Silesian University of Technology

DOCTORAL DISSERTATION

in the discipline: Civil Engineering, Geodesy, and Transport

Digital twins of bridges: establishing principles of virtualization with practical use cases

author:

Kamil Korus

supervisor:

Professor Marek Salamak PhD, DSc

additional supervisor:

Jan Winkler PhD

March 2023

Gliwice, Poland

Contents

List of Abbreviations ... ix

List of Figures ... xi

List of Tables ... xiii

List of Listings ... xv

1 Introduction ... 1

1.1 Motivations ... 1

1.1.1 Adapt (or shape) to thrive ... 1

1.1.2 Connectors of people and data ... 2

1.2 Aims of the dissertation ... 3

1.3 Organization of the dissertation ... 3

2 Digital twins – the next wave in simulations ... 5

2.1 Introduction ... 5

2.2 The evolving idea of digital twins ... 5

2.2.1 Origins ... 5

2.2.2 Definitions ... 7

2.3 Digital twins overview ... 11

2.3.1 General ... 12

2.3.2 Civil engineering ... 15

2.3.3 Bridges ... 17

2.4 Conclusions ... 22

3 The concept of digital twins of bridges ... 25

3.1 Introduction ... 25

3.1.1 Specificity of bridges ... 26

3.1.2 Practitioners' view ... 27

3.1.3 Evolution into civil engineering digital twinning ... 28

3.1.4 The approach to defining the framework ... 29

3.2 Techniques of civil engineering digital twinning ... 29

3.2.1 Building Information Modeling (BIM) ... 30
3.2.1.1 Maturity levels ... 30
3.2.1.2 Tools and methodologies: Little BIM/Big BIM, Closed BIM/Open BIM ... 31
3.2.1.3 BIM dimensions ... 32
3.2.1.4 Level of Development (LOD) ... 32
3.2.1.5 Common Data Environments (CDE) ... 33
3.2.1.6 Standardization ... 33
3.2.1.7 BIM model is not a digital twin ... 34
3.2.2 Industry Foundation Classes (IFC) ... 36
3.2.2.1 Origins and development ... 36
3.2.2.2 Multi-industrial base ... 37
3.2.2.3 IFC architecture ... 37
3.2.2.4 IFC for bridges and infrastructure ... 42
3.2.2.5 Modeling IFC ... 44
3.2.2.6 Horizon of advances ... 49
3.2.3 Structural Health Monitoring (SHM) ... 50
3.2.3.1 SHM in the digital twinning context ... 51
3.2.4 Artificial Intelligence (AI) ... 51
3.2.4.1 AI in civil engineering ... 52
3.2.4.2 AI in the digital twinning context ... 52
3.2.5 Visual programming ... 52
3.2.5.1 Visual programming in the digital twinning context ... 53
3.2.6 Point clouds ... 53
3.2.6.1 Point clouds automation ... 54
3.2.6.2 Point clouds in the digital twinning context ... 56
3.3 Notions towards digital twinning of bridges ... 56
3.3.1 Digital twins allow business thriving in the future ... 57
3.3.2 Digital twins’ frameworks should be object-specific ... 57
3.3.3 Ultra-high fidelity is not always required ... 58

3.3.4 Data synchronization does not need to be perfect 58
3.3.5 Digital twin should not neglect any data freely 59
3.3.6 Digital twin provides data interfaces 59
3.3.7 Digital twin includes processes 60
3.3.8 Digital twin is intelligent, autonomous 60
3.3.9 Digital twin is a system 61
3.3.10 Digital twins widen empirical science 62
3.3.11 Digital twins must provide practical benefits 62
3.3.12 It is the right time to establish digital twinning frameworks 63
3.4 Characteristics of digital twins of bridges 64
3.4.1 Actual reflection: fidelity and synchronization 64
3.4.2 Intelligence and autonomy 65
3.4.3 Interaction 66
3.4.4 Interoperability 68
3.4.5 Modularity 69
3.4.6 Expansibility and scalability 70
3.4.7 Accessibility and security 71
3.4.8 Uniqueness 72
3.5 Conclusions: the need for practical use cases 73
4 Optimization using visual programming and genetic algorithm 75
4.1 Introduction 75
4.1.1 Optimization techniques for bridges 75
4.1.2 Genetic algorithms 77
4.2 Methods 77
4.2.1 FEM and visual programming: deploying a package 77
4.2.2 Generating models 79
4.2.3 Generative design optimization 86
4.3 Optimization process and results 87
4.3.1 Reference construction analysis 88

4.3.2 Optimization assumptions 90
4.3.3 Cross-sections optimization 93
4.3.4 Geometry optimization 94
4.3.5 Geometry and cross-sections optimization 95
4.3.6 Geometry, cross-sections, and material optimization 97
4.4 Conclusions 98
5 Generating synthetic data for point clouds’ machine learning automation 99
5.1 Introduction 99
5.1.1 Need for point clouds’ data 100
5.1.2 Generating synthetic point clouds 101
5.1.3 Requirements of a versatile simulation tool 104
5.2 Visual programming synthetic point cloud simulator 104
5.2.1 Characteristics and operation 104
5.2.1.1 Emulating real scanners operation: ray tracing 104
5.2.1.2 Applicability: visual programming and use of non-specific models 105
5.2.1.3 Fully-automated labeling: use of model semantic data 106
5.2.1.4 Emulating scanning errors 106
5.2.1.5 Introducing noise 107
5.2.1.6 Collecting additional point cloud properties 107
5.2.2 Features and workflow 108
5.2.2.1 Model (a pre-stage) 108
5.2.2.2 Mesh 108
5.2.2.3 Scanner 110
5.2.2.4 PointCloud 110
5.3 Validation for machine learning 113
5.3.1 Data validation approach 114
5.3.2 Real data 114
5.3.3 Synthetic data 115
5.3.4 Machine learning algorithm 116

5.4 Conclusions ... 118

6 Conclusions ... 121

6.1 Summary ... 121

6.2 Limitations and future work ... 122

6.2.1 Bridge digital twinning framework ... 122

6.2.2 Optimization using visual programming and genetic algorithm ... 123

6.2.3 Generating synthetic data for point clouds' machine learning automation ... 123

6.3 Concluding remarks ... 125

Appendix A. Digital twinning overview bridge-oriented articles ... 127

Appendix B. Modeling IFC ... 131

Appendix C. Results of the optimization using visual programming and genetic algorithm ... 133

Appendix D. DynamoPCSim methods ... 136

References ... 151

Acknowledgments ... 169

Abstract ... 171

Extended abstract ... 173

Streszczenie (PL) ... 187

Poszerzone streszczenie (PL) ... 189

List of Abbreviations

AEC	Architecture, Engineering, and Construction
AI	Artificial intelligence
ALS	Airborne laser scanning
API	Application Programming Interface
AR	Augmented Reality
BCF	BIM Collaboration Format
BIM	Building Information Modeling
BMS	Bridge management system
bSDD	buildingSMART Data Dictionary
CDE	Common Data Environment
COBie	Construction Operations Building Information Exchange
DT	Digital twin
FEM	Finite element modeling
FM	Facility Management
GIS	Geographic Information System
GUID	Globally Unique Identifier
IDM	Information Delivery Manual
IDS	Information Delivery Specification
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
LCA	Life Cycle Assessment
LiDAR	Light Detection and Ranging
LOD	Level of Development / Level of Detail
ML	Machine learning
MLS	Mobile laser scanning
MR	Mixed Reality
MVD	Model View Definition
REST	Representational state transfer
SaaS	Software as a Service
SAR	Synthetic aperture radar

SHM	Structural health monitoring
TLS	Terrestrial laser scanning
UAV	Unmanned aerial vehicle
VP	Visual programming
VPL	Visual programming language
VR	Virtual Reality
XML	Extensible Markup Language

List of Figures

Fig. 2-1. Digital twinning research interest (Scopus database; access: 15.01.2023) ... 12
Fig. 2-2. Digital twinning publications areas (Scopus database; access: 15.01.2023) ... 13
Fig. 2-3. Digital twinning publications map (Scopus database; access: 15.01.2023) ... 13
Fig. 2-4. Digital twinning bridge-oriented research interest (Scopus database; access: 15.01.2023) ... 17
Fig. 2-5. Digital twinning bridge-oriented publications map (Scopus database; access: 15.01.2023) ... 18
Fig. 2-6. Digital twinning bridge-oriented publications word cloud ... 22
Fig. 3-1. A bridge digital twin vision ... 26
Fig. 3-2. Industry Foundation Classes architecture layers and parts [159] ... 38
Fig. 3-3. Modeling IFC: superstructure's geometry and characteristics ... 49
Fig. 3-4. Scheme of a visual programming language script ... 53
Fig. 3-5. Point clouds' segmentation types: a) segmentation, b) instance-based semantic segmentation, c) class-based semantic segmentation ... 55
Fig. 3-6. Parallel, superior, and subordinate relations between digital twins ... 67
Fig. 3-7. Modular structure of a bridge digital twin ... 70
Fig. 3-8. Management of model elements identifiers in BIM software (Revit): a) BIM software model element; b) matching IFC model element; c) modified BIM software model element; d) matching modified IFC model element ... 73
Fig. 4-1. Fragment of Dynamo script with Dynamo FEM package methods ... 79
Fig. 4-2. Script operation flowchart ... 80
Fig. 4-3. Bar model and its bars groups ... 80
Fig. 4-4. Support system schema ... 81
Fig. 4-5. The schema of live load cases applied to model ... 83
Fig. 4-6. Visualizations of a) basic programming environment geometries and b) generated BIM Model ... 86
Fig. 4-7. Generative design optimization system operation flowchart ... 87
Fig. 4-8. Reference construction selected for the optimization process ... 88
Fig. 4-9. Optimization cases and their adjustable parameters ... 90
Fig. 4-10. Results of the ranges setting analysis ... 92
Fig. 4-11. Cross-sections optimization results ... 93
Fig. 4-12. Comparison of the reference and optimized constructions' cross-sections ... 94
Fig. 4-13. Geometry optimization results ... 95
Fig. 4-14. Geometry and cross-sections optimization results ... 96
Fig. 4-15. Geometry, cross-sections, and material optimization results ... 97

Fig. 5-1. Ray tracing implemented with the possibility of limiting the maximum number of reflections: a) no reflections, b) limited reflections, c) unlimited reflections 105
Fig. 5-2. DynamoPCSim flowchart 108
Fig. 5-3. Mesh creation example 109
Fig. 5-4. Examples of MeshElementCreationStrategies implementations: a) ByDetailedCreation, b) SelectFacesWithBiggestAreaByParameter, c) SelectFacesWithBiggestAreaByParameter and SkipCurvedFaces 109
Fig. 5-5. Synthetic point clouds examples: a) scanned model; b) scanning without reflections c) scanning with reflections 111
Fig. 5-6. Joining synthetic point clouds example: a), b), c) single-locations point clouds, d) merged point cloud 112
Fig. 5-7. Transforming synthetic point clouds examples: a) non-transformed point cloud, b) AddRandomNormalTranslation, c) DeletePointsWithProbability, d) AddNoisePoints 113
Fig. 5-8. Point cloud used for the validation experiment: a) whole point cloud, b) ColumnCap, c) ColumnPier, d) Car, e) Terrain segments. 115
Fig. 5-9. Model for generating synthetic point clouds' dataset 115
Fig. 5-10. Examples of generated synthetic point clouds for the validation experiment 116
Fig. 5-11. Confusion matrices (with graphical outcomes) of predictions on a) the real point cloud and b) a synthetic point cloud 118

List of Tables

Table 2-1. Digital twins definitions 7
Table 3-1. Level of Development specification (based on [148]) 32
Table 3-2. BIM standards 33
Table 3-3. Exemplary IFC *Pset*s (based on [160]) 41
Table 3-4. IFC formats (based on [155]) 49
Table 4-1. Geometrical input parameters 81
Table 4-2. Input geometry parameters of reference construction 88
Table 4-3. Output parameters of reference construction calculations 89
Table 4-4. Ranges and steps of cross-sections' parameters 91
Table 4-5. Ranges and steps of *Arch rise* and *Hangers number* parameters 92
Table 5-1. Z+F Imager 5010 scanner specification [357] 114
Table 5-2. Performance of the trained PointNet model on a) real and b) synthetic validation batches and c) their comparison 116

Table A-1 Digital twinning bridge-oriented publications 127

Table C-1. Geometry optimization final parameters and outputs 133
Table C-2. Geometry and cross-sections optimization final parameters and outputs 133
Table C-3. Geometry, cross-sections, and materials optimization final parameters and outputs 134

Table D-1. DynamoPCSim *Mesh* main classes and methods 136
Table D-2. DynamoPCSim *Scanner* main classes and methods 142
Table D-3. DynamoPCSim *PointCloud* main classes and methods 144

List of Listings

Listing 3-1. Modeling IFC: header 45
Listing 3-2. Modeling IFC: project initiation and units 45
Listing 3-3. Modeling IFC: geometric representation contexts 46
Listing 3-4. Modeling IFC: basic geometry primitives 46
Listing 3-5. Modeling IFC: spatial elements 46
Listing 3-6. Modeling IFC: attaching material 47
Listing 3-7. Modeling IFC: alignment 47
Listing 3-8. Modeling IFC: alignment 47
Listing 3-9. Modeling IFC: profile (superstructure cross-section) 48
Listing 3-10. Modeling IFC: geometry representation (superstructure geometry) 48
Listing 3-11. Modeling IFC: external data source (performance history reference) 49
Listing 3-12. Concept of a digital twinning standardization nested inheritance scheme 69
Listing 5-1. Saved point cloud in the XYZ format 113

Listing B-1. Modeling IFC example 131

1 Introduction

1.1 Motivations

1.1.1 Adapt (or shape) to thrive

The world is always changing. This statement – although trivial – is a purpose of digital twinning research. The statement can be expanded. The world is always changing at an increasing pace. This is being validated especially in the last decade with the digital transformation following the concept of Industry 4.0.

Changes divide people into enthusiasts and pessimists, not only for moral and cultural concepts but also for technical innovations. During the Industrial Revolutions, people perceived the coming automation differently: some were excited about more comfortable lives; others were afraid of losing jobs and struggling in the new environment. As we know today, the groups that took the opportunities of technical advances shaped our world.

The ongoing digitization comes with many shifts: reality is being virtualized into a parallel world; people are constantly connected to the web, producing thoroughly analyzed data; artificial intelligence is taking over tasks formerly reserved for humans. The changes bring excitement and fear – both feelings are justified. However – following the maxim "you must develop not to stay in place" – people and industries, whether enthusiasts or pessimists, should be prepared for the changes of the digitization of our lives.

When juxtaposing "bridges" with terms like "artificial intelligence" or "virtual reality", bridges seem to be a technological relict of ancient times. After all, we are talking about static, monumental structures that, once built, can be left alone to stand for decades. But is it so? Not exactly. Bridges are constantly reacting to varying environmental conditions and forces. The resulting gradual degradation of bridges' condition can be unnoticed by users but is essential for managers and inspectors. Because of that, bridges must be monitored and maintained. Also, with new materials, structural solutions, building techniques, and complex equipment, bridges are still an area of civil engineering innovations. Nonetheless, the modern world's conditions push bridges to take the next steps into technical advances. Civil engineering is transforming to adapt to sustainable development goals, ecological aims, and a growing population.

