge into the BMS. The latter, a more robust method, would require certain extensions of the current BMS software, such as an IFC viewer or custom tools for point cloud processing and damage detection.

Once the data is acquired and properly integrated into BrIM, it can be used as a basis for the straightforward assessment of bridge KPIs such as safety and serviceability. Research by Isailović et al. [99] proposes using Bayesian nets to assess the probability of bridge failure based on inspection findings. Damage location and severity are roughly estimated and introduced to Bayesian mesh by manually updating node values. Having the BrIM that contains all the damage information, nodes of the Bayesian mesh can be automatically updated. Therefore, accurate and exhaustive damage information contained in the BrIM can be introduced to a finite element model representation of the bridge (eventually resulting in an accurate and up-to-date structural analysis of the bridge). Such an analysis would directly provide the most important bridge KPIs: safety and serviceability.

Besides the benefits listed above, the limitations of the proposed approach should be pointed out. Due to both the legal and physical limitations, UAVs cannot always be used

in bridge inspection without road(s) closure, implying additional costs. In the proposed approach, the as-designed BrIM is a prerequisite. However, a minority of existing bridges have BrIM representation, so the creation of such models by using BIM authoring tools is required. Lack of digital drawings or even the paper ones makes this task rather difficult. Another requirement of the proposed approach is highly precise registration of point cloud to IFC-based geometry. Considering the usually present construction imperfections, this task is sometimes very difficult. For that reason, manual registration is proposed, rather than the automated one. Although the registration precision is achieved, this task increased the total inspection duration for 1.5 hours. Even with precise registration, to be able to detect fine damage, such as crack, the proposed approach would require an ultrahigh-resolution point cloud. Last, but not least, the case study showed that the proposed approach in the bridge inspection lasts approximately seven times longer than the traditional inspection.

7.2 Future work

The proposed data model for bridge damages has proven the ability to fully represent all identified damages, both by semantical and geometrical means. The data model is established to comply with one particular BMS, thus the extension of the current IFC schema has not been considered. Nevertheless, a more efficient way to store large portions of data, originating from point clouds with millions of points, in the data structure such as IFC should be established. This improvement should mainly address the graphic processing issue.

Following the used workflow, defined criteria, and requirements for the establishment of a new data model for damages, a more general data model compliant with many different BMSs can be developed. When such a comprehensive data model is established, the need for extension of the current version of IFC schema should be considered. Namely, it is likely that the newly established model would be rather more complex than the one proposed in this dissertation and therefore it would be reasonable to thoroughly analyze the ability of current schema to support the IFC representation of this data model.

Although proven successful, the proposed method for generation of the as-is BrIM can be improved in terms of efficiency. Development of a comprehensive software application comprising the majority of activities from the proposed process (Figure 4.1, Figure 4.2, and Figure 4.12) would significantly shorten the duration of an entire process. Few process activities, such as the photogrammetric survey, and the adjustment of as-designed to as-built BrIM are for obvious reasons impossible to integrate in the previously described comprehensive software application. However, the latest autonomous UAV technology promises to enable the completely programmed flight path, even in very demanding environments. Recent commercial solutions combine a flight trajectory derivation from a

3D model of the subject, and highly agile aircrafts equipped with ultra-sensitive sensors enabling the fast avoidance of static and moving solid obstacles. For the time being, the unapproachable terrain, mostly irregular shapes of bridges, and the water proximity exceed the power of this technology. However, with the advances in computer vision, augmented reality and machine learning, in not so distance future it might be possible to insert a georeferenced BrIM into a specialized software, set the requirements, and let the UAV autonomously inspect the bridge.

After the UK adoption of BIM as a standard for data exchange on governmental construction projects, it is likely to expect the rest of the world will follow. The BIM mandates will thus provide repositories of BrIMs for newly constructed bridges. However, to apply the proposed method for the as-is BrIM generation in the inspection of old bridges will still require the as-designed BrIM modeling. Besides the often difficulty of finding the documentation, which is usually paper based, as shown in the presented case study, this additional step is significantly time consuming. The recent efforts in the so-called "scan-to-BIM" technology enables the automatic BrIM generation based on a 3D point cloud. The SeeBridge project [104] had success in generating the basic geometry of few common types of overpasses. The development in this field, combined with the development of UAV scanning technology, would eventually enable the completely automatic generation of BrIM with LOD 300, or even higher, thus eliminating the need for two steps of the proposed process: generation of as-designed BrIM and the adjustment of as-designed to as-built BrIM.

Third, the image-based classification could theoretically be performed right after the image capture stage and before the 3D point cloud generation stage. Therefore, the generated point cloud model could already contain point with visual or encoded semantics indicating the presence of potential damage features. The system would then just need to extract these clusters without needing to classify the point cloud itself and reconstruct them into geometric representations that would be used for semantic injection into as-is BIM model via CSG operations.

Digital Twin (DT) technology uses the Internet of Things (IoT) technology for integrating large amounts of various digital data from sensors embedded into bridge, for the purpose of bridge monitoring. The data structure storing this information and the system for its procession are commonly named DT. The current data models used for Digital Twins are purely semantic and include no visual information. The utilization of BrIM as a data repository for DT on one hand, and integration of the proposed as-is BrIM generation method with the sensor data on the other may definitely change the paradigm of Bridge Management. Moreover, DT can be perceived as the BIM successor, reflecting the subject in various aspects throughout its entire lifetime.

Finally, the method presented in this research is applicable to a much broader range of challenges in design, construction, and maintenance of the built environment, including the processing of large point clouds to compile BIM models and to detect damage, a

compilation of as-built and as-is models with explicit geometry and semantics. Further development of this research aims to contribute to the acquiring and use of DTs for managing the built environment.

REFERENCES

List of references

[1] P. Virilio, The Administration of Fear, Los Angeles: Semiotext(e), 2012.

[2] A. Smith, An Inquiry into the Nature and Causes of the Wealth of Nations, McMaster University Archive for the History of Economic Thought, 1776.

[3] J.-P. Rodrigue, The Geography of Transport Systems, New York: Routledge, 2017.

[4] P. Virilio and S. Lotringer, Crepuscular Dawn, New York: Semiotext(e), 2002.

[5] Eurostat, "Modal split of inland passenger transport," 2015. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Modal_split_of_inland_passenger_transport,_2015_(%25_of_total_inland_passenger-kilometres).png. [Accessed 10 October 2018].

[6] Eurostat, "Freight transport in the EU-28 modal split of inland transport modes," 2016. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Freight_transport_in_the_EU-28_modal_split_of_inland_transport_modes_(%25_of_total_tonne-kilometres)_2016.png#file. [Accessed 22 October 2018].

[7] World Road Association (PIARC), "The Importance Of Road Maintenance," World Road Association, 2014.

[8] M. J. Ryall, Bridge Management, Oxford: Elsevier, 2009.

[9] Close-up Engineering Network, "Close-up Engineering," [Online]. Available: https://buildingcue.it/genova-silenzio-prof-cosenza/11655/. [Accessed 15 April 2020].

[10] J. Lauridsen, J. Bjerrum, N. Hutzen Andersen and B. Lassen, "Creating a Bridge Management System," *Structural Engineering International,* vol. 8, no. 3, pp. 216 - 220, 1998.

[11] Službeni glasnik, "Pravilnik o tehničkim normativima za eksploataciju i redovno održavanje mostova," 10 July 1992. [Online]. Available: http://www.pravno-informacioni-sistem.rs/SlGlasnikPortal/eli/rep/slsrj/drugidrzavniorganiorganizacije/pravilnik/1992/20/9/reg. [Accessed 6 December 2019].

[12] H. Hawk, "BRIDGIT: User-Friendly Approach to Bridge Management," Transportation Research Board, Washington DC, 1999.

[13] I. Zambon, A. Vidovic, A. Strauss and J. C. Matos, "Condition Prediction of Existing Concrete Bridges as a Combination of Visual Inspection and Analytical Models of Deterioration," *Applied Sciences,* vol. 9, no. 1, 2019.

[14] Federal Roads Office FEDRO, *KUBA 5.0 [Computer software],* Bern, 2019.

[15] Bundesamt für Strassen ASTRA, *KUBA 5.0 Fachapplikation Kunstbauten und Tunnel,* Bern: ASTRA, 2012.

[16] Z. Mirzaei, B. Adey, L. Klatter and P. Thompson, "The IABMAS Bridge Management Committee Overview Of Existing Bridge Management Systems," International Association for Bridge Maintenance and Safety - IABMAS, 2014.

[17] C. Eastman, D. Fisher, G. Lafue, J. Lividini, D. Stoker and C. Yessios, "An Outline of the Building Description System," Carnegie-Mellon University, Institute of Physical Planning, Pittsburgh, 1974.

[18] National Institute of Building Sciences, "United States National Building Information Modeling Standard," BuildingSMARTalliance, 2007.

[19] BIM Forum, "Level of Development Specification," 2014. [Online]. Available: https://bimforum.org/lod/. [Accessed 26 December 2019].

[20] Graphisoft, "BIM Curriculum," 2015. [Online]. Available: https://www.graphisoft.com/learning/bim-curriculum/. [Accessed 28 December 2019].

[21] R. Sacks, C. Eastman, G. Lee and P. Teicholz, BIM Handbook A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, Hoboken: John Wiley & Sons, 2018.

[22] Trimble, *Tekla Structures [computer sofware].*

[23] Transportation Research Board, "Nondestructive Testing to Identify Concrete Bridge Deck Deterioration," National Academy of Sciences, Washington D.C., 2013.

[24] A. Strauss, A. Mandić Ivanković, J. C. Matos and J. R. Casas, "COST Action TU1406 WG1 Technical Report Performance Indicators for Roadway Bridges," European Cooperation in Science and Technology (COST), 2016.

[25] J. Bień, K. Jakubowski, T. Kamiński, J. Kmita, P. Rawa, P. Cruz and M. Maksymowicz, "Railway bridge defects and degradation mechanisms," in *Sustainable Bridges Assessment for Future Traffic Demands and Longer Lives*, Wrocław, Dolnośląskie Wydawnictwo Edukacyjne, 2007, pp. 105-116.

[26] F. Ghodoosi, A. Bagchi and T. Zayed, "System-Level Deterioration Model for Reinforced Concrete Bridge Decks," *Journal of Bridge Engineering,* vol. 20, no. 5, p. 04014081, 2014.

[27] C. J. Ball and D. W. Whitmore, "Innovative corrosion mitigation solutions for existing concrete structures," *International Journal of Materials and Product Technology,* vol. 23, no. 3-4, pp. 219-239, 2005.

[28] Carpet Removal Sydney, "Concrete Spalling," Carpet Removal Sydney, [Online]. Available: https://www.carpetremovalsydney.com.au/concrete-spalling/. [Accessed 12 January 2020].

[29] B. D. Zakić, A. Ryzinski, C. Guo-Hong and J. Jokela, "Classification of damage in concrete bridges," *Materials and Structures,* no. 24, pp. 268-275, 1991.

[30] N. Steinkamp, "Bridge Deck Deterioration," 8 December 2015. [Online]. Available: https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=3269&context=roadschool. [Accessed 13 January 2020].