To thrive, one must adapt to changes – especially in technological contexts. But adapting does not indicate a passive and mindless following. It may also mean shaping. In the digital world, industries become interconnected. This cooperation requires new principles and paradigms. Digital twins, a concept perceived as the key component of Industry 4.0, are becoming this multi-industrial paradigm. Digital twinning is already being applied, but its principles are still forming. The principles should be formed not only by programmers or IT engineers but, in the case of bridges, also by the civil engineering community. In that way, civil engineers will contribute to forming the principles of worldwide digitization, and digital twins of bridges will address the needs of designers, engineers, and managers of infrastructure.

1.1.2 Connectors of people and data

During the rainy season in northern Nicaragua, local rivers become unpassable, cutting off small villages from market, healthcare, and education. Villagers cannot trade crops nor access work outside of the village – floodings rule the villagers' day-to-day lives. In 2015, the organization Bridges to Prosperity constructed six trail bridges, making six villages independent of the flood. It invoked a chain of welfare implications [1]: dwellers' incomes increased and stabilized; the rate of women employed outside of their village nearly doubled; due to reliable access to the market, farmers stored fewer harvests, selling more and reinvesting in their farms, empowering development.

The example shows bridges' influence on the day-to-day lives of individuals. Bridges' impact is also visible on the macro scale, which can be expressed with numbers; the numbers also reveal problems of the existing infrastructure. Only in the USA, there are 619 thousand of bridges, of which 224 thousand require major repair work or replacement [2]. The problem is expanded by its economic scale: the estimated cost of identified repairs is USD 260 billion – a value comparable to the gross domestic product of the Czech Republic (USD 282 billion), New Zealand (USD 250 billion), or Portugal (USD 250 billion).

Through the ages, bridges have followed human development, transforming from fallen trees into technical wonders and becoming societies' wealth. The number and complexity of bridges increased the importance of managing the growing and also aging infrastructure network. It resulted in bridge management systems and standardization of maintenance, as well as processes of processing facilities' data. These should be adjusted to the novel technical capabilities of digital transformation. However, infrastructure managers still use older practices, despite their flaws: improper data management diminishes efficiency; condition assessment relies on periodic human inspections, displaying risks of omitting defects; aging infrastructure is unproperly maintained, leading to economic losses and posing the danger of catastrophes. The problems of existing infrastructure convince that civil engineering needs the plan to ensure resilience and safety through the entire lifecycle. The plan must consider the specificity of bridges and the requirements of today's world and also benefit from the technical advances.

Bridges performed various roles through the ages. They have been places of living (e.g., Ponte Vecchio in Florence), meeting points (e.g., Khaju Bridge in Isfahan), and supplies providers (e.g., Pont du Gard near Nîmes). They are a cultural and technical heritage but can also act as evidence of liaison with nature (e.g., living root bridges in India). Bridges have always been strategic components: Xerxes' pontoon bridges enabled the second Persian invasion of Greece in 480 BC; capturing the Ludendorff Bridge in march 1945 allowed the Allies to cross the Rhine, the last German natural defense barrier. The specific contribution of bridges is also needed in the era of digital transformation. In the digitized world, bridges evolve from static constructions into services providing multidimensional connection: physical – of communities, and digital – of data. This shift results from the requirements of facilities' managers and the need to ensure the safety and comfort of users. This forces connectivity and integration in the digital dimension, especially in acquiring, processing, and sharing data. Bridges will cooperate with other objects to empower global holistic management and to constitute the digital networks of smart cities. To enable that, we need a multi-industrial paradigm – the world is shifting towards digital twins.

1.2 Aims of the dissertation

The aim of this dissertation is to provide a foundation for creating and utilizing digital twins of bridges – the virtual counterparts of objects in their whole lifecycle. The proposed concept addresses the general principles of digital twins, the specificity of bridges, and the practices of the civil engineering industry. The practicality of the proposed solutions is upheld with novel use cases utilizing the techniques identified as components of the proposed digital twinning concept.

To reach the general aim of the dissertation, several intermediate aims have been realized.

- Assessment of the current stage of development of digital twins through the study of literature and analysis of the implementations in bridges.
- Identification of the primary techniques to create and utilize digital twins of bridges.
- The proposal of the primary characteristics of digital twins of bridges regarding their lifecycle.
- The implementation of a practical integration of BIM and FEM models in a visual programming environment, aiming to create a partial digital twin in the design phase with the use of an optimization algorithm to automate the design process.
- The implementation of a technique to generate synthetic point clouds, aiming to create and update geometric models of digital twins in the operational phase with machine learning algorithms.

The listed intermediate aims undoubtedly do not fulfill all the general digital twins' requirements defined in the literature. However, digital twins, in the current development stage, are still a forming concept, continually enhanced with new techniques and solutions. The approach presented in the dissertation enables the implementation of digital twins in an evolutionary way. Digital twins do not have to be complete and perfect from the beginning; the concept should mature naturally, alluring practitioners with its benefits. Other important factors are the limitations of time and the scope of the dissertation. The proposed description of the digital twins' characteristics states the primary principles, not the details of technical implementations. Also, the presented techniques for creating and utilizing digital twins are limited to show the potential benefits. Used technologies and solutions regard the selected needs of the phases of design (automation and optimization of the design processes) and maintenance (changes of the facilities' geometries in time). Nonetheless, the use case of generating the synthetic point clouds does not describe the whole process of creation and update of the models with machine learning algorithms but focuses on the first key step to the implementation of such solutions – the acquisition of data. Future work will regard the creation of machine learning algorithms to create and utilize digital twins.

1.3 Organization of the dissertation

Chapter 2: *Digital twins – the next wave in simulations* analyzes the idea of digital twinning to asses its stage of development. The analysis starts with a brief history (section 2.2.1) and definitions (section 2.2.2) of digital twins showing their evolution and maturing. Following is the review of digital twinning concepts and applications in general (section 2.3.1), civil engineering (section 2.3.2), and bridge (section 2.3.3) contexts.

Chapter 3: *The concept of digital twins of bridges* introduces the proposed foundations for digital twins of bridges. It starts with the analysis of the specificity of bridges (section 3.1.1) and the demands of civil engineering practitioners (section 3.1.2), which must be regarded for a suited, practical solution. Section 3.1.3 describes the proposed approach to civil engineering digital twinning as the evolution of current techniques. Section 3.2 identifies the primary techniques and places them in the digital twinning concept. Section 3.3 presents notions and guidelines for the digital twinning of bridges. Section 3.4 presents the proposed characteristics of digital twins of bridges.

Chapter 4: *Optimization using visual programming and genetic algorithm* introduces a practical use case technique to integrate BIM and FEM models in a visual programming environment, aiming to create a partial digital twin in the design phase. The technique also implements an optimization algorithm to automate the geometrical optimization of bridges' structures. The solution is evaluated on a real-bridge case study.

Chapter 5: *Generating synthetic data for point clouds' machine learning automation* introduces another practical use case: synthetic point clouds' simulator for creating machine learning datasets. Focusing on acquiring proper data for training machine learning algorithms, the solution is the first step in the process of automated generating and updating geometrical models of digital twins, and finally monitoring geometrical changes in time. The solution is evaluated with a neural network trained on the generated synthetic data and tested on a real point cloud.

Chapter 6: *Conclusions* summarizes and concludes the research presented in this dissertation, identifies limitations and states the paths for future work.

.

2 Digital twins – the next wave in simulations

2.1 Introduction

Digital twins are virtual counterparts of objects throughout their entire lifecycle. First adopted by NASA [3], today are being implemented cross-industrially, becoming a component of the digital transformation. The digital twin market is estimated to grow from USD 3.2 billion in 2020 to USD 184.5 billion in 2030 [4], giving a compound annual growth rate of 50%.

A digital twin, as a model coexisting with an object, should be initialized on a design stage, first reflecting the object's idea. With advances in design, it expands into a high-fidelity central data-management system. Then, it is used to plan and optimize production. Utilizing models for simulation and visualization during design and production is already a multi-industrial practice. But digital twinning aims to provide benefits through the entire object's lifecycle. Due to synchronization with the physical counterpart also on the operational stage, a digital twin exceeds the capabilities of isolated models. The virtual and physical counterparts collaborate. Data from the physical object update the virtual system. The virtual system affects the operation of the physical object.

The digital twin collects data to map processes. It registers how the object reacts to different conditions and how the reactions alter over time. Analyses of such big data lead to a better understanding of the object operation. The data also allows training intelligent algorithms to detect patterns and anomalies, evolving the digital twin into a system of growing autonomy in decision-making. The model reflecting the actual state of an element allows extended reliable simulations. Since digital twinning is a cross-sectoral framework, unified data systems allow holistic management, considering relationships of cooperating elements. It leads to global optimizations of economic, ecological, and social costs, keeping the safety of the objects.

The described concept of digital twins may have been perceived as a technological utopia only a few years ago. Today, however, many of these ideas are already being applied in academia and industry. Digital twinning is still a forming idea, with many ambiguities to resolve. Nonetheless, it has already started pushing the vision of multi-industrial digital cooperation into practice. Digital twinning starts to be a practical tool, described as the next wave in simulation [5].

This chapter analyzes the idea of digital twinning. The analysis starts with a brief history (section 2.2.1) and definitions (section 2.2.2) of digital twins showing their evolution and maturing. Following is the review of digital twinning concepts and applications in general (section 2.3.1), civil engineering (section 2.3.2), and bridge (section 2.3.3) contexts.

2.2 The evolving idea of digital twins

2.2.1 Origins

The history of *technical twinning* started with the Apollo XIII mission in 1970. NASA built physical simulators to mimic the behavior of real spacecraft. It enabled not only training before the start. The simulators exposed their value also during the mission, after an explosion in a spacecraft's module. With the simulators, engineers on Earth could have mimicked the malfunction, understood it, and prepared a rescue plan. The Apollo XIII simulators, physical objects, are an example of *technical twinning*, the precursor of *digital twinning*.

The physical simulators, although powerful, have been reserved for exceptional cases – not every project, especially if focused on individual non-mass production, could have afforded to build physical replicas of a designed object. However, with the advent of computer techniques, simulations migrated to the digital space, to computers, becoming an essential engineering tool. This opened the gates for digital twins.

In 2002 Michael Grieves presented the Conceptual Ideal for Product Lifecycle Management [6]. The Conceptual ideal for PLM comprised digital twinning foundations – real and virtual spaces, as well as linking data from real to virtual and from virtual to real – and is considered the origin of digital twins. The same author presented Mirrored Space Model in 2005 [7] and Mirroring Model in 2006 [8], both leveraging the real-virtual connection idea.

The term *digital twin* has been used in the NASA roadmaps ("Modeling, Simulation, Information Technology & Processing Roadmap – Technology Area 11" [3] and "Materials, Structures, Mechanical Systems, and Manufacturing Roadmap - Technology Area 12" [9]) published in November 2010. The digital twin has been defined as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin", restricting the idea to spacecraft.

(Although NASA roadmaps are considered the first public introductors of digital twins, a profound literature review encloses an interesting case not present in the digital twins' history descriptions. In June 2010 (5 months before the NASA roadmaps), Janina Puig and Jaume Duran presented a manuscript entitled "Digital Twins" at the *4th International Multi-Conference on Society, Cybernetics and Informatics*. They described an idea of a digital twin of a person in the internet environment. The digital twin was to be created by a photo and measurements of a person and then used on social media. The idea lacks the characteristics of a digital twin in a technical sense, and the article is not popular, having only one citation in January 2023. However, the direct use of the digital twin term seems to justify including the case here, if only as a curious fact.)

In 2011, the first digital twinning scientific articles were published: "Condidtion-based Maintenance Plus Structural Integrity (CBM+SI) & the Airframe Digital Twin" [10] and "Reengineering aircraft structural life prediction using a digital twin" [11]. The digital twin still has been reserved for aircraft and has been defined as "ultrarealistic in geometric detail, including manufacturing anomalies, and in material detail, including the statistical microstructure level, specific to this aircraft tail number". In the same year, also Michale Grieve's book, "Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management" [12] expanded the Mirroring Model idea referring to digital twins. A year later, in 2012, digital twins still have been researched as an aircraft concept [13,14].

In 2013, Jay Lee et al. [15] defined the digital twin as “the coupled model of the real machine that operates in the cloud platform and simulates the health condition with an integrated knowledge from both data-driven analytical algorithms as well as other available physical knowledge.” Referring to the “machine”, instead of the “aircraft” or “flying vehicle”, the definition introduced digital twins as a generic concept, able to be used for elements of various types.

In 2014, the article “On the Effects of Modeling As-Manufactured Geometry: Toward Digital Twin” [16] referred to digital twins in a practical use case to predict crack paths. Undoubtedly, the project does not fulfill all the requirements of the fully-performing digital twin (as it will be enclosed in subsequent sections, it is still a case in today's research). However, the practical use enhanced the previous research focused solely on conceptualizing the digital twins' principles. In 2015, Grieves published the first digital twins’ white paper: “Digital Twin: Manufacturing Excellence through Virtual Factory Replication” [17]. The following years brought a boom of digital twinning research exploited by various use cases and for various elements. In 2016, analytics of Gartner placed digital twins as one of the top technology trends [18], predicting that billions of things will be represented by digital twins in the following years.

The history of digital twins’ origins is rather brief – omitting the Apollo XIII physical simulators, it began not much more than two decades ago. Technical innovations of these two decades enabled transforming digital twins from an academic *technical utopia* to a practical concept. Indeed, digital twins left scientific-only domains with implementations of industrial tycoons and are currently perceived as a crucial factor in digital transformation. Nonetheless, digital twinning is still a dynamically evolving idea, with new use cases being explored and principles being established to match the requirements of the modern world.

2.2.2 Definitions

The digital twinning concept is still being formed, which is reflected in the multitude of definitions. Table 2-1 lists selected definitions in chronological order, which enables analysis of the concept’s evolution over time.

Table 2-1. Digital twins definitions

Publication	The digital twin paradigm for future NASA and US Air force vehicles [13]
Publication year	2012
A Digital Twin is an integrated multiphysics, multiscale, probabilistic **simulation** of an **as-built** vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin. […] A digital twin consists of three parts: **physical product, virtual product, and connected data that tie the physical and virtual product.**	

Publication	The airframe digital twin: Some challenges to realization [14]
Publication year	2012
The Airframe Digital Twin is a tail number specific computational model of an individual aircraft. It has the potential to improve the way US Air Force aircraft are managed over their entire lifecycle by creating individualized structural management plans. […]	

An ADT is a **cradle-to-grave model** of an aircraft structure's ability to meet mission requirements. It is a submodel of an all encompassing Aircraft Digital Twin which would include **submodels** of the electronics, the flight controls, the propulsion system, and other subsystems. The ADT, as an **ultra-realistic model of the as-built and maintained airframe**, is explicitly tied to the materials and manufacturing specifications, controls, and process used to build and maintain the aircraft. It is a consistent model of an individual airframe by tail number that includes all variation and uncertainty in that aircraft.

Publication	Challenges with structural life forecasting using realistic mission profiles [19]
Publication year	2012
The Airframe Digital Twin (ADT) is envisioned to be an **ultra-realistic, cradle-to-grave computer model of an aircraft structure** that is used to assess the aircraft's ability to meet mission requirements. The ADT will virtually fly each flight that the physical aircraft flies in order to determine loading and subsequent damage.	

Publication	Multiphysics stimulated simulation digital twin methods for fleet management [20]
Publication year	2013
The Digital Twin concept is to integrate **ultra-high fidelity simulation with an on-board health management system, maintenance history, and historical vehicle and fleet data** to mirror the life of a specific flying physical twin (or "tail number") to enable significant gains in safety and reliability. The intent is to move away from "factors-of-safety" heuristics and similitude rules towards **a complete, fundamental understanding of physical processes** related to degradation of the materials, structures, and systems that control the vehicle life cycle.	

Publication	Recent advances and trends in predictive manufacturing systems in big data environment [15]
Publication year	2013
The coupled model is a digital twin of the real machine that operates in the **cloud platform** and **simulates** the health condition with an integrated knowledge from both **data driven analytical algorithms** as well as **other available physical knowledge**. […] The simulation model can then be considered a **mirror image of the real machine**, which is able to **continuously record and track** machine condition during the later utilization stage.	

Publication	Coupling Damage-Sensing Particles to the Digitial Twin Concept [21]
Publication year	2014
Digital twin is a **life management** and certification paradigm whereby models and simulations consist of **as-built vehicle state, as-experienced loads and environments**, and other vehicle-specific history to enable **high-fidelity** modeling of individual aerospace vehicles throughout their service lives.	

Publication	About the importance of autonomy and digital twins for the future of manufacturing c
Publication year	2015
Very realistic models of **the current state** of the process and their **own behavior in interaction** with their environment in the real world	

Publication	Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems [6]
Publication year	2016
Digital Twin is a set of **virtual information constructs** that fully describes a **potential or actual physical manufactured product from the micro atomic level to the macro geometrical level**. At its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin. Digital Twins are of two types: Digital Twin Prototype (DTP) and Digital Twin Instance (DTI). DT's are operated on in a Digital Twin Environment (DTE).	

Publication	The US air force digital thread/digital Twin – life cycle integration and use of computational and experimental knowledge [22]
Publication year	2016
Digital Twin - an integrated multi-physics, multi-scale, probabilistic **simulation** of an **as-built system**, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin. The DT/DTw is an integrated use of **knowledge over the entire life cycle** of a weapon system providing life cycle information connectivity and continuity.	

Publication	Digital Twin - The Simulation Aspect [5]
Publication year	2016
Digital Twin refers to a **comprehensive physical and functional description** of a component, product or system, which includes **more or less all information**, which could be useful in **later lifecycle phases**.	

Publication	From Simulation to Experimentable Digital Twins [23]
Publication year	2016
Digital twin is a **virtual representation** of a real world subject (person, software system, …) or a real world object (machine, component, part of the environment, …). A digital twin contains models of its **data** (geometry, structure, …), its **functionality** (data processing, behavior, …) and its **communication interfaces**.	
Publication	A simulation-based architecture for smart cyber-physical systems [24]
Publication year	2016
In order to accurately predict **future states** of a smart cyber-physical system, which can change its behavior to a large degree in response to environmental influences, the existence of precise models of the system and its surroundings is demandable. In machine engineering, **ultra-high fidelity simulations** have been developed to better understand both constraints in system design and possible consequences of external influences during the system's operation. These digital twins enable further applications in software design for complex cyberphysical systems as online planning methods can utilize good simulations to continuously optimize the system behavior, yielding a software architecture framework based on the information flow between the **cyber-physical system, its physical environment and the digital twin model.**	

Publication	Industrial IoT lifecycle via digital twins [25]
Publication year	2016
Digital Twins are a new mechanism to manage **IoT devices and IoT systems-of-systems** throughout their **life-cycle**. […] Currently, the Digital Twin technology is moving towards creating a digital representation of a real world object with **focus on the object itself**.	

Publication	Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange [26]
Publication year	2016
The Digital twin being a **virtual representation of the real product**. It has product's **information since the beginning of the life until the disposal of the product**. The Digital Twin can be seen as a cyber representation in the context of the Cyber Physical Systems. The Digital Twin model **can be composed by different kind of models and data**.	

Publication	A Review of the Roles of Digital Twin in CPS-based Production Systems [27]
Publication year	2017
The Digital Twin (DT) is meant as the **virtual and computerized counterpart of a physical system** that can be used to simulate it for various purposes, exploiting a **real-time synchronization** of the sensed data coming from the field; such a synchronization is possible thanks to the enabling technologies of **Industry 4.0** and, as such, the DT is deeply linked with it.	

Publication	Deloitte Industry 4.0 and the digital twin [28]
Publication year	2017
A near-real-time digital image of a physical object or process that helps optimize **business performance**.	

Publication	Digital Twin" in "CIRP Encyclopedia of Production Engineering [29]
Publication year	2019
A digital twin is a digital representation of an **active unique product** (real device, object, machine, service, or intangible asset) or **unique product-service system** (a system consisting of a product and a related service) that comprises its selected characteristics, properties, conditions, and behaviors by means of models, information, and data within a **single or even across multiple life cycle phases**.	

Publication	On the Integration of Agents and Digital Twins in Healthcare [30]
Publication year	2020
A digital twin is a digital representation of a physical asset reproducing its data model, its **behaviour and its communication** with other physical assets.	

Publication	Review of digital twin about concepts, technologies, and industrial applications [31]
Publication year	2021
A digital twin is a digital entity that **reflects physical entity's behavior rule** and keeps **updating through the whole life cycle.**	

Publication	Digital Twin Technology Challenges and Applications: A Comprehensive Review [32]
Publication year	2022
A digital twin is a virtual representation of a **physical object or process** capable of **collecting information** from the real environment to represent, **validate and simulate** the physical twin's **present and future behavior**.	

A digital twin is still not a uniform concept. Various definitions focus on various aspects. From the very beginning, the digital twin has been envisioned as a simulation reflecting the actual conditions of the physical counterpart in its whole lifecycle. The digital twin also early has been spotted as a data-management system; the data could lead to intelligent algorithms [15] facilitating the understanding of physical processes [20]. Definitions highlight digital twins' characteristics, like interaction with the environment [20], modularity [26], or compatibility with IoT devices [25]. The industry sees digital twins as optimizers of business performance [28].

The chronology of the definitions reveals a change of attitude. The early definitions envisioned "ultra-high fidelity" [20] simulations at the "micro atomic level" [6]. These definitions were formulated when digital twinning was limited by technical maturity; practical implementations only aspirated to fulfill the near-perfect requirements in the future. Most current definitions, however, are less strict and focus on practical benefits resulting from digital twinning. Due to technical advances, digital twins transformed from a *technical utopia* to a practical industrial tool.

2.3 Digital twins overview

Digital twins are still being formed, so the research is dualistic: both theoretical frameworks and definitions are being established, and practical use cases are being explored. This section describes activities regarding digital twinning. The activities are divided into domains: general, civil engineering, and bridges – with a particular focus on the latter.

Since digital twinning comprises many technologies, not only research referring explicitly to "digital twins" is indeed digital twinning research; many publications about simulations, modeling, IoT devices, sensors, or intelligent algorithms can undoubtedly be classified in the digital twinning category. The categorization is even more complicated with mature BIM applications that share a plethora of features with digital twins – some articles about BIM are actually about DT [33], even though they do not refer to it. Nonetheless, the articles referenced in the upcoming subsections refer explicitly to "digital twins". This approach enables to disclose how the digital twins are perceived by implementors.

2.3.1 General

Tao et al. [34] divide digital twinning development into three stages: formation, incubation, and growth. The *development magnitude* of the phases is well reflected in the research interest. Fig. 2-1 presents the scientific journals' articles and conferences' papers with the term "digital twin" (or "digital twins" or "digital twinning") in the title in the Scopus database. The stages of formation and incubation (until 2015) are represented by the relatively low number of publications compared to the later growth stage of dynamic interest increase.

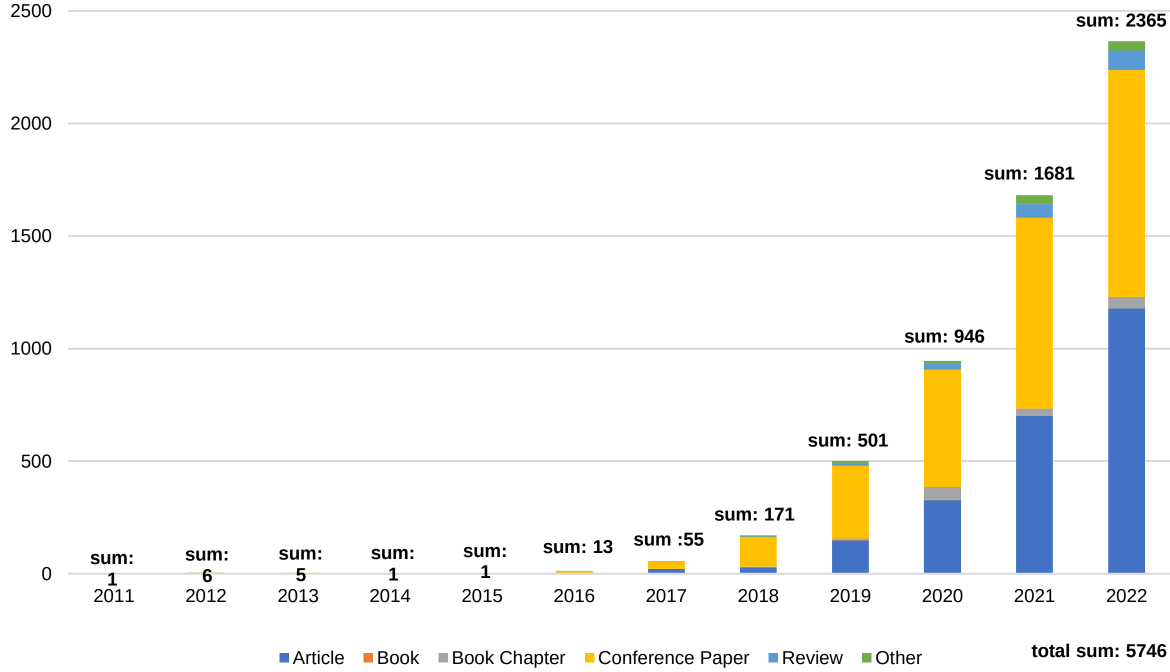


Fig. 2-1. Digital twinning research interest (Scopus database; access: 15.01.2023)

The growth stage brought not only research interest but also its diversification. Fig. 2-2 shows the areas of digital twinning research. Engineering and computer science – the origin domains – dominate, but digital twins are applied in a variety of sectors. Applications in the business and decision sciences, as well as economics, show the DT business potential. Applications in social sciences, biochemistry, medicine, pharmacology, or neuroscience show that digital twinning regard not only technical objects but even humans. Niche attempts in psychology, dentistry, or veterinary show the widening of digital twinning usage.

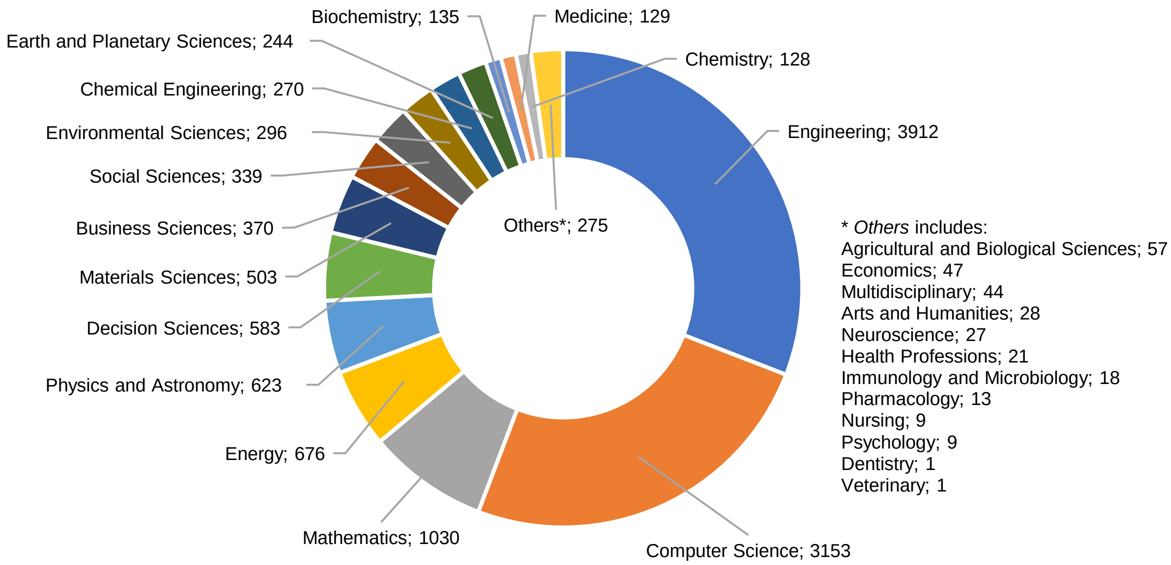
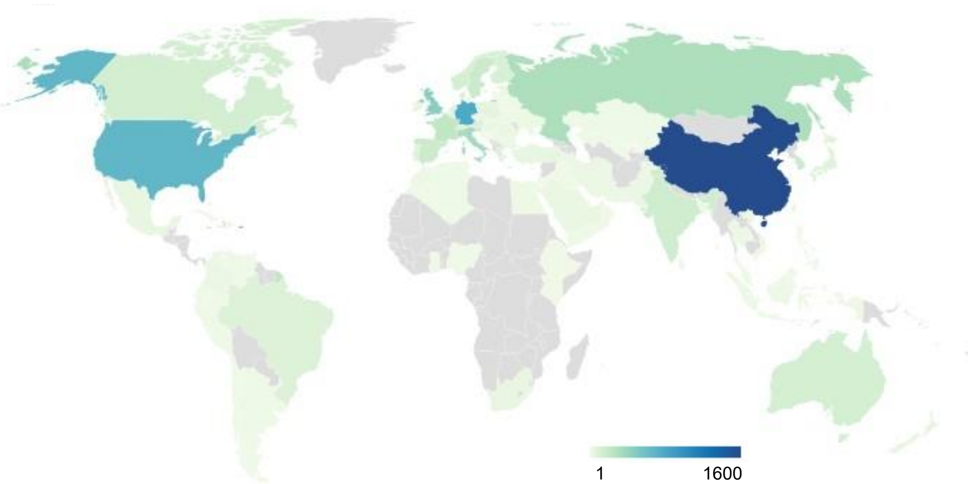


Fig. 2-2. Digital twinning publications areas (Scopus database; access: 15.01.2023)

Fig. 2-3 shows a map of publications. China is leading in publication count (which is a general trend, not specific to digital twinning). Germany, a country linked with the Industry 4.0 idea, is second, and the United States is third. Subsequent places are split between countries of various continents. Overall, digital twinning can be perceived as a global concept.