[31] L. Lindbladh, *Bridge Inspection Manual,* Jonkoping: Swedish National Road Administration, 1996.

[32] Kentucky Transportation Cabinet, *Kentucky Bridge Inspection Procedures Manual,* Frankfort: Kentucky Transportation Cabinet, 2017.

[33] G. Hearn, "Bridge Inspection Practices," National Academy of Sciences, Washington D.C., 2007.

[34] M. Moore, B. Phares, B. Graybeal, D. Rolander and G. Washer, "Reliability of visual inspection for highway bridges," Federal Highway Administration, Atlanta, 2001.

[35] Federal Highway Administration, "National Bridge Inspection Standards," 14 December 2004. [Online]. Available: https://www.govinfo.gov/content/pkg/FR-2004-12-14/pdf/04-27355.pdf. [Accessed 15 January 2020].

[36] C. R. Farrar, K. Worden and J. Dulieu-Barton, "Principles of Structural Degradation Monitoring," in *Encyclopedia of Structural Health Monitoring*, John Wiley & Sons, Ltd, 2009.

[37] H.-P. Chen, Structural Health Monitoring of Large Civil Engineering Structures, John Wiley & Sons, Ltd, 2018.

[38] T. M. Lillesand, R. W. Kiefer and J. W. Chipman, Remote Sensing and Image Interpretation, Hoboken: John Wiley & Sons, Inc, 2015.

[39] W. W. Greenwood, J. P. Lynch and D. Zekkos, "Applications of UAVs in Civil Infrastructure," *Journal of Infrastructure Systems,* vol. 25, no. 2, pp. 1-21, 2019.

[40] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz and A. Wiechert, "Point Clouds: Lidar versus 3D Vision," *Photogrammetric Engineering & Remote Sensing,* vol. 76, no. 10, pp. 1123-1134, 2010.

[41] C. Eschmann and T. Wundsam, "Web-Based Georeferenced 3D Inspection and Monitoring of Bridges with Unmanned Aircraft Systems," *Journal of Surveying Engineering,* vol. 143, no. 3, p. 04017003, 2017.

[42] M. Golparvar-Fard, J. Bohn, J. Teizer, S. Savarese and F. Peña-Mora, "Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques," *Automation in Construction,* vol. 20, no. 8, pp. 1143-1155, 2011.

[43] S. Chen, D. F. Laefer, E. Mangina, I. S. M. Zolanvari and J. Byrne, "UAV Bridge Inspection through Evaluated 3D Reconstructions," *Journal of Bridge Engineering,* vol. 24, no. 4, pp. 1-15, 2019.

[44] J. Wells and B. Lovelace, "Improving the Quality of Bridge Inspections Using Unmanned Aircraft Systems (UAS)," Minnesota Department of Transportation, St. Paul, 2018.

[45] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures," *Automation in Construction journal,* vol. 22, pp. 567-576, 2012.

[46] S. German, I. Brilakis and R. DesRoches, "Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments," *Advanced Engineering Informatics,* vol. 26, no. 4, pp. 846-858, 2012.

[47] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich and V. Rodehorst, "Framework for automated UAS-based structural condition assessment of bridges," *Automation in Construction,* vol. 97, pp. 77-95, 2019.

[48] A. Sironi, V. Lepetit and P. Fua, "Multiscale Centerline Detection by Learning a Scale-Space Distance Transform," in *Conference on Computer Vision and Pattern Recognition*, 2014.

[49] P. Hüthwohl and I. Brilakis, "Detecting healthy concrete surfaces," *Advanced Engineering Informatics,* vol. 37, pp. 150-162, 2018.

[50] Y. Xu and Y. Turkan, "Bridge Inspection Using Bridge Information Modeling (BrIM) and Unmanned Aerial System (UAS)," in *Advances in Informatics and Computing in Civil and Construction Engineering*, Springer, 2019, pp. 617-624.

[51] T. Qu and W. Sun, "Usage of 3D Point Cloud Data in BIM (Building Information Modelling): Current Applications and Challenges," *Journal of Civil Engineering and Architecture,* vol. 9, pp. 1269-1278, 2015.

[52] S. Tuttas, A. Braun, A. Borrmann and U. Stilla, "Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4D BIM," *PFG--Journal of Photogrammetry, Remote Sensing and Geoinformation Science,* vol. 85, pp. 3-15, 2017.

[53] E. B. Anil, P. Tang, B. Akinci and D. Huber, "Deviation analysis method for the assessment of the quality of the as-is Building Information Models generated from point cloud data," *Automation in Construction,* vol. 35, pp. 507-516, 2013.

[54] H. Barki, F. Fadli, A. Shaat, P. Boguslawski and L. Mahdjoubi, "BIM models generation from 2D CAD drawings and 3D scans: an analysis of challenges and opportunities for AEC practitioners," *Building Information Modelling (BIM) in Design, Construction and Operations,* vol. 149, pp. 369-380, 2015.

[55] International Organization for standardization (ISO), "ISO 10303-11:2004 Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual," November 2004. [Online]. Available: https://www.iso.org/standard/38047.html. [Accessed 23 January 2020].

[56] A. Borrmann, J. Beetz, C. Koch, T. Liebich and S. Muhic, "Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models," in *Building Information Modeling Technology Foundations and Industry Practice*, Cham, Springer, 2018, pp. 81-126.

[57] International Organization for standardization (ISO), "ISO 10303-21:2016 Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure," March 2016. [Online]. Available: https://www.iso.org/standard/63141.html. [Accessed 23 March 2020].

[58] D. Schenck and P. Wilson, Information Modeling the EXPRESS Way, New York: Oxford University Press, Inc., 1994.