Country	*Publications*
China	1594
Germany	786
United States	671
United Kingdom	463
Italy	342
Russian Federation	303
France	216
Sweden	166
Spain	163
South Korea	158
India	147
Canada	136
Australia	127
Norway	122
Finland	111
Netherlands	104
Denmark	102
Singapore	102



** countries with more than 100 publications included*

Fig. 2-3. Digital twinning publications map (Scopus database; access: 15.01.2023)

Digital twinning research is diversified also in its nature. Dedicated frameworks and use cases are one of the areas. Digital twins have been applied in various engineering domains, e.g., cyber-physical systems [21,32], smart manufacturing [35–37], and product design [38], as well as for various subjects, e.g., hollow glass production line [39], space satellites [40], wind turbines [41], and printing machines [42]. The

Experimentable Digital Twins [43] are an example of an inheriting concept. Extensive research is also held into orchestrating digital twinning operations, e.g., scaling digital twinning systems [44], developing them in clouds [45], and ensuring their cybersecurity [46].

Digital twins are perceived as key components of global industrial strategies like Industry 4.0 [47–49] or Made in China 2025 [37]. This is enabled by the linkage of digital twins with technological techniques: big data [48], Internet of Things [45] and Internet of Services [50] (and envisioned Internet of Everything [48]), and blockchain [51,52]. Digital twins emerge from and are interconnected with model-based concepts like intelligent products [53], smart products [54], product avatars [55,56], virtual factories [27,57], and cyber twins [38]. The capabilities of digital twins make them a proper base for introducing engineering methodologies: data-driven design [48], event-driven engineering [58], simulation-based engineering [59], personalized medicine [60,61], communication by simulation [5], and Software as a Service (SaaS; with its dedicated Digital twin as a Service [62] approach).

Digital twinning is not restricted to engineering. Human-related digital twins are an emerging research area. Digital twins are applied in medicine for, e.g., drug discovery (the Virtual Liver concept [63]), precision nutrition [61], dentistry [64], and psychology [65]. Clothes design [66] is another human-related DT application. Digital twinning is also utilized for animals [67]. The premise of digital twinning capabilities triggers futuristic visions: [68] envisions a human digital twin as a technological approach to immortality (in a Steven Spielberg's Avatar fashion); [69] envisions a digital twin of Earth.

The digital twinning capabilities and its current and potential use cases place digital twinning as a sociological phenomenon. It is reflected in research on digital twinning impact [70] and its ethical implications [60]. Maturing of digital twinning research is reflected in the multitude of review articles [31,32,34,71–75] describing definitions, frameworks, use cases, technologies, challenges, and business perspectives. Digital twinning is also announced in *apostolic* articles, like "Make more digital twins" in Nature [76].

The digital twinning research brought many concepts and general principles. [13] described three parts of the digital twinning system: physical product, virtual product, and connected data linking the physical and virtual products. The parts are linked by interactions: physical-physical, virtual-virtual, and virtual-physical [34]. [41] introduced five-dimensional architecture consisting of a physical entity, virtual model, physical-virtual connection, data, and services [41]. [6] divided digital twin types into digital twin prototypes (the prototypical framework) and digital twin instances (the virtual model describing the physical counterpart) existing in the digital twin environment. [22] introduced digital thread, the aggregation of processes leading to creating and utilizing digital twins. [71] defined digital twinning levels of integration regarding manual and automatic data flow between the physical and virtual counterparts: digital model (manual data flow), digital shadow (automated data flow from physical to virtual and manual from virtual to physical), and digital twin (bi-directional automated data flow). The capabilities of these concepts resulted in the conclusion that digital twins are the next wave in simulations [5].

Digital twins’ are also present in the industry. Deloitte [49], Siemens [77,78], and Gartner [79] published their reports on digital twinning. Also, many industrial tycoons declare their implementations. Tesla, with its autonomous cars, is a prominent representative. Besides cars, Tesla uses digital twinning for smaller-scale devices, like batteries, and whole production lines to identify bottlenecks and optimize the flow of parts. Mercedes (Daimler) is another automotive industry representative. Boening, Airbus, and Rolls-Royce use digital twinning in aerospace engineering to design aircraft and optimize engines. GE Digital (General Electric subsidiary) developed the “Aviation Digital Twin”. Siemens creates digital twins of power plants, wind farms, and buildings; Shell – of offshore platforms and refineries. Microsoft developed the “Azure Digital Twins” platform, intended to be a multi-industrial simulation environment. These examples show that digital twins have already transformed from scientific vision into industrial practice.

2.3.2 Civil engineering

Digital twinning, as a global concept, also reached civil engineering resulting in both scientific and industrial interest. Similar to the general trend, civil engineering digital twinning research is divided into applications and establishing principles. The research often sources from current AEC techniques, like Building Information Modeling (BIM) or structural health monitoring (SHM), enhancing it with, e.g., artificial intelligence. Especially the view on BIM is controversial – as a methodology based, in practice, on data-rich models, BIM is sometimes perceived as digital twinning. More thorough analyses, however, highlight their differences.

Civil engineering digital twinning principles are already being structured into initial frameworks. The frameworks – though not unified in all concepts – form a great foundation for civil engineering digital twinning.

[80] introduces the Construction Digital Twin (CDT). The framework attempts to address the limitations of BIM (incompatibility with IoT and dynamic data; lacks in automation and interoperability) with the addition of semantic web linked data, big data, artificial intelligence, and services. Construction Digital Twin is based on the physical part (sense, monitor, actuate), the data (BIM, IoT, data linking, knowledge storing), and the virtual part (simulate, predict, optimize, agency). Construction Digital Twin framework also suggests the evolutionary approach of three generations: 1) monitoring platforms (as-built BIM; features: sensing, analyzing, monitoring), 2) intelligent semantic platforms (web-based platform with limited intelligence linking digital twin with IoT devices; features: sensing, analyzing, monitoring, AI), and 3) agent-driven socio-technical platforms (fully autonomous systems; features: sensing, simulating, AI, optimizing, learning, end-user engagement).

[81] introduces the Digital twin construction (DTC). The framework is based on BIM, lean construction, automated data acquisition, and artificial intelligence. Although BIM plays a role in the framework, the article highlights the naivety of the synonymization of BIM models and digital twins: digital twins for construction are connected to physical counterparts, not only replicate them; they also operate in a network with other digital twins. The framework highlights the importance of modeling both product characteristics and processes. It also envisions the lifecycle of digital building twins and their physical counterparts,

differentiating Foetal Digital Twin, Child Digital Twin, Adult Digital Twin, as well as Child Physical Twin and Adult Physical Twin.

Civil engineering digital twinning concepts are a domain of not only academics. National Digital Twin (NDT) is a Centre for Digital Built Britain framework [82] envisioned as an ecosystem of digital twins collaborating in data exchange. The National Digital Twin is a next-step framework focused not on creating singular digital twin instances but rather on their linkage. The UK government expects the National Digital Twin to contribute to the national economy, public good, and environment.

Civil engineering digital twinning research provides also use cases. It is important to notice, however, that in the current stage of digital twinning maturity, the applications are not complete digital twins but rather focus on particular components: BIM, SHM, AI, point clouds, virtual, augmented, and mixed reality, automation, optimization, automated modeling, etc., sometimes mixing the components. Although partial, these applications are pushing the idea of civil engineering digital twinning. [83] describes a limited (as claimed by authors) digital twin of a building façade with sensor networks enabling an analysis of the actual light, temperature, and humidity conditions. [84] introduces an IFC-based digital twin for detecting anomalies in HVAC systems. [52] is an attempt to accountable information sharing with digital twins and blockchain, tested on prefabricated brick positioning. [85] presents a semi-automatic approach (referred to as geometric digital twinning) to generate IFC models from images and CAD drawings. [86] describes digital twins for bulk silos to manage materials' supply chains. [87] uses digital twinning for replicating hospitals (HospiT'Win framework) to monitor, analyze, and predict patients' pathways. The exemplar use cases show the variety of facility types (or their parts) and objectives on different lifecycle stages. Increasing research resulted in reviews. [88] reviews 22 publications and depicts six main research areas: BIM, structural system integrity, facilities management, monitoring, logistics processes, and energy simulation. [89] reviews 100 publications focusing on BIM features facilitating the evolution into digital twins; the study diversified research into the construction process, building energy performance, and indoor environment monitoring. [33] reviews 134 publications (diversified into the design, construction, and maintenance stages applications) and highlights differences between digital twins, BIM models, and cyber-physical systems. [90] is a review focused on digital twinning for construction workforce safety (e.g., workforce behavior monitoring, identification of risks, safety planning, and training).

Smart cities is another prominent research area regarding both methodologies and city-specific use cases. [91] uses BIM for visualizing, estimating, and finally promoting the Net Zero Energy Buildings concept for existing facilities. [92] focuses on smart cities' disaster management. [93] processed LiDAR point clouds (with points' clustering and object recognition) for creating Digital Twin Cities. [94] describes Digital Twin city of Atalanta, a virtual reality platform with a 3D model. [95] developed a digital twin of the West Cambridge site of the University of Cambridge as a unified data source for effective management on the city level. [96] describes the digital twin of Zurich to address the pressing requirements of the growing city. [97] introduced a digital twin of the Docklands area in Dublin, freely accessible to citizens to acquire feedback on planned changes or existing problems. [98] describes the Helsinki 3D city model for energy-related data acquisition and analysis; the study highlights the importance of the CityGML data model for the smart cities concept. [99] conceptualized a smart city architecture with machine learning and semantic modeling and tested it for Chicago buildings.

The business potential of digital twins resulted in interest outside the academic environment. The government of the United Kingdom founded the Centre for Digital Built Britain and initiated the Industrial Strategy Transforming Construction Programme with the National Digital Twin as its component [82]. The United Kingdom perceives digital twins as beneficial for the internal economy and export business, opening new markets and business models. Digital twinning in infrastructure is a subject of white papers of industrial organizations (e.g., Digital Twin Consortium [100]) and business companies. Deloitte published "New Technologies Case Study: Data Sharing in Infrastructure" [101], stating predictions and business opportunities. Siemens published "Digital twin – Driving business value throughout the building life cycle" [77]. This dedicated interest of technological, non-civil engineering companies reveals a new potential trend in the construction industry – an industry of enormous economic and social impact, which still has a vast margin of increasing its efficiency (resulting in accountable benefits for the optimizers).

2.3.3 Bridges

Bridges are a relatively new area of digital twinning research. Fig. 2-4 presents the number of scientific publications referring to digital twins and bridges in the title in the Scopus database (code: TITLE (("digital twin" OR "digital twins" OR "digital twinning") AND (bridge OR bridges)) and further manual exclusion of irrelevant articles; publications' data listed in Table A-1). The publications started to appear in 2019. Counts in the following years convince that bridge-oriented digital twinning research is still nascent but growing.

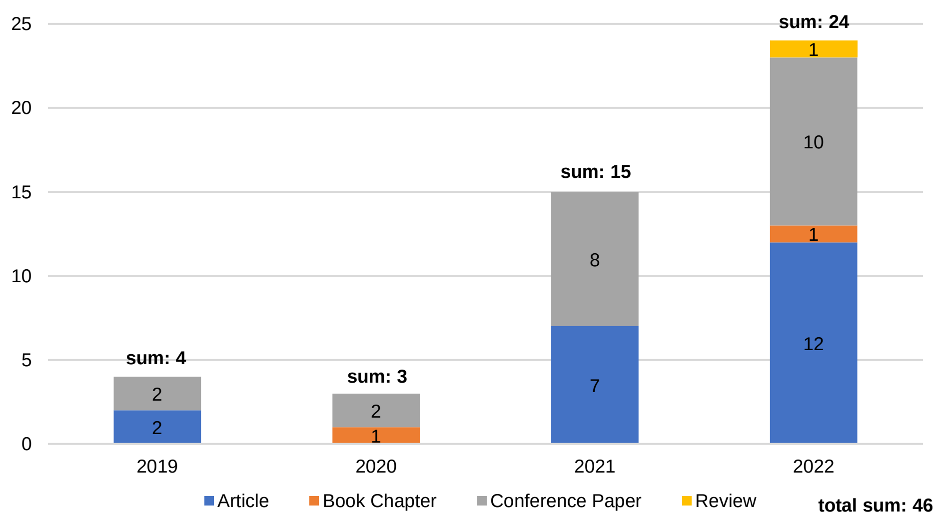


Fig. 2-4. Digital twinning bridge-oriented research interest (Scopus database; access: 15.01.2023)

The bridge-oriented digital twinning research is diversified worldwide (Fig. 2-5), following the general trend with leading China, Germany, and United States.

Country	Publications
China	10
Germany	8
United States	6
South Korea	5
United Kingdom	4
Australia	2
Norway	2
Sweden	2
Czechia	1
Dominican Republic	1
Indonesia	1
Iran	1
Morocco	1
Netherlands	1
Spain	1

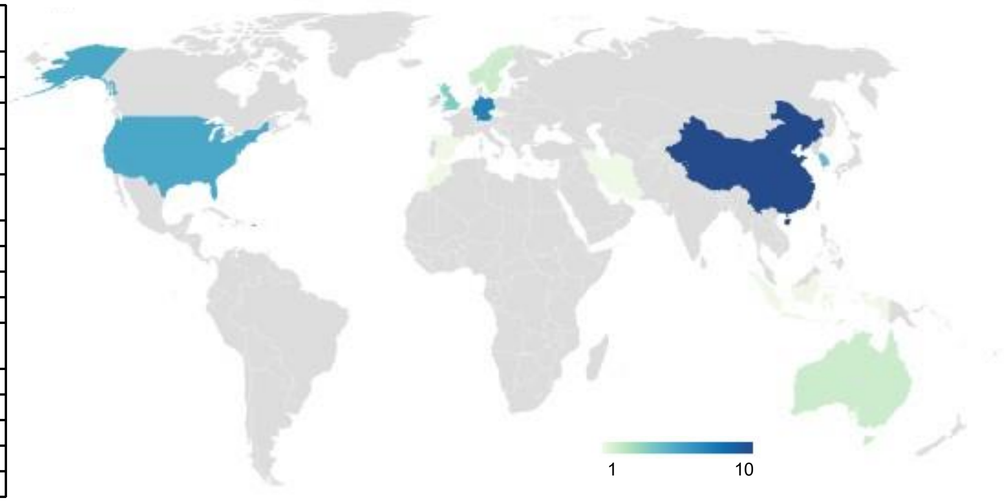


Fig. 2-5. Digital twinning bridge-oriented publications map (Scopus database; access: 15.01.2023)

Bridge-oriented digital twinning research comprises a variety of use cases. The use cases, similar to civil engineering solutions presented in the previous section, are not yet full digital twins but rather implementations focused on particular components: BIM, SHM, AI, point clouds, virtual, augmented, and mixed reality, automation, optimization, automated modeling, etc., sometimes mixing the components.

[102] discusses "digital twin models" for cable-supported bridges. The article explains that the definition of a digital twin model for a bridge should be different depending on its intended use and that it requires knowledge and collaboration from all participants in the bridge engineering process. The article describes a pilot application of a digital twin model for a cable-stayed bridge in Korea; an initial model was developed for maintenance and field inspection. The article also mentions challenges such as limitations of current technology, data security, and regulations.

[103] aims to provide clear and actionable steps for creating an operational digital twin that meets the demands of infrastructure managers. The five steps in the method are: 1) Data and Need Acquisition, 2) Digital Modelling, 3) Sensor System Design, 4) Data/Model Integration, and 5) Operation. Clifton Suspension Bridge case study in Bristol, UK, illustrates the method. The authors aim to justify the investment in digital twinning by ensuring that each design decision is made to fulfill an operational need identified during the requirements capture process.