[59] C. Koch and M. König, "Data Modeling," in *Building Information Modeling Technology Foundations and Industry Practice*, Cham, Springer, 2018, pp. 43-62.

[60] buildingSmart International, "Industry Foundation Classes Version 4.1.0.0," June 2018. [Online]. Available: https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/. [Accessed 23 January 2020].

[61] J. Beetz, *Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies,* Eindhoven: Technische Universiteit Eindhoven, 2009.

[62] B. McGuire, R. Atadero, C. Clevenger and M. Ozbek, "Bridge Information Modeling for Inspection and Evaluation," *Journal of Bridge Engineering,* vol. 21, no. 4, p. 04015076, 2016.

[63] S. B. Chase, Y. Adu-Gyamfi, A. Aktan and E. Minaie, "Synthesis of National and International Methodologies Used for Bridge Health Indices," Federal Highway Administration, Georgetown Pike McLean, 2016.

[64] A. Borrmann, M. König, C. Koch and J. Beetz, Building Information Modeling Technology Foundations and Industry Practice, Cham: Springer, 2018.

[65] N. Yabuki, E. Lebegue, J. Gual, T. Shitani and L. Zhantao, "International Collaboration for Developing the Bridge Product Model “IFC-Bridge”," in *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, 2006.

[66] A. Borrmann, S. Muhic, J. Hyvärinen, T. Chipman, S. Jaud, C. Castaing, C. Dumoulin, T. Liebich and L. Mol, "The IFC-Bridge project – Extending the IFC standard to enable high-quality exchange of bridge information models," in *Proceedings of the 2019 European Conference for Computing in Construction*, Chania, 2019.

[67] P. Hüthwohl, I. Brilakis, A. Borrmann and R. Sacks, "Integrating RC Bridge Defect Information into BIM Models," *Journal of Computing in Civil Engineering,* vol. 32, no. 3, 2018.

[68] D. Isailović, M. Petronijević and R. Hajdin, "The future of BIM and Bridge Management Systems," in *Proceedings of 40th IABSE Symposium*, Guimaraes, 2019.

[69] P. P.-S. Chen, "The entity-relationship model - toward a unified view of data," *ACM Transactions on Database Systems,* vol. 1, pp. 9-36, 3 1976.

[70] World Wide Web Consortium (W3C), "Xml essentials - w3c," [Online]. Available: https://www.w3.org/standards/xml/core. [Accessed 20 February 2020].

[71] Federal Roads Office, "KUBA-DB 3.2 IT-Konzept - Klassenmodell," FEDRO, Bern, 2006.

[72] S. Daum and A. Borrmann, "Simplifying the Analysis of Building Information Models Using tQL4BIM and vQL4BIM," in *Proceedings of the EG-ICE workshop 2015*, 2015.

[73] buildingSmart International, "Industry Foundation Classes Version 4.2.0.0," April 2019. [Online]. Available: https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/. [Accessed 11 March 2020].

[74] A. Borrmann and V. Berkhahn, "Principles of Geometric Modeling," in *Building Information Modeling Technology Foundations and Industry Practice*, Cham, Springer, 2018, pp. 27-41.

[75] S. Chaudhuri, "FuzzyPhoton," 2002. [Online]. Available: http://fuzzyphoton.tripod.com/rtref/rtref_c.htm. [Accessed 29 March 2020].

[76] D. Isailović, V. Stojanovic, M. Trapp, R. Richter, R. Hajdin and J. Döllner, "Bridge damage: Detection, IFC-based semantic enrichment and visualization," *Automation in Construction,* vol. 112, p. 103088, 2020.

[77] Object Management Group, "Business Process Model and Notation (BPMN 2.0)," January 2011. [Online]. Available: https://www.omg.org/spec/BPMN/2.0/PDF. [Accessed 24 January 2020].

[78] Agisoft, "Agisoft Metashape User Manual: Professional Edition - Version 1.5," Agisoft LLC, St. Petersburg, 2019.

[79] Bentley Systems, "ContextCapture [computer software]," 2017. [Online]. Available: https://www.bentley.com/en/products/product-line/reality-modeling-software/contextcapture. [Accessed 24 January 2020].

[80] Autodesk, "ReCap [computer software]," 2017. [Online]. Available: https://www.autodesk.com/products/recap/overview. [Accessed 24 January 2020].

[81] Agisoft, "Photoscan [computer software]," 2017. [Online]. Available: https://www.agisoft.com/. [Accessed 24 January 2020].

[82] L. Tang, J. Braun and R. Debitsch, "Automatic aerotriangulation — concept, realization and results," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 52, pp. 122-131, 1997.

[83] Bentley Systems, "Microstation [computer software]," 2017. [Online]. Available: https://www.bentley.com/en/products/product-line/modeling-and-visualization-software/microstation. [Accessed 25 January 2020].

[84] T. Krijnen, "IfcOpenShell," 2018. [Online]. Available: http://ifcopenshell.org/. [Accessed 24 January 2020].

[85] D. Girardeau-Montaut, "Cloudcompare - open source project [computer software]," 2011. [Online]. Available: https://www.danielgm.net/cc/. [Accessed 24 January 2020].

[86] P. Besl and N. McKay, "A method for registration two 3-D shapes," *IEEE Trans Pattern Analysis and Machine Intelligence,* vol. 14, pp. 232-256, 1992.

[87] Open Source, "MeshLab [Computer software]," [Online]. Available: http://www.meshlab.net/. [Accessed 7 February 2020].

[88] M. Kazhdan, M. Bolitho and H. Hoppe, "Poisson Surface Reconstruction," in *Eurographics Symposium on Geometry Processing (2006)*, Cagliari, 2006.

[89] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics,* vol. 5, pp. 349-359, 1999.