[104] proposes a framework for bridge management systems using digital twinning. Digital models add automation, efficiency, and accuracy to the system. The study divides the digital twin approach into five steps: bridge inspection, BIM modeling, damage identification, data transmission, and facility management. The framework aims to ensure the safety of bridges' operation and minimize investments in maintenance.

[105] discusses the use of fiber optic sensor networks and BIM to monitor the behavior of two newly constructed railway bridges in Staffordshire, UK. Intersection Bridge 5 was instrumented during construction with 291 Fibre Bragg Grating sensors, and Underbridge 11 was instrumented with 220 Fibre Bragg Grating sensors and 260 m of Brillouin Optical Time Domain Reflectometry cables. The primary

objective of the study was to perform early-age behavior assessments to inform long-term condition monitoring. The research uses BIM to visualize real-time sensor data and associated bridge behavior, FEM to predict the performance of the bridge, and statistical modeling to detect anomalies in the data.

[106] describes a multimedia knowledge-based bridge health monitoring system. The study suggests that by simulating various situations, it will be possible to provide digital services to ensure bridge health, analyze situations based on a small amount of data, predict the optimal maintenance techniques, and apply them in practice.

[107] presents a digital twin architecture for the six-step cycle of monitoring and maintenance of road and bridge construction in West Java, Indonesia. The architecture consists of a survey process, technical planning, budgeting, procurement, implementation, and service. The authors also discuss challenges in implementing digital twin technology for road and bridge management, including map availability, license ownership, and security systems.

[108] presents an automated pipeline for processing sensor-based data produced during load tests on high-speed railway bridges. The pipeline is developed within the H2020 European project ASHVIN, which focuses on Assistants for Healthy, Safe, and Productive Virtual Construction, Design, Operation & Maintenance using Digital Twins. The pipeline uses event-driven microservices, which integrate the ASHVIN IoT platform, the IFC model, and services automating processes. The pipeline is demonstrated using a load test of a high-speed railway bridge in Spain. The study aims to automate the process of load testing, which would reduce labor times and costs and increase productivity.

[109] proposes a structure health hybrid monitoring method for cable-stayed bridges, focusing on the orthotropic steel deck. The method synthesizes monitoring data with a FEM model to reconstruct the un-monitored structure responses. The study uses submodels to examine the distribution of welding stress and its effect on coupling with vehicle- and temperature-induced stress. The proposed method evaluates vulnerable parts of bridges and then updates the digital twin based on the evaluation results.

[110] uses digital twinning to assess the safety, durability, and reliability of bridges. The study uses a chemo-mechanical model to simulate the initiation and propagation of chlorides or carbonation in concrete, which is combined with nonlinear modeling of cracking, bond failure, and reinforcement yielding. The model is implemented in ATENA software and validated using experimental data. The study demonstrates the concept on a bridge in Germany.

[111] proposes digital twinning for managing the fatigue of steel bridges. The proposed digital twin architecture is supported by eight functional modules: artificial intelligence, analysis, data link, data storage, coupling, computation, simulation, and user interface. The implementation of digital twinning is envisioned over different phases during the bridge's lifecycle, including the design, construction, and operation. The main identified challenge is the lack of understanding of steel bridge fatigue and the insufficiency of the present technologies.

[112] proposes a framework for non-deterministic fatigue life prediction of steel bridges. A probabilistic multiscale model is developed to depict the fatigue evolution throughout the bridge lifecycle. The small

crack initiation period is modeled considering microstructure uncertainties. The model is calibrated using historical fatigue data.

[113] uses BIM and a risk inspection model to assess the vulnerability and plan maintenance for the Zhongcheng Village Bridge in Zhejiang Province, China. The study used Revit software to construct a 3D model based on the CAD drawings. The study also used a concrete multi-function detector to assess concrete diseases and analyzed the scanned data to identify internal risks and decide on treatment methods.

[114] proposes a framework for fatigue lifecycle management of steel bridges. The framework includes three main components: fatigue prediction, maintenance, and inspection/monitoring. The framework uses a probabilistic multi-scale fatigue deterioration model for predictions. Bayesian inference of the deterioration parameters allows for real-time updating of the predicted fatigue evolution process, providing a basis for lifecycle optimization. The framework aims to establish maintenance, inspection, and monitoring strategies for deteriorating steel bridges.

[115] proposes a method for conducting seismic fragility analysis on deteriorated bridges. The method consists of two phases: 1) using UAVs to inspect the bridge and update the digital twin with information on the extent of deterioration, and 2) using the updated digital twin to conduct seismic fragility analysis. The study demonstrates the method with a deteriorated concrete bridge case study. The study states that the method allows for accurate seismic fragility analysis of deteriorated bridges.

[116] presents a collapse fragility assessment for long-span cable-stayed bridges under strong earthquakes. The method utilizes a scaled long-span cable-stayed bridge and its shake table tests. Also, three FEM models were created: a design documentation-based model, a linearly updated model, and a nonlinearly updated model. The assessment results are compared regarding collapse mechanisms and ground motion intensities.

[117] proposes a digital twin-based life-cycle seismic performance assessment for long-span cable-stayed bridges. The method aims to evaluate the structural performance of bridges over their lifecycle, during which they may experience multiple earthquakes and accumulate seismic damage. The proposed method includes three components: 1) a seismic hazard analysis-based generation of earthquake occurrence sequence, 2) a digital twin-based structural response prediction, and 3) a service life quantification.

[118] presents a method for creating a digital twin of a bridge using output-only Bayesian model updating and recorded seismic measurements. The study’s motivation is to facilitate rapid post-earthquake damage diagnosis of bridges to guide decision-making for emergency management. The method comprises linear and nonlinear FEM models. The method is examined with a numerical verification and a real-world case study using seismic data recorded from the San Rogue Canyon Bridge in California.

[119] presents the development of seismic fragility curves for a precast reinforced concrete bridge instrumented with an SHM system. The bridge is located in a seismic activity area in the Dominican Republic and links several local communities. It was designed with outdated techniques not adequate for

structures in seismic regions. To obtain the fragility curves, a FEM model and the SHM data were combined. The proposed methodology supports disaster mitigation and post-disaster decision-making strategies.

[120] introduces a concept of a Lifecycle Digital Twin (LDT) for bridges, which aims to facilitate the use of BIM throughout the entire lifecycle of an asset. The proposed model is based on ontological modeling resulting in converting the conventional BIM models to LDT. The study used a Dutch contractor's Revit models as case studies. The authors interviewed domain experts from different disciplines to identify the information that should be included in LDT. The proposed ontology was then validated through a workshop session where domain experts assessed the results.

[121] describes a method for automating the creation of parametric BIM models for bridges using the particle swarm optimization algorithm. The method introduces the modeling of bridges from point clouds with parametric profiles. The profiles are created based on pre-knowledge about the existing elements in a typical bridge. The algorithms are applied to extract the parameters' values for the profiles.

[122] describes the automated generation of 3D models of existing reinforced concrete bridges from labeled point clouds. The method uses a slicing-based object fitting method to address the challenges of irregular geometries of existing bridges. The quality of the generated models was evaluated using cloud-to-cloud distance-based metrics. Experiments on ten bridge point clouds resulted in an average modeling distance of 7.05 cm and an average modeling time of 37.8 s, which is a significant improvement over manual modeling. The method's output is an IFC model in LOD 250-300, considered as a base for a digital twin. The method consists of two main steps: extracting geometric features and detecting shapes in point clusters, and fitting the extracted features and identified shapes into IFC objects.

[123] regards creating a model of Africa's longest cable-stayed bridge (with a span of 950 m): Mohammed VI Bridge in Morocco. The technique involves the Leica Pegasus Two Unit, which combines LiDAR and high-resolution panoramic images to create a 3D model. The data collected by the Pegasus Two Unit is processed using specialized software, which converts it into numerical formats and creates a virtual display of the bridge. The model is then used for analysis and inspection.

[124] describes photogrammetry and laser scanning as techniques to digitize bridges providing accurate data for indirect inspection and assessment. The authors investigate and evaluate the geometric accuracy of two 3D models generated from an existing heritage bridge in Australia, one using unmanned aerial vehicles photogrammetry and the other using terrestrial laser scanning. The results show that both UAV photogrammetry and TLS-based point clouds are proper for bridge inspection, but TLS has a higher points' density with suitable accuracy. The article also describes a case study of a timber truss McKanes Fall Bridge in New South Wales, Australia, to upgrade and rehabilitate the bridge to a safer, more reliable structure.

[125] presents an approach for creating digital twins of existing bridges with AI-based methods for point clouds. The article proposes a point clouds' segmentation method that is the basis for the parametric approach to semi-automatic geometric modeling. The article proposes an overall processing workflow for the digital twinning of bridges using AI-based methods.

[126] proposes a digital twin system for a group of bridges around Shanghai, China. The system measures traffic loads on the target bridge and employs lightweight sensors for structural response information on other bridges in the group. Using analysis models and measuring traffic loads, the system analyses the working condition of the bridges and constitutes a collective safety warning technique. The proposed method aims to be a technical basis for future intelligent transportation infrastructure systems.

[127] reviews the use of laser scanning in the context of bridge digital twinning. The study was conducted using a literature search of Scopus, Google Scholar, and Web of Science, covering the period of 2017-2022. The findings suggest that while the use of digital twinning is increasing, there are still challenges to overcome, such as standardization, before the construction sector can fully benefit from it.

[128] is a literature review on digital twins in the construction industry, specifically focusing on bridges (16 bridge-oriented articles were reviewed). The review found a significant difference between the amount of research on bridge inspection and BIM compared to digital twins. The main identified digital twinning challenges relate to managing large amounts of data gathered from different platforms. The review concludes that many papers that include the term "digital twins" in their keywords or text did not actually discuss the creation of DT models but rather used the term as a synonym for BIM models or stated that the research could lead to a digital twin in the future.

The presented use cases show diversification of the bridge digital twinning research areas (which is revealed in the cloud word composed of the articles' keywords in Fig. 2-6). Digital twinning is being applied to bridges of various functions, structures, and materials. Also, various techniques are utilized for a variety of aims. The positive trend is blending the techniques and goals: SHM sensors are linked with BIM; BIM models with FEM; point clouds with machine learning; monitoring with management. Bridge-oriented digital twinning research also comprises reviews which is a sign of research maturing. On the other hand, the research lacks framework-specific activities stating the principles and concepts suited for bridges.

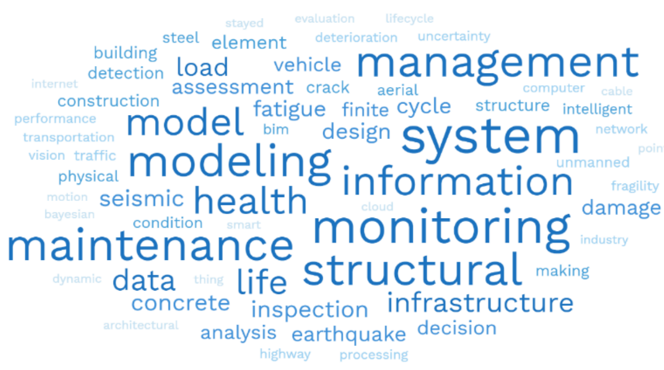


Fig. 2-6. Digital twinning bridge-oriented publications word cloud

2.4 Conclusions

Digital twins, as a relatively new concept, are constantly evolving with forming principles and new use cases. Nonetheless, they are already perceived as the key component of Industry 4.0, being a global paradigm of digitization. The concept of virtual counterparts of objects through the entire lifecycle is gaining momentum with increasing academic and industrial interest. It is enabled by technological advances and motivated by the modern world's demands of holistic data analysis, management, and multi-industrial cooperation.

Digital twins have reached civil engineering and bridges. Civil engineering researchers propose general frameworks and initial applications utilizing current and adapted techniques in an enhanced manner. The applications are not yet comprehensive digital twins and focus rather on particular aspects. Although these partial applications are beneficial, more activities are still needed to push the idea of comprehensive digitization. Bridges, complex and reliable structures, requires suited digital twinning frameworks and novel use cases providing coherent benefits, motivating designers, engineers, and managers to adopt and develop the digital twinning idea. In this manner, bridges will be ready to act as connectors of people and data in future smart cities.

3 The concept of digital twins of bridges

3.1 Introduction

The modern world states new challenges, also for civil infrastructure. Sustainable development goals, economic aims, growing population, and pushing demands for adapting to technical advances require the transformation of current practices. The transformation is not only a challenge – it is also an opportunity. *Globally*, regarding the entire industry, the paradigm of digital twins (DT) is an enabler in increasing effectiveness and connecting civil engineering facilities to multi-domain digital cooperation. *Locally*, regarding practitioners, it provides tools for enhancing day-to-day tasks.

As industries become more interdisciplinary, models of different assets will collaborate more, similar to their physical counterparts. We may imagine digital twins of autonomous cars partnering with digital twins of bridges. Objects will collaborate on different dimensions [34]. The physical-physical is already enabled – a car can interact with a bridge. Digital twinning introduces additional virtual-virtual and virtual-physical cooperation (Fig. 3-1). In the example, a car crossing a bridge sends its data. Sensors measure bridge response (e.g., deflections, vibrations) to the particular car regarding the actual conditions of the bridge and environment (e.g., wind, temperature, precipitations). Collected data refine predicting algorithms, contributing to the decision-making of major and everyday actions. If a vehicle is predicted to affect the bridge, the system can reject its carriage. Multidimensional cooperation strengthens the resilience of the physical structure.

Digital twins also establish networks of relations on different levels. In the parallel relation, partnering systems are independent (like a car and a bridge); in the subordinate-superior relation, one model contains another. A bridge digital twin can contain, e.g., digital twins of measurement devices. But on the other hand, a bridge digital twin can be a part of a superior system, such as a digital twin of a smart city. And management of future cities becomes vital with a prediction of 6.66 billion people (68.4% of the forecasted population) living in urban areas in 2050 [129]. Bridges, as connectors between smart city components, must be prepared to provide connections on physical and digital dimensions.

This chapter introduces the proposed foundations for digital twins of bridges. It starts with the analysis of the specificity of bridges (section 3.1.1) and the demands of civil engineering practitioners (section 3.1.2), which must be regarded for a suited, practical solution. Section 3.1.3 describes the proposed approach to civil engineering digital twinning as the evolution of current techniques. Section 3.2 describes the techniques and places them in the digital twinning concept. Section 3.3 presents notions and guidelines for the digital twinning of bridges. The notions complement characteristics (section 3.4) in defining the digital twinning framework.

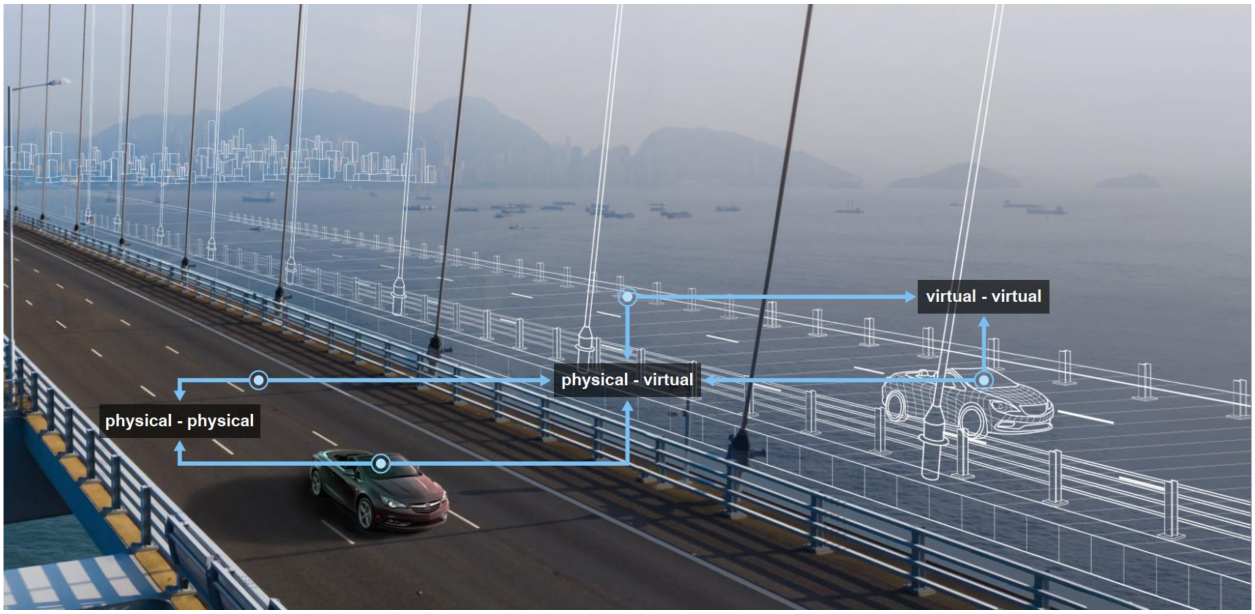


Fig. 3-1. A bridge digital twin vision

3.1.1 Specificity of bridges

Bridges, like every other object, have specific characteristics, requirements, and context of operation that should be regarded in conceptualizing specified frameworks. This section analyzes the features of bridges and the challenges of bridge engineering.

The first factor is the strategic role of bridges. Infrastructure systems provide essential services for the proper functioning of societies [130]. As links between the infrastructure, bridges are indispensable for logistics. Historically, they have also been crucial from a military perspective. Due to these, bridges have always been a strategic component of societies' wealth. This wealth is not easy to assess, but some numbers are helping in the realization of scale. Only in the USA are there 619 thousand of bridges [2]. One of them, the San Francisco Oakland Bay Bridge, appears in the Guinness Book as the most expensive bridge. Just the restoration in 2002 is estimated at 6.3 billion USD.

Safety is – or should be – an indispensable characteristic of bridges. Bridges need to be resilient due to their strategic role and the impact of their malfunctions, which can have a direct influence on people's lives and an indirect influence on economies. The infamous catastrophes of Ponte Morandi in Genua in 2018 (with 43 killed), the Mexico City Metro overpass in 2021 (with 26 killed), and the Julto Pul pedestrian bridge in Morbi, India, in 2022 (with 135 killed) convince that bridges' collapses are not just an imaginary black scenario.

A nontrivial part of the civil infrastructure is aging and suffer from alarming deterioration [130]. It also regards bridges. Of the mentioned USA bridges, 224 thousand require major repair work or replacement [2]. Due to the need for safety, bridges must be adequately monitored and maintained. Nonetheless, bridges' condition assessment still relies mostly on periodic human inspections. The aging bridges require holistic management solutions, especially given their assumed lifespan: many bridges are designed for 100 years.

The modern world aims at new goals; one of them is ecology. The construction and operation of buildings were responsible for 36% of global energy demand and 37% of energy-related CO_2 emissions in 2020 [131]. The ecological factors already influence civil engineering – advanced Life Cycle Assessment (LCA) regarding economic, social, and environmental costs are already being required.

Civil engineering makes up around 15% of the world’s Gross Domestic Product [132]. Although of high impact, civil engineering is ineffective compared to other sectors: since World War II, productivity in manufacturing, retail, and agriculture has grown by around 1500%, but productivity in civil engineering has nearly not changed [133]. Civil engineering ineffectiveness is reflected in chronic misses of deadlines and overspends of budgets [134]. As estimated, civil engineering productivity can be increased by 50-60% [133]. This margin is a business motivation for the implementation of modern solutions.

Bridges are complex and unique. The need for liaison with the environment results in various structural, material, and manufacturing solutions. Even if bridges are of typical structure, their uniqueness comes from the different conditions in which they operate – the same structure influenced by different phenomena throughout its lifecycle can have noticeably different characteristics. The uniqueness and complexity of bridges demand thorough monitoring and analysis.

The analyzed factors regard the bridges and the bridge engineering community. The factors guide in creating – but also convince about the need for creating – the enhanced paradigms of bridge engineering.

3.1.2 Practitioners’ view

Bridges are designed, constructed, managed, and used by people. Addressing the demands and expectations of practitioners is essential for creating a framework providing expected benefits and ready for practical implementation. This section reviews surveys of civil engineering and bridge engineering practitioners.

[135] interviewed 19 expert bridge professionals in the U.K, including managers, operators, and consultants. The main findings of the survey were: 1) safety is a key concern for bridge practitioners; 2) two main types of bridge condition appraisal are damage detection and structural assessment; 3) damage detection is mostly reactive, targeting known issues rather than proactively detecting new damages; 4) very few bridges in the U.K. currently have real-time SHM systems, and these are typically used to investigate known defects or issues; 5) FEM models are typically not kept and maintained by asset owners unless the bridge is a landmark structure of strategic importance; 6) the most highlighted barriers to bridge monitoring are limited budget and the financing model.

[136] investigated the use of data in bridge asset management in the UK by interviewing 17 practitioners. The results revealed a lack of consistency in visual inspection and limited implementation of SHM. Additionally, the study emphasized the need for advanced technologies, such as artificial intelligence, in bridge condition monitoring. Comprehensive and up-to-date data on facilities was also highlighted as crucial for effective risk management and planning.

[137] presents the findings of a survey on unmanned aerial vehicles (UAVs) for bridge inspections and damage quantification. The survey revealed that 56% of respondents have either used or are planning to use UAVs for bridge inspections, however, only 19% of respondents have begun to quantify damage using images captured from UAVs due to technical challenges. Additionally, the study noted that UAV-digital image correlation platforms for bridges are gaining popularity.

[138] surveyed eight experts on BIM-pilot projects. The findings are: 1) replacing a decentralized data system with a Common Data Environment improved collaboration among project participants and provided better task completion; 2) BIM models enabled contractors to proactively manage project progress and construction quality; 3) although beneficial, the BIM model development cost and time might limit the adoption for small and medium projects and stakeholders; 4) all the BIM project maintained or lowered scheduled time and budget; 5) the primary obstacles to BIM implementation are the lack of awareness and knowledge among stakeholders, a shortage of BIM investments, and the incompatibility of traditional project delivery methods.

[139] conducted 20 interviews with academic researchers and industrial practitioners from the West Midlands in the UK. The study found that improved collaboration and automating software make BIM effective for a construction project. The industry-based participants highlighted that the understanding of BIM in academia and industry differs. The respondents identified challenges, such as a lack of understanding of BIM workflow complexity.

[140] analyses data collected from 205 construction projects' stakeholders on critical success factors for BIM implementations in high-rise buildings. The five depicted success components are productivity, visualization, coordination, sustainability, and safety improvement. Visualization was found crucial in enhancing construction safety by identifying potential hazards before construction begins.

[141] examined the use of BIM in China through a survey of 94 BIM professionals. Most respondents believed that BIM's primary advantage is reducing design errors and associated construction rework. Interoperability among different tools was deemed the most crucial factor in realizing the full value of BIM. The owners were perceived as the group benefitting from BIM the most.

The analysis of practitioners' opinions reveals diversified needs and expectations. Emerging civil engineering technologies, like SHM and BIM, were perceived as effective but still challenging in practical implementations. Practitioners also articulated the need for additional advances in analysis (e.g., artificial intelligence) and devices (e.g., unmanned aerial vehicles). Improved collaboration, optimization of costs and time, proactive management, and increased safety and automation were the highlighted benefits of new technologies. Limited investments, lack of awareness, and differences between theoretical and practical views are the identified challenges.

3.1.3 Evolution into civil engineering digital twinning

Establishing the foundations for bridge digital twins must start with a vision. But equally important are the enablers for realization. There is no point in forming idealized concepts that are perfect in theory but inapplicable in practice. To develop, digital twinning must be adopted by civil engineering practitioners, so designers, engineers, and facilities' managers. The digital twinning framework should, therefore, rather guide into practical benefits than restrain with idealized rules.

The first civil engineering applications of digital twins utilize modern civil engineering techniques (e.g., BIM, SHM, 3D reconstruction), often blending them and integrating with, e.g., artificial intelligence. The use cases are still partial. They do not implement complete digital twinning but focus on selected functionalities (e.g., isolated semantic models or acquiring SHM data). Nonetheless, the applications promote the automation and digitization of civil engineering, making a ground for digital twins.

This dissertation proposes principles for creating and utilizing the digital twins of bridges (Fig. 1). The proposed concept addresses the general multi-industrial digital twinning fundamentals (identified with the literature study), the specificity of bridges, and the demands of designers, engineers, and managers of civil engineering facilities. The proposed digital twin is a virtual counterpart of the bridge in its entire lifecycle characterized by actuality, intelligence and autonomy, interaction, interoperability, modularity, expansibility and scalability, accessibility and security, and uniqueness (section 3.4). From the technological perspective, it is an evolution of current practices, which have not yet been properly integrated. The proposed digital twin utilizes BIM (Building Information Modeling), SHM (Structural Health Monitoring), and AI (Artificial Intelligence). It states IFC (Industry Foundation Classes) as the base for the central model. It is also enhanced by other techniques (e.g., visual programming, point clouds). This approach enables the evolutionary process of adopting digital twins in civil engineering. Civil engineering digital twinning does not have to be perfect from the start. It should mature naturally, alluring designers, engineers, and managers of civil engineering facilities with its benefits.

3.1.4 The approach to defining the framework

Digital twins are complex systems. The few-sentence definitions are insufficient to describe them and lead rather to ambiguities than clarifications. Digital twin frameworks, therefore, should be defined thoroughly by used techniques and characteristics.

The following sections describe the components of the proposed framework. Section 3.2: *Techniques of civil engineering digital twinning* describes techniques identified as crucial for creating and utilizing digital twins of civil engineering facilities. Section 3.3: *Notions towards digital twinning of bridges* present guidelines for creating and operating bridge digital twins. Section 3.4: *Characteristics of digital twins of bridges* describe the key characteristics.

3.2 Techniques of civil engineering digital twinning

This section describes the techniques proposed as enablers in the evolutionary approach to civil engineering digital twinning. The description regards the principles of techniques and places them in the digital twinning concept. The list of digital twinning techniques expands, especially given the fast development of civil engineering and digital innovations [142].

3.2.1 Building Information Modeling (BIM)

Building Information Modeling (BIM) is the AEC approach to managing the information in the entire lifecycle of facilities. The need for holistic management results from extensive data loss in conventionally-conducted AEC projects [143]; digital BIM workflow is the approach to minimize the loss and effectively utilize the data. BIM, at its core, is the methodology of creating, storing, and managing information. Theoretically, it can be applied with CAD techniques and tools. In practice, however, BIM methodology is conjugated with BIM models and modern digital software.

The idea for virtualizing civil engineering facilities as semantic-rich models originated decades ago. Charles Eastman's "Outline for the Building Description System" [144] from 1974 is commonly recognized as the start of the BIM concept. The idea has been researched independently by various research communities – Jerzy Weseli's "Teoretyczne podstawy opisu i analizy uogólnionego modelu układu most – środowisko" ("Theoretical basis for description and analysis of the generalized bridge-environment system") [145] from 1987 also introduced the BIM principles years before the technical advances allowed the practical implementation. Currently, BIM is the leading trend in the digital transformation of AEC. Nonetheless, it is still not a common practice.

BIM adoption level and the pressing today's needs reveal a paradox. BIM is undoubtedly a significant advance to traditional CAD techniques. Just like the exchange of drawing boards for CAD software increased effectiveness, now, CAD should make a place for BIM. Nonetheless, although promising initiatives, both *top-down* (introducing BIM through legal authorities) and *bottom-up* (adopting BIM by practitioners), BIM is not yet a common practice – the fight for wide BIM adoption is still ongoing. Here comes the paradox: although BIM is not yet mature and still has a vast potential for enhancing AEC, it may already be almost outdated. BIM is not providing tools to fulfill the modern digitization requirements of multi-industrial cooperation. Therefore, there is a need to keep in mind the approaching need for evolution – from BIM (and other accompanying techniques) to digital twinning. Evolution, not revolution – civil engineering digital twinning can source from current practices.

BIM comprises many techniques, practices, and tools that can be transferred to civil engineering digital twinning. Subsequent sections review them.

3.2.1.1 Maturity levels

BIM maturity level [146] asses the BIM implementation in companies or countries. The level is often referred to in the implementation roadmaps. It regards exchange formats, depth of information, as well as coordination and collaboration techniques. The maturity levels indicate that BIM, as the information management methodology, theoretically can be initiated with CAD, omitting BIM models.

- Level 0:
  Low collaboration with CAD drawings and paper documentation.

- Level 1:
  Partial collaboration with 2D drawings and 3D models in proprietary formats and digital file-based documentation.
- Level 2:
  Full collaboration with federated, discipline-specific BIM models in proprietary formats with central documentation management (Common Data Environment).
- Level 3
  Full integration with interoperable BIM models in standardized formats for the entire lifecycle with cloud-based model-based documentation management.

3.2.1.2 Tools and methodologies: Little BIM/Big BIM, Closed BIM/Open BIM

BIM, at its core, is a methodology. Nonetheless, the methodology comes with a plethora of tools and software. Little BIM and Big BIM [147] divide the software-oriented and methodology-oriented activities.

Little BIM regards implementing software. These activities are non-centralized – companies can choose various tools depending on needs and skills. The BIM software is used to increase the effectiveness. The *littleness* may indicate that these subjects are less important. Nonetheless, proficiency in the tools is essential for successful practical implementations.

Big BIM addresses the methodology issues – practices of information management. Often, these activities are performed on a central level. The Big BIM regards implementing standards and practices to utilize BIM as the lifecycle management technique. The Big BIM, complemented by Little BIM, lets exploiting the full BIM potential.

Closed BIM/Open BIM differentiation is linked with Little BIM/Big BIM. Closed BIM comprises practices of implementing BIM in a vendor-dependent manner. BIM software producers provide a wide variety of tools, often working in convenient integration: considering Autodesk, a Revit model may be generated with Dynamo from AutoCAD drawing, linked with Robot Structural Analysis, and placed in Infraworks. Such a process is alluring for a single company's effectiveness. Nonetheless, relying entirely on vendors' formats poses dangers. First, it hinders migrating to other software if it is recognized as more effective. Second, more importantly, it poses questions about the utilization of models in the entire lifecycle. If the model is to be used for several decades, it should be based not on vendor-secured format but on open standards. The standards are addressed by Open BIM activities, with Industry Foundation Classes (IFC) as the main representative. IFC is an interoperable format, so it should be recognized by various BIM software. In practice, this is still the Open BIM utopia – although IFC import and export software certifications, IFC models still cannot be produced by one software and successfully (without losing data) imported to another. Nonetheless, hopefully, the community demands and software maturing will enhance this Open BIM idea.