[90] 3D Systems, *Geomagic Wrap [computer software].*

[91] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi and R. Scopigno, "Marching intersections: an efficient resampling algorithm for surface management," in *Proceedings International Conference on Shape Modeling and Applications*, 2001.

[92] RDF, *IFC Engine [dynamic library],* 2018.

[93] S. Mašović and R. Hajdin, "Modelling of bridge elements deterioration for Serbian bridge inventory," *Structure and Infrastructure Engineering,* vol. 10, pp. 976-987, 2014.

[94] Autodesk, "Autodesk Revit [Computer software]," 2019. [Online]. Available: https://www.autodesk.com/products/revit/overview. [Accessed 27 January 2020].

[95] IBM, "Weather Underground," The Weather Company, 16 August 2018. [Online]. Available: https://www.wunderground.com/. [Accessed 18 April 2019].

[96] DJI, "Phantom 4 Pro Specification," [Online]. Available: https://www.dji.com/phantom-4-pro/info#specs. [Accessed 18 April 2019].

[97] HuffPost Media, "HuffPost," [Online]. Available: https://www.huffpost.com/entry/dji-phantom-4-pro-review-a-new-standard-for-drones_b_59301e64e4b00afe556b0b7d. [Accessed 18 April 2019].

[98] R. Hajdin, M. Kušar, L. Paul, P. Linneberg, J. Amado and N. Tanasić, "COST Action TU1406 WG3 Technical Report Establishment of a Quality Control Plan," European Cooperation in Science and Technology (COST), 2018.

[99] D. Isailović, R. Hajdin and J. Matos, "Bridge quality control using Bayesian net," in *40th IABSE Symposium 2018: Tomorrow's Megastructures*, 2018.

[100] E. Masoud, A. Clarke-Sather and J. McConnell, "Lean Construction Applications for Bridge Inspection," 2017.

[101] R. S. Kirk and W. J. Mallett, "Highway Bridge Conditions: Issues for Congress," 2018.

[102] A. Zulfiqar, M. Cabieses, A. Mikhail and N. Khan, "Design of a Bridge Inspection System (BIS) to Reduce Time and Cost," 2014.

[103] Small Unmanned Aircraft System Aviation Rulemaking Committee, *Comprehensive Set of Recommendations for sUAS Regulatory Development,* 2009.

[104] Technion - Israel Institute of Technology, "SeeBridge," [Online]. Available: https://seebridge.net.technion.ac.il/. [Accessed 15 April 2020].

[105] T. Hoffmann, "SunCalc," [Online]. Available: www.suncalc.org. [Accessed 26 November 2018].

APPENDICES

Appendix 1: Bridge Inventory

A1.1 General information

Facility ID	14818
Facility code	R100-00-0337-010-MAPG
Facility name	Bridge over river Gročica
Affiliation type	1 (Bridge in the road body)
Composite facility ID	*None*
Road Label	R-100
Road name	BGD.(GL.POSTA)-SMEDEREVO1
Road section ID	337
Chainage (start)	*None*
Chainage (middle)	5 km + 124 m
Chainage (end)	*None*
Node ID	*None*
Node name	*None*
Node prong	*None*
Obstacle type and name	River Gročica
Closest township	Grocka
Cadastre No.	*None*
Municipality	Grocka
County	City of Belgrade
Region	City of Belgrade
State	Serbia
Possible detour	No
Microlocation name	Grocka
Latitude	*None*
Longitude	*None*
Position Z	*None*
Rank	*None*
Structural system	Simply supported beam
Designer	Unknown
Project date	*None*
Design code	Unknown
Contractor	Unknown
Construction year	*None*
Supervisor	Unknown
Manager	State Enterprise "BGD - Održavanje", Public Comunal Enterprise "Beogradput"
Project stored by	*None*
Group manager	*None*
File date	1.9.1999
Data prepared and processed by	N. Vukelic, Z. Milisic
Reconstructed facility ID	*None*
Note	"Chainage from video clip, state road II - 127"

A1.2 Description Elements

No.	Element Name	Quantity	Value	Unit	Description
130	Bridge type	*None*	*None*	*None*	Bridge
140	Spans	*None*	12	m	*None*
150	Spanned structure length	*None*	12.5	m	*None*
160	Overall structure length	*None*	26.3	m	*None*
170	Roadway width	*None*	7.1	m	*None*
180	Left sidewalk width	*None*	1.8	m	*None*
190	Right sidewalk width	*None*	1.8	m	*None*
210	Expansion joints type	*None*	*None*	*None*	Don't exist
220	Description of abutments	*None*	*None*	Piece	*None*
230	Description of piers	*None*	*None*	Piece	*None*
240	Opening	*None*	*None*	m	*None*
250	Drainage system	*None*	*None*	*None*	Free fall
260	Number of drainage pipes	*None*	0	Piece	*None*
270	Instalations	*None*	*None*	Piece	Electric and telephone
340	Foundation	*None*	*None*	*None*	Unknown
350	Shape of the bridge axis	*None*	*None*	*None*	Straigth
360	Shape of the cross section	*None*	*None*	*None*	*None*
370	Bridge skewness	*None*	0	Degree	Non skewed
380	Level line shape	*None*	*None*	%	< 2
390	Transversal roadway slope	*None*	*None*	%	two-way
410	Bearings type	*None*	*None*	*None*	*None*
420	Superstructure material	*None*	*None*	*None*	Reinforced concrete
430	Substructure material	*None*	*None*	*None*	Concrete
440	Roadway material	*None*	*None*	*None*	Asphalt
450	Sidewalk cover	*None*	*None*	m^2	*None*
460	Sidewalk dissociation type	*None*	*None*	*None*	*None*
470	Safety rail material & type	*None*	*None*	*None*	*None*
480	Guard rail material & type	*None*	*None*	*None*	Doesn't exist
490	Hydroinsulation	*None*	*None*	m^2	Unknown
500	Construction technology	*None*	*None*	*None*	Classic
540	Geological profile	*None*	*None*	*None*	Unknown