An interesting software trend is *opening* the Closed BIM. Software producers spotted the benefits of extending software utilities to the community. Producers often reveal the programs' APIs (Application Programming Interface), which allows the community to create customized add-ins to enhance software capabilities. Most commercial software publishes just part of the API, but some (e.g., Dynamo Visual

Programming) are open-source: all the details of programming implementations can be seen and utilized by users. With the web-oriented trends (e.g., Software as a Service), vendors also create online platforms with web REST APIs (e.g., Forge, Bimplus), opening the platforms' capabilities to customized web applications.

3.2.1.3 BIM dimensions

BIM models are not just 3D geometrical representations; they also include semantics. The level and depth of semantics (inside the model and linked) extend BIM utilization areas – these are described by the BIM dimensions. Typically, 7 BIM dimensions are depicted:

- 3D: geometry (shape),
- 4D: time (scheduling),
- 5D: costs (estimating),
- 6D: sustainability (performance),
- 7D: maintenance (facility management).

The 3D dimension regards the geometrical data. The 4D adds data about time (e.g., predicted construction time and phase), enabling scheduling, also with simulations. The 5D adds costs of materials, elements, and processes, enabling estimating. 6D is linked with sustainability; the data about physical properties and elusive social factors enables extended performance analysis. 7D assumes continual data enhancement to utilize models in lifecycle management.

The classification currently finalizes in seven dimensions. However, the need for thorough virtualization adds new requirements. Digital twins must comprise the capabilities resulting from the dimensions and extend them.

3.2.1.4 Level of Development (LOD)

Level of Development (LOD; sometimes referred to as Level of Details) asses the geometrical and semantic accuracy of BIM model elements. With various institutions declaring their accuracy standards (BIMForum, British Standards Institution, CanBIM, US Institute of Building Documentation [148]), LOD is often ambiguous. Nonetheless, it is most commonly expressed in the range of 100-400 (occasionally 500) [149].

Table 3-1. Level of Development specification (based on [149])

LOD	*Interpretation*
100	The elements are graphically represented as symbols or other generic, but not geometric, representations. The LOD 100 data must be considered approximate.
200	The elements are graphically represented as generic systems, objects, or assemblies with approximate quantities, sizes, shapes, locations, and orientations with optional non-graphic information. The elements are generic placeholders represented as components or space-reserving volumes. The LOD 200 data must be considered approximate.

300	The elements are graphically represented as specific systems, objects, or assemblies with quantities, sizes, shapes, locations, and orientations represented directly in the model with optional non-graphic information.
350	The elements are graphically represented as specific systems, objects, or assemblies with quantities, sizes, shapes, locations, orientations, and interfaces with other building systems represented directly in the model with optional non-graphic information. The model contains parts linking elements, like supports and connections.
400	The elements are graphically represented as specific systems, objects, or assemblies with quantities, sizes, shapes, locations, and orientations, as well as detailing, fabrication, assembly, and installation data with optional non-graphic information. The elements are sufficiently detailed for fabrication processes.
500	The elements are as-built representations of quantities, sizes, shapes, locations, and orientations with optional non-graphic information.

The highest BIM Level of Development represents the as-built representation. It is also the final aim of digital twinning.

3.2.1.5 Common Data Environments (CDE)

The concept of the Common Data Environment (CDE) is vital in BIM implementation and is required for the level 2 maturity level. CDE is a digital platform for storing and sharing models and other files regarding a project. CDE is typically an online cloud-based platform (e.g., Autodesk BIM 360, Trimble Connect, Bentley ProjectWise). It organizes data access with specific roles. It also ensures that all the participants are working on actual data. CDE also enables communication – both with the model-based communication concepts and typical text-messages systems incorporated into CDEs. Requirements for Common Data Environments are stated in ISO 19650 standard [150].

In digital twinning, the digital twin itself acts as a data management system. Nonetheless, the principles of BIM Common Data Environments can be beneficial in designing the digital twinning system architecture.

3.2.1.6 Standardization

The maturity of BIM activities led to thorough standardization. It is a positive trend. Although exaggerated or non-practical standardization can hinder atypical, potentially creative attempts, it is beneficial for wide implementations. In fact, the lack of standardization is expressed as one of the main obstacles to digital twinning [76,151,152]. Table 3-2 lists selected BIM standards. BIM standards can guide civil engineering digital twinning.

Table 3-2. BIM standards

Institution	*Standard*
ISO	ISO 19650-1:2018. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles

	ISO 19650-2:2018. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 2: Delivery phase of the assets
	ISO 19650-3:2020. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 3: Operational phase of the assets
	ISO 19650-4:2022. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 4: Information exchange
	ISO 19650-5:2020. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 5: Security-minded approach to information management
	ISO/CD 19650-6. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 6: Health and Safety (under development)
	ISO 16739-1:2018. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema
	ISO 29481-1:2016. Building information models — Information delivery manual — Part 1: Methodology and format
BSI	PAS 1192-2:2013. Specification for information management for the capital/delivery phase of construction projects using building information modelling (withdrawn; replaced by ISO 19650-2)
	PAS 1192-3:2014. Specification for information management for the operational phase of assets using building information modelling (withdrawn)
	BS 1192-4:2014. Collaborative production of information - Fulfilling employer’s information exchange requirements using COBie. Code of practice (withdrawn)
	PAS 1192-5:2015. Specification for security-minded building information modelling, digital built environments and smart asset management (withdrawn)
	PAS 1192-6:2018. Specification for collaborative sharing and use of structured Health and Safety information using BIM
NBIMS-US	National BIM Standard - United States V3

3.2.1.7 BIM model is not a digital twin

BIM introduced extended virtualization to AEC. The facilities are reflected not merely by CAD drawings, containing only symbolic geometrical representations – BIM brings 3D modeling and additional semantics. These are usually implemented first in the BIM adoption path. However, the mature BIM also assumes the virtualization of processes. The ideas like “Build it Twice” [153] are implemented with thorough simulations based on reliable BIM models. Additionally, BIM models are thought to be utilized in the entire facilities’ lifecycles. This resembles BIM into digital twinning. Because of that, BIM models are often perceived as digital twins. However, this view is not entirely correct. BIM methodology can be a component of civil engineering digital twinning. However, a BIM model is not a digital twin.

Researchers are still unsure about the BIM-digital twinning relation. In most primitive approaches, BIM models are perceived as digital twins [81]. Others see BIM as a starting point or a technology serving digital twinning [101] and highlight the role of BIM in the digital twinning transformation [100]. Some warn about seeing BIM as the progression or extension of the BIM model [81]. Others propose an evolutionary approach comprising BIM [80].

The here-proposed approach assumes that BIM is the component of civil engineering digital twinning; BIM practices, tools, and standards can be beneficial in the transformation. Nonetheless, acknowledging the differences between BIM and digital twinning is essential for designing effective digital twin systems.

The first difference, which is also a key motivation for implementing digital twinning, is interoperability. Digital twinning is multi-industrial; BIM is AEC oriented. The digitized world will require collaboration between digital models based on multi-industrial standards. Stopping the digital advances at the level of AEC-oriented BIM will exclude civil engineering from the cooperation. BIM introduces interoperability, but in terms of a single project, e.g., concise data exchange between one project's models and files. Digital twinning adds interoperability between various models of the same and various domains. This comprehensive cooperation is crucial for implementing the concept of smart cities.

Other differences regard data management approaches. BIM uses data passively; it is more concerned about data storage. In digital twinning, data is used actively to distinguish patterns and train intelligent algorithms. It leads to a difference in BIM and digital twinning views on autonomy and automation. Intelligent algorithms are an indistinct part of digital twins. In BIM, the algorithms are merely an optional use case. BIM increases the effectiveness of AEC processes but lacks thorough automation [80]. Digital twinning, on the other hand, aims at developing autonomous systems, which is the ultimate automation stage.

Researchers acknowledge a difference in BIM models and digital twins' usage. BIM is perceived as a design and construction aid, while DTs are to be used in the maintenance stage [83]. This view should be discussed. BIM models are indeed used chiefly for design and construction in practice, but it is mostly because of the BIM implementation maturity and technical obstacles; at the matured level, the BIM model should *live* with an object through its lifecycle. Nonetheless, the lack of established practices for IoT integration [80] and dynamic data management [83] hinders implementing the "Living BIM". The view that digital twins are only for maintenance is also not quite right. Although the maintenance stage reveals the full digital twinning potential, digital twins should originate at the design stage and be utilized during the before-maintenance phases. Therefore, although the mature implementations should not differ in utilization phases, the compatibility with technologies favors digital twins as the lifecycle counterparts.

Digital twinning and BIM differ in operating philosophy. BIM is the methodology of working with information. The BIM model is a tool to implement the methodology, but theoretically, BIM is possible without BIM models. In digital twinning, the model, digital twin, is the indispensable component, and methodologies (e.g., model-based engineering [154], data-driven design [48], communication by simulation [155]) result from the model-centrism approach. These issues, however, are not so important for practical implementations and day-to-day usage.

What is vital for everyday practice is the difference in data exchange formats. BIM still oscillates in the Closed BIM areas, relying mostly on vendor-dependent formats. The open formats (e.g., IFC) implementation states challenges for practitioners, so it is often neglected. But digital twinning – because of the need for interoperability – requires open formats. Therefore, digital twinning benefits must be clear and compelling enough to allure practitioners into implementation.

The last of the listed differences regards standardization. BIM is already standardized; digital twinning is not. It is another purpose for incorporating BIM into civil engineering digital twinning. The BIM standards and practices can guide civil engineering digital twinning.

3.2.2 Industry Foundation Classes (IFC)

Industry Foundation Classes (IFC) is a schema for the digital description of AEC assets [156]. IFC is open and standardized (ISO 16739-1:2018 [157]), which makes it a vendor-independent, interoperable data format. IFC codifies objects, semantics, attributes, relationships, abstractions, processes, and people participating in AEC projects. IFC is the key component of the Open BIM trend.

Given its open interoperable characteristics, IFC is the proposed schema for central models of civil engineering digital twins. Stating IFC as the digital twin center is also beneficial for practical implementations. Most BIM modeling is done with commercial software able to export IFC. The BIM practitioners are skilled in the software, so IFC-based digital twinning is compatible with modeling competencies. Nonetheless, it is important to remember that relying completely on a vendor's environment might hinder digital twinning interoperability – purely IFC-based systems are the aim of the proposed solution.

Although IFC is included in BIM, its importance for the proposed digital twinning framework inclines to a more thorough description in a separate section. This section describes the origins and concepts of Industry Foundation Classes. The IFC description focuses on novel IFC 4.3 TC1; although this schema is not yet official (currently under ISO voting), it includes bridges.

3.2.2.1 Origins and development

IFC has been developed and still is being enhanced by buildingSMART. The organization was founded in 1995 as the International Alliance for Interoperability; it was renamed buildingSMART in 2003. In 1997, the organization released the IFC version 1.0. The current official version, IFC 4 ADD 2 TC1 (4.0.2.1), was released in October 2017. This version is still focused mainly on buildings, lacking entities for semantically-reliable modeling of infrastructure, but ongoing buildingSMART works address also this area. IFC 4.1 (4.1.0.0), released in June 2018, provided the basis for describing rails, roads, tunnels, ports, and waterways. IFC 4.2 (4.2.0.0), released in April 2019, addressed bridges. However, these versions have been withdrawn and are now unofficial. IFC 4.3 TC1 (4.3.0.1), currently under ISO voting, restores the infrastructure entities, including bridges. Plans for IFC 4.4 assumes additional infrastructure functionalities, mainly for tunnels.

3.2.2.2 Multi-industrial base

IFC is AEC-specified but is technically based on general, multi-industrial schemas. It has been originated with EXPRESS data modeling language (ISO 10303-11:2004 [158]). The EXPRESS, in the form of STEP format, is the most common way to code IFC data [159]. Nonetheless, IFC also incorporates other formats, e.g., XML (Extensible Markup Language) and JSON (JavaScript Object Notation). The general-format open-source base and activities to enhance the compatible formats list increase the IFC potential for multi-industrial interoperability.

3.2.2.3 IFC architecture

IFC data is modeled by entities. The IFC entities enabled coding data of various types and nature: objects, semantics, attributes, relationships, abstractions, processes, and people. The multitude of entities makes the standardization extensive. To order the entities and modeling workflow, IFC standardization is structured with layers.

IFC comprises four conceptual layers: Core, Interop, Domain, and Resource (Fig. 3-2). The layers are further grouped for more thorough semantic division. Every IFC entity is placed in one group regarding its semantical attachment. The layers also order the logical workflow of modeling data. The high-level abstract entities are often superior to the more specific ones on the lower level of the IFC entities hierarchy. For instance, both *IfcBuilding* and *IfcBridge*, representing specific facilities, inherit from abstract *IfcSpatialElement*; *IfcBuilding* and *IfcBridge*, therefore, share common *IfcSpatialElement* features.

The following description of layers and their parts comprises many examples of IFC entities. The names of IFC entities, especially those rarely used, properly reflect the capabilities of IFC data modeling – IFC enables modeling data exceeding typical single elements’ geometries and semantics.