A1.3 Inspection Elements

No.	Abbr.	Element Name	Quantity	Unit	Importance Factor	Security Factor	Economic Factor
50	TEM	Foundations	*None*	*None*	11.3	1	1
55	TKS	Abutment foundations	*None*	Piece	11.3	1	1
60	TSS	Pier foundations	*None*	Piece	11.3	1	1
70	KST	Abutments with embedded wings	*None*	*None*	11.3	1	1
80	SSK	Free standing wings	*None*	*None*	8	1	1
90	SST	Piers	*None*	*None*	11.3	1	1
100	LEZ	Bearings	*None*	Piece	11.3	1	1
110	GNO	Main girders	*None*	Piece	11.3	1	1
120	PNO	Cross girders	*None*	Piece	11.3	1	1
130	SPR	Bracings	*None*	Piece	11.3	1	1
140	PIK	Deck slab and consoles	*None*	Piece	11.3	1	1
150	KOR	Occurence of renforcement and steel structure corrossion	*None*	m^2	8	1	1
160	HIZ	Hydroinsulation	*None*	m^2	5.65	1	1
170	DIL	Expansion jonts	*None*	m	5.65	1	1
180	KOL	Driveway	*None*	m^2	5.65	1	1
190	PRE	Watercourse or the area under the bridge	*None*	m^2	5.65	1	1
200	OUT	General impression	*None*	*None*	5.65	1	1
210	KLI	Cups	*None*	Piece	4	1	1
220	PRP	Approach slabs	*None*	Piece	4	1	1
230	KEG	Semi-rertaining abutment	*None*	Piece	4	1	1
240	OGR	Railings	*None*	m	4	1	1
250	IVI	Curbs	*None*	m	4	1	1
260	PST	Sidewalks	*None*	m^2	4	1	1
270	ODV	Drainage system	*None*	m	4	1	1
280	INS	Instalations	*None*	Piece	2	1	1
290	SIG	Signalization	*None*	*None*	2	1	1
300	POL	Position of the bridge in the network	*None*	*None*	2	1	1
310	GEO	Bridge geometry	*None*	*None*	2	1	1
320	SAO	Traffic load PGDS	*None*	Vehicles per day	2.82	1	1

Appendix 2: Inspection report

A2.1 Element inspection record

Facility Condition ID	2
Report No.	0260010000594
Inspection Date	3.10.2006
Inspection Type	Regular
Inspectors	M. Bebeic, B. Stoisavljevic
Responsible Person	*None*
Facility ID	14818
Facility code	R100-00-0337-010-MAPG
Facility name	Bridge over river Gročica
Rating	483.95
Condition category	Fair/Unfavourable
Rating grade	N9 Unfavourable (200-500)
Maintenance Type	Intensive Regular

55	**Abutment foundations**
Importance Factor	11.3
Condition Rating	1
Condition category	Good
Condition description	*None*

60	**Pier foundations**
Importance Factor	11.3
Condition Rating	0
Condition category	Don't exist
Condition description	*None*

70	**Abutments with embedded wings**
Importance Factor	11.3
Condition Rating	5
Condition category	Satisfactory
Condition description	"Signs of water penetrating into the concrete columns. Intervention is currently impossible"

80	**Free standing wings**
Importance Factor	8
Condition Rating	0
Condition category	Don't exist
Condition description	*None*

90	**Piers**
Importance Factor	11.3
Condition Rating	0

Condition category	Don't exist
Condition description	*None*

100	Bearings
Importance Factor	11.3
Condition Rating	0
Condition category	Unnecessary
Condition description	Directly supported

110	Main girders
Importance Factor	11.3
Condition Rating	10
Condition category	Unfavourable
Condition description	"Signs of water penetrating on lateral surfaces. Damaged reinforcement cover. Reinforcement corrossion"

Damages and intervention measures	
No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	2
Note	"Reconstruction of reinforcement cover"

120	Cross girders
Importance Factor	11.3
Condition Rating	1
Condition category	Good
Condition description	

130	Bracings
Importance Factor	11.3
Condition Rating	0
Condition category	Don't exist
Condition description	

140	Deck slab and consoles
Importance Factor	11.3
Condition Rating	5
Condition category	Satisfactory
Condition description	"Console edge damaged by water and frost"

Damages and intervention measures	
No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined

Work Quantity	2
Note	"Reconstruction of the console (concrete MB30)"

160 Hydroinsulation

Importance Factor	5.65
Condition Rating	3
Condition category	Satisfactory
Condition description	"Signs of water penetrating on the consoles"

170 Expansion jonts

Importance Factor	5.65
Condition Rating	8
Condition category	Serious
Condition description	"Don't exist. Cracks on the connection with the road body. Necessary to install new expansion joints"

Damages and intervention measures

No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	12
Note	"Concrete cutting and installation of expansion joints"

180 Driveway

Importance Factor	5.65
Condition Rating	2
Condition category	Satisfactory
Condition description	"Cracks on the connection with the road body. Reconstruction works described in the Expansion joints section"

190 Watercourse or the area under the bridge

Importance Factor	5.65
Condition Rating	1
Condition category	Good
Condition description	*None*

200 General impression

Importance Factor	5.65
Condition Rating	2
Condition category	Satisfactory
Condition description	"Bridge is constructed about 70 years ago. After the reconstruction desribed in the report, with regular maintenance, an estimated life expectancy of the bridge is additional 40 years"

210 Cups

Importance Factor	5.65

Condition Rating	1
Condition category	Good
Condition description	*None*

220 Approach slabs

Importance Factor	4
Condition Rating	1
Condition category	Good
Condition description	*None*

230 Semi-rertaining abutment

Importance Factor	4
Condition Rating	2
Condition category	Satisfactory
Condition description	"Abutments covered by vegetation"

Damages and intervention measures

No.	1
Damage Name	Vegetation on the abutments
Maintenance	Urgent
ID	14
Position Code	14
Position Name	Vegetation removal
Work Quantity	100 m²
Note	*None*

240 Railings

Importance Factor	4
Condition Rating	1
Condition category	Good
Condition description	"Safety rail in good condition."

250 Curbs

Importance Factor	4
Condition Rating	5
Condition category	Serious
Condition description	"Need for total asphalt pouring. Lifting necessary."

Damages and intervention measures

No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	40
Note	"Lifting"

260 Sidewalks

Importance Factor	4
Condition Rating	5

Condition category	Serious
Condition description	"Not lifted enough and covered with waste. Damaged cornice."

Damages and intervention measures

No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	12
Note	"/M3/ concrete for lifting"
No.	3
Damage Name	Sidewalk without an asphalt cover
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	40
Note	"Installation of pipes"
ID	16
Position Code	16
Position Name	Asphalt laying on the sidewalk
Work Quantity	50 m²
Note	*None*

270 Drainage system

Importance Factor	4
Condition Rating	4
Condition category	Poor
Condition description	"Drains out of order."

Damages and intervention measures

No.	1
Damage Name	*None*
Maintenance	Urgent
ID	0
Position Code	0
Position Name	Undefined
Work Quantity	2
Note	"Installation of drains"

280 Instalations

Importance Factor	2
Condition Rating	2
Condition category	Satisfactory
Condition description	"Corroded installation cover"

290 Signalization

Importance Factor	2
Condition Rating	1

Condition category	Good
Condition description	*None*

300	**Position of the bridge in the network**
Importance Factor	2
Condition Rating	3
Condition category	M:30 < La < 50 km, R:20 < La < 50 km
Condition description	*None*

310	**Bridge geometry**
Importance Factor	2
Condition Rating	1
Condition category	Geometry tuned
Condition description	*None*

320	**Traffic load PGDS**
Importance Factor	2
Condition Rating	1
Condition category	PGDS < 2500
Condition description	*None*

50	**Foundations**
Importance Factor	11.3
Condition Rating	1
Condition category	Good
Condition description	"Without signs of damage"

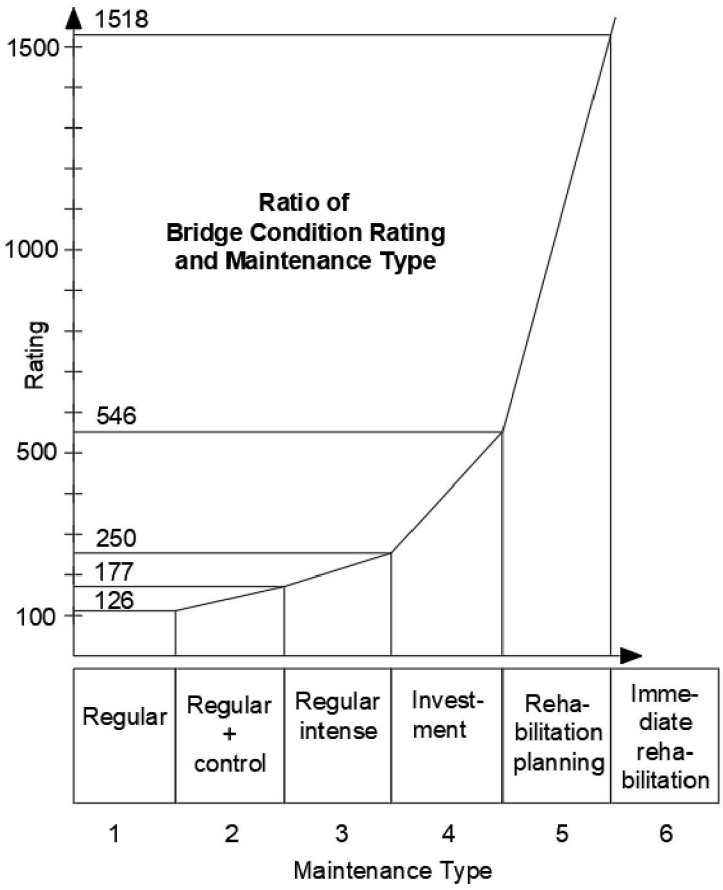
A2.2 Report summary

No.	Element	Importance factor	Condition Rating	Characteristic numbers			
				R_1	R_2	R_3	R_4
1	Abutment foundations	11.3	1	11.3			
2	Pier foundations	11.3	0	0			
3	Abutments with embedded wings	11.3	5	56.5			
4	Free standing wings	8	0		0		
5	Piers	11.3	0	0			
6	Bearings	11.3	0	0			
7	Main girders	11.3	10	113			
8	Cross girders	11.3	1	11.3			
9	Bracings	11.3	0	0			
10	Deck slab and consoles	11.3	5	56.5			
11	Hydroinsulation	5.65	3		16.95		
12	Expansion jonts	5.65	8		45.2		
13	Driveway	5.65	2			11.3	

14	**Watercourse or the area under the bridge**	5.65	1		5.65			
15	**General impression**	5.65	2		11.3			
16	**Cups**	5.65	1			5.65		
17	**Approach slabs**	4	1			4		
18	**Semi-rertaining abutment**	4	2			8		
19	**Railings**	4	1			4		
20	**Curbs**	4	5			20		
21	**Sidewalks**	4	5			20		
22	**Drainage system**	4	4		16			
23	**Instalations**	2	2			4		
24	**Signalization**	2	1			2		
25	**Position of the bridge in the network**	2	3				6	
26	**Bridge geometry**	2	1				2	
27	**Traffic load PGDS**	2	1				2	
				248.6 R_1	95.1 R_2	78.95 R_3	10 R_4	432.65 R

Condition rating categorization

Type of work	R_1	R_2	R_3	R_4	R
Regular maintenance	83	29	39	29	
Regular maintenance + control	115	40	54	41	
Regular intense maintenance	161	57	76	56	432.65
Investment maintenance	221	78	104	77	
Rehabilitation planning	322	113	151	114	
Immediate rehabilitation	>322	>113	>151	>114	
	Bridge safety (based on inspection findings)	Risk of further structural damage	Functionality	Final prioritization preference	Rehabilitation planning



Overall condition rating: 4 (investment maintenance).

Appendix 3: Solar data for Grocka

The following Geo and Solar Data for Grocka is obtained from [105].

Geo data for the Location		
Height	78 m	
Latitude	N 44°40'21.6"	44.67267°
Longitude	N 20°43'7.38"	20.71872°
Timezone	Europe/Belgrade CEST	

Solar data for the Location		
Time	August 10th, 12:00 UTC +2	August 16th, 7:00 UTC +2
Dawn	05:01:02	05:08:45
Sunrise	05:33:03	05:40:04
Sun peak level	12:42:30	12:41:25
Sunset	19:51:10	19:41:58
Dusk	20:23:03	20:13:10
Duration	14h18m7s	14h1m54s
Altitude	59.49°	12.95°
Azimuth	159.51°	83.29°
Shadow length at an object level: 6 m	3.54	26.1

Sun position and shadow disposition on the bridge:

August 10th, 12:00

August 16th, 7:00

Author's biography

Dušan Isailović is born on May 15th, 1989, in Kragujevac, Serbia, where he finished primary school and gymnasium. In 2009, Dušan enrolles Faculty of Civil Engineering, at the University of Belgrade. In 2013, he gains Bachelor degree, and in 2015 he gains Master degree, at the Department of Structural Engineering. In 2015, Dušan enrolles the PhD studies at the Faculty Civil Engineering.

During the period from July until October of 2014, Dušan was employed by the company "Tankmont d.o.o.", Belgrade. He worked there as a site engineer, organizing and controlling the construction of four oil reservoirs in Purpe, Russian Federation.

From August 2015 until March 2016, he was employed by the company "Centroprojekt d.o.o.", Belgrade, as a structural designer. He was enrolled in the structural design, supervision and revision of several domestic structural design projects.

In January 2016, Dušan has been nominated by the Academic-Scientific Council of the Faculty of Civil Engineering in Belgrade as a Teaching Assistant - PhD Student in the scientific field of Information Technologies in Civil Engineering and Geodesy. Since then, Dušan is engaged teaching exercises in several courses at the Department for Construction Project Managament.

Since 2018, Dušan Isailović is a member of the association BIM Serbia.

Dušan Isailović is an author of one journal paper and several conference articles on Building Information Modeling and Infrastructure Asset Management.

Биографија аутора

Душан Исаиловић је рођен 15. маја 1989. године у Крагујевцу, где је завршио основну и средњу школу. 2009. године уписује Грађевински факултет Универзитета у Београду. 2013. године завршава основне, а 2015 мастер академске студије, на Одсеку за уконструкције. Докторске студије на Грађевинском факултету Универзитета у Београду душан је уписао 2015. године.

У периоду од јула до октобра 2014. године био је запослен у у фирми „Танкмонт“ д.о.о. Београд. За то време, радио је као инжењер сарадник на организацији и контроли радова на извођењу четири нафтна резервоара у Пурпеу, Руска Федерација.

Од августа 2015. до марта 2016. године био је запослен у у фирми „Центропројект“ д.о.о. Београд на позицији пројектанта бетонских конструкција. За то време, учествовао је у пројектовању, надзору и ревизији пројеката неколико објеката у Србији и иностранству.

У јануару 2016. године одлуком Изборног већа Грађевинског факултета Универзитета у Београду, изабран је, а у новембру 2018 је поново изабран у звање асистента-студента докторских студија за ужу научну област Информационе технологије у грађевинарству и геодезији. Од тада учествује у настави на више предмета у оквиру Катедре за управљање пројектима у грађевинарству.

Од 2018. године, Душан Исаиловић је члан удружења BIM Србија.

Душан Исаиловић је аутор једног рада у међународном научном часопису и више радова који су објављени у зборницима радова са међународних конференција који се баве информационим моделирањем грађевинских објеката и управљањем инфраструктуре.

Изјава о ауторству

Име и презиме аутора Душан Исаиловић

Број индекса 910/15

Изјављујем

да је докторска дисертација под насловом

Digital representation of as-damaged reinforced concrete bridges

Дигитални приказ оштећених армиранобетонских мостова

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

_____

У Београду,

12.5.2020.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Душан Исаиловић

Број индекса 910/15

Студијски програм Грађевинарство

Наслов рада Digital representation of as-damaged reinforced concrete bridges

Дигитални приказ оштећених армиранобетонских мостова

Ментор Проф. др Раде Хајдин

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

_____

У Београду,

12.5.2020.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Digital representation of as-damaged reinforced concrete bridges

Дигитални приказ оштећених армиранобетонских мостова

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци. Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

_____

У Београду,

12.5.2020.

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.