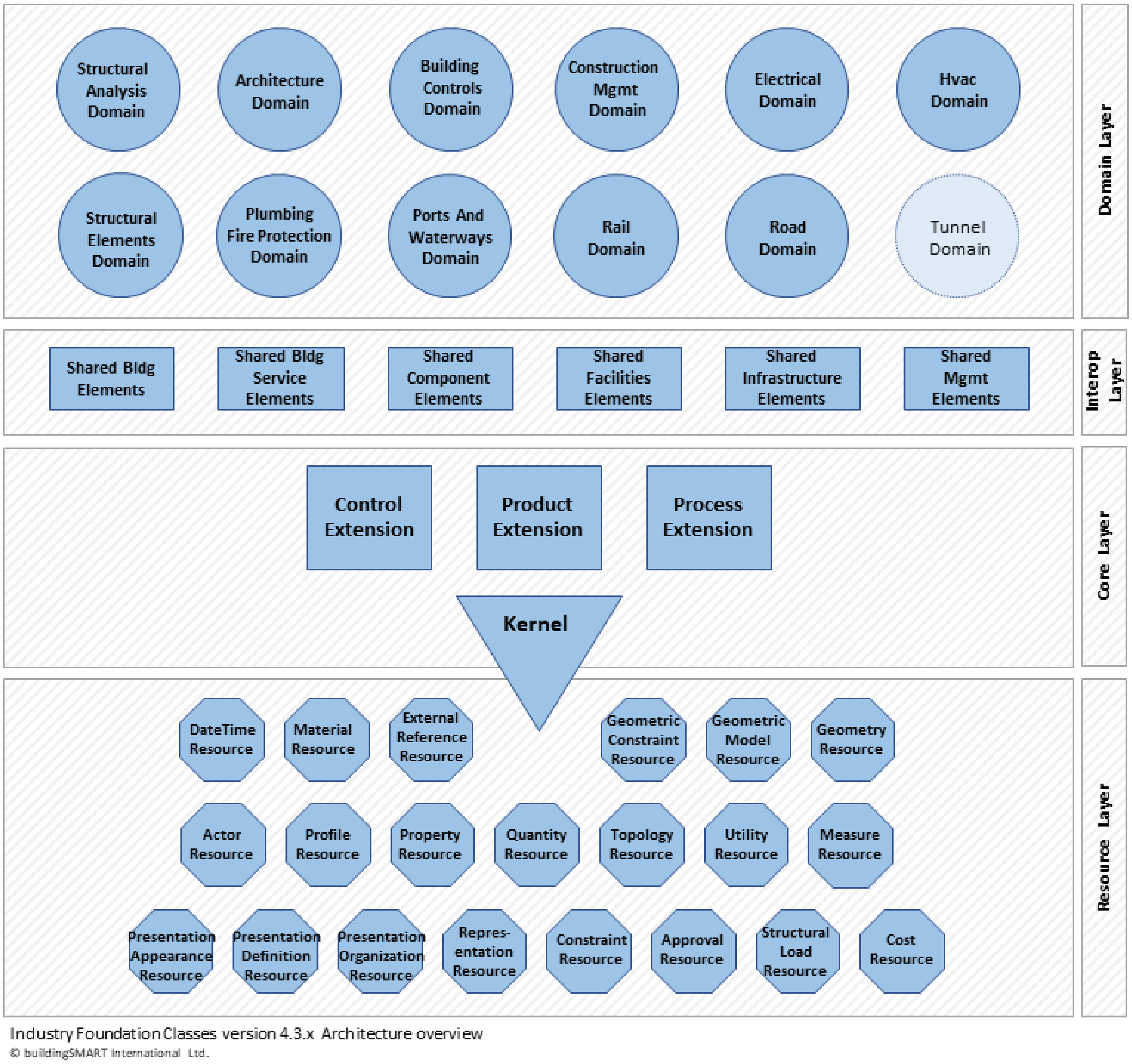


Fig. 3-2. Industry Foundation Classes architecture layers and parts [160]

The most general Core Layer comprises fundamental abstract entities for data modeling. The Core Layer is divided into Kernel, Control Extension, Product Extension, and Process Extension. *IfcKernel* schema comprises most abstract, non-AEC specified entities, e.g., *IfcRoot* (base class for the majority of entities), *IfcActor* (for all human agents), *IfcProduct* (for any geometrical or spatial object), and *IfcProcess* (for activities and events). It also specifies base classes for relationships, e.g., *IfcRelAggregates*. *IfcControlExtension* schema specifies controlling objects and information flow, e.g., *IfcPerformanceHistory*. *IfcProcessExtension* schema enables coding the planning, scheduling, and documenting processes, e.g., *IfcEvent*, *IfcTask*, *IfcWorkSchedule*. *IfcProductExtension* comprises entities for coding elements. The elements can be physical, e.g., *IfcBuilding*, *IfcFacility*, or non-physical, e.g., *IfcAlignment*, *IfcGrid*.

The Interop layer contains shared element data schemas. These entities are on the mid-level of generalization – they are not as general as the Core layer entities but not as specialized as the ones of the Domain layer. The Interop layer is grouped into:

- *IfcSharedBldgElements*
  Subtypes of *IfcBuildingElement* (e.g., *IfcBeam*, *IfcColumn*, *IfcBearing*).
- *IfcSharedBldgServiceElements*
  Building services (e.g., *IfcFlowController*, *IfcFlowTerminal*, *IfcDistributionPort*).
- *IfcSharedComponentElements*
  Small parts and accessories (e.g., *IfcDiscreteAccessory*, *IfcFastener*, *IfcElementComponent*).
- *IfcSharedFacilitiesElements*
  Management of assets, especially non-permanent (e.g., *IfcAsset*, *IfcFurniture*, *IfcOccupant*).
- *IfcSharedMgmtElements*
  Basic lifecycle management (e.g., *IfcActionRequest*, *IfcCostItem*, *IfcPermit*).
- *IfcSharedInfrastructureElements*
  Infrastructure (e.g., *IfcGeomodel*, *IfcPavement*, *IfcSign*).

The Domain layer addresses domain-specific requirements. This layer specifies concepts introduced by the Interop layer. The Domain layer is grouped into:

- *IfcArchitectureDomain*
  e.g., *IfcDoorLiningProperties*, *IfcPermeableCoveringProperties*, *IfcWindowPanelProperties*.
- *IfcBuildingControlsDomain*
  e.g., *IfcAlarm*, *IfcController*, *IfcSensor*.
- *IfcConstructionMgmtDomain*
  e.g., *IfcConstructionMaterialResource*, *IfcCrewResource*, *IfcSubContractResource*.
- *IfcElectricalDomain*
  e.g., *IfcCableFitting*, *IfcLamp*, *IfcSolarDevice*.
- *IfcHvacDomain*
  e.g., *IfcAirTerminal*, *IfcFan*, *IfcValve*.
- *IfcPlumbingFireProtectionDomain*
  e.g., *IfcFireSuppressionTerminal*, *IfcSanitaryTerminal*, *IfcWasteTerminal*.
- *IfcPortsAndWaterwaysDomain*
  e.g., *IfcLiquidTerminal*, *IfcMarinePart*, *IfcNavigationElement*.
- *IfcRailDomain*
  e.g., *IfcKerb*, *IfcRoad*, *IfcRoadPart*.
- *IfcRoadDomain*
  e.g., *IfcDoorLiningProperties*, *IfcPermeableCoveringProperties*, *IfcWindowPanelProperties*.
- *IfcStructuralAnalysisDomain*
  e.g., *IfcStructuralItem*, *IfcStructuralLoadCase*, *IfcStructuralReaction*.

- *IfcStructuralElementsDomain*
  e.g., *IfcFooting*, *IfcReinforcingBar*, *IfcTendon*.

The last of the layers, the Resource, is not in the same inheritance hierarchy that Interop and Domain layers. It is not focused on entities to model elements but rather on tools to enhance the modeled elements' data. The most often used features of the Resource layer, especially in the current stage of IFC adoption, are the ones addressing modeling geometry. Nonetheless, the Resource layer gives tools for a much more thorough enhancement of model data. The Resource layer is grouped into:

- *IfcActorResource*
  e.g., *IfcAdress*, *IfcPerson*, *IfcOrganization*.
- *IfcApprovalResource*
  e.g., *IfcApproval*, *IfcApprovalRelationship*, *IfcResourceApprovalRelationship*.
- *IfcConstraintResource*
  e.g., *IfcConstraint*, *IfcMetric*, *IfcObjective*.
- *IfcCostResource*
  e.g., *IfcAppliedValue*, *IfcCostValue*, *IfcCurrencyRelationship*.
- *IfcDateTimeResource*
  e.g., *IfcDate*, *IfcDuration*, *IfcTime*.
- *IfcExternalReferenceResource*
  e.g., *IfcDocumentInformation*, *IfcExternalInformation*, *IfcExternalReference*.
- *IfcGeometricConstraintResource*
  e.g., *IfcGridAxis*, *IfcLinearPlacement*, *IfcLocalPlacement*.
- *IfcGeometricModelResource*
  e.g., *IfcCartesianPointList3D*, *IfcExtrudedAreaSolid*, *IfcSphere*.
- *IfcGeometryResource*
  e.g., *IfcPoint*, *IfcCurve*, *IfcVector*.
- *IfcMaterialResource*
  e.g., *IfcAlarm*, *IfcController*, *IfcSensor*.
- *IfcMeasureResource*
  e.g., *IfcValue*, *IfcLength*, *IfcUnit*.
- *IfcPresentationAppearanceResource*
  e.g., *IfcColour*, *IfcFontStyle*, *IfcImageTexture*.
- *IfcPresentationDefinitionResource*
  e.g., *IfcAnnotationFillArea*, *IfcPresentationItem*, *IfcTextLiteral*.
- *IfcPresentationOrganizationResource*
  e.g., *IfcLightIntensityDistribution*, *IfcLightSource*, *IfcPresentationLayerAssignment*.

- *IfcProfileResource*
  e.g., *IfcArbitraryClosedProfileDef*, *IfcParametrizedProfile*, *IfcProfileProperties*.
- *IfcPropertyResource*
  e.g., *IfcProperty*, *IfcPropertySingleValue*, *IfcPropertyEnumeration*.
- *IfcQuantityResource*
  e.g., *IfcPhysicalQuantity*, *IfcQuantityArea*, *IfcQuantityTime*.
- *IfcRepresentationResource*
  e.g., *IfcGeometricRepresentationContext*, *IfcProductDefinitionShape*, *IfcShapeModel*.
- *IfcStructuralLoadResource*
  e.g., *IfcBoundaryCondition*, *IfcStructuralLoadConfiguration*, *IfcStructuralLoadSingleForce*.
- *IfcTopologyResource*
  e.g., *IfcEdgeLoop*, *IfcFace*, *IfcOpenShell*.
- *IfcUtilityResource*
  e.g., *IfcApplication*, *IfcOwnerHistory*, *IfcGloballyUniqueId*.

The entities are the core of IFC, but it also includes additional explicit specification and implicit modeling concepts. The specification of *Pset*s standardizes the entities' properties. The IFC entities can be freely linked with various properties. It gives the great freedom of enhancing elements' data but hinders the uniformity of models (e.g., a beam can have dimensions properties called "width" and "height" in one model but "L" and "H" in another; also, the set of properties in both models can be different, depending on a modeler). To address that, IFC standardization includes *Pset*s, the standardized property sets for various entities. The *Pset*s state names, property types, and data types that should characterize objects. An entity can have numerous *Pset*s, depending on data-richness needs; IFC documentation suggests which *Pset*s can be attached to which entities. Table 3-3 lists exemplary *Pset*s.

Table 3-3. Exemplary IFC *Pset*s (based on [161])

Pset	*Property Name*	*Property Type*	*Data Type*	*Description*
Pset_ Bridge Common	StructureIndicator	*IfcPropertyEnumeratedValue*	*PEnum_ StructureIndicator*	Type of the structure (*COATED*, *COMPOSITE*, *HOMOGENEOUS*).
Pset_ Bearing Common	Displacement Accommodated	*IfcPropertyListValue*	*IfcBoolean*	A list of three booleans representing displacement capabilities on X, Y, Z axes.
	Rotation Accommodated	*IfcPropertyListValue*	*IfcBoolean*	A list of three booleans representing rotation capabilities on X, Y, Z axes.

Besides explicit standardization, IFC documentation describes concepts and assumptions for data modeling. These guide how to, e.g., assign objects, add their attributes, or define geometries.

3.2.2.4 IFC for bridges and infrastructure

For most of its existence, IFC has been oriented mainly on buildings, omitting infrastructure facilities. It forced BIM bridge practitioners to look for workarounds (e.g., modeling all bridge parts as general entity *IfcBuildingElementProxy*, because not implementing semantic information about element type is better than coding untruth by modeling bridge elements as non-bridge classes). The recognized need for infrastructure-compatible IFC resulted in initiating the IFC Bridge project held by buildingSMART Infrastructure Room. The project extended the IFC alignment activities and proposed concepts for coding bridge data, included in the IFC 4.2 candidate release [162,163]. Although IFC 4.2 has been withdrawn, bridges and other infrastructure facilities are included in the recent IFC 4.3 TC1 (4.3.0.1), currently under ISO voting. This section reviews the infrastructure-oriented data modeling capabilities of IFC 4.3 TC1, focusing on bridges.

IFC Bridge project aimed at addressing the most common bridge types. The following types have been considered as the base for the project:

- slab bridge,
- girder bridge,
- slab-girder bridge,
- box-girder bridge,
- frame bridge,
- rigid frame bridge,
- culvert.

The project also assumed the possibility of modeling additional types (resulting from concepts regarding the base group):

- truss bridge,
- arch bridge,
- cantilever bridge,
- cable-stayed bridge,
- suspension bridge.

In terms of materials, the following groups have been addressed:

- reinforced concrete bridges,
- prestressed concrete bridges,
- steel/concrete composite bridges,
- steel girder bridges,
- steel bridges.

The project also addressed the extensive set of use cases able to be modeled by existing or new entities:

- initial state modeling,

- import of major road / railway parameters,
- technical visualization,
- coordination / collison detection,
- 4d construction sequence modeling,
- quantity take-off,
- progress monitoring,
- as-built vs. as-planned comparison,
- handover to asset management,
- handover to gis for spatial analysis,
- design to design (reference model).

Bridges have been addressed by introducing new IFC entities and extending the existing ones. The modifications regarded various levels of data schema hierarchy and modeling concepts.

Modeling IFC objects starts with defining a spatial entity – a *master* entity (or set of entities) *containing* others. In the building-oriented IFC, a typical solution was *IfcSite* containing *IfcBuilding.* IFC 4.3 introduced new high-level entities, enabling modeling infrastructure:

- *IfcSpatialStructureElement* subclasses (matching *IfcSite* hierarchy level):
  - *IfcFacility,*
  - *IfcFacilityPart.*
- *IfcFacility* subclasses (matching *IfcBuilding* hierarchy level):
  - *IfcBridge,*
  - *IfcMarineFacility,*
  - *IfcRailway,*
  - *IfcRoad.*
- *IfcFacilityPart* subclasses:
  - *IfcBridgePart,*
  - *IfcFacilityPartCommon,*
  - *IfcMarinePart,*
  - *IfcRailwayPart,*
  - *IfcRoadPart.*

Besides entities, also matching enumerators (entities listing available values) have been specified, for instance:

- *IfcBridgeTypeEnum:*
  ARCHED, CABLE_STAYED, CANTILEVER, CULVERT, FRAMEWORK, GIRDER, SUSPENSION, TRUSS, USERDEFINED, NOTDEFINED.
- *IfcBridgePartTypeEnum*:
  ABUTMENT, DECK, DECK_SEGMENT, FOUNDATION, PIER, PIER_SEGMENT, PYLON,

SUBSTRUCTURE, SUPERSTRUCTURE, SURFACESTRUCTURE, USERDEFINED, NOTDEFINED.

The listed entities show that bridge elements do not have specific entities (like *IfcDoor* or *IfcWindow* for buildings' elements) but rather should be modeled by general *IfcBridgePart* with the enum parameter specifying the element type. This approach follows current assumptions of limiting the IFC schema entities number [162]. Nonetheless, also some specific infrastructure-related entities have been added, for instance:

- *IfcBearing,*
- *IfcSign,*
- *IfcVehicle.*

Another concept for modeling bridge elements is the usage of previously available entities but attaching to them newly introduced infrastructure-related types, for instance:

- *IfcBeam:*
  new *IfcBeamTypeEnum* values: *CORNICE, DIAPHRAGM, EDGEBEAM, GIRDER_SEGMENT, HATSTONE, PIERCAP.*
- *IfcSlab:*
  new *IfcSlabTypeEnum* values: *APPRAOCH_SLAB, PAVING, SIDEWALK, TRACKSLAB, WEARING.*
- *IfcWall:*
  new *IfcWallTypeEnum* values: *RETAININGWALL, WAVEWALL.*
- *IfcColumn:*
  new *IfcColumnTypeEnum* values: *PIERSTEM, PIERSTEM_SEGMENT, STANDCOLUMN.*
- *IfcSensor:*
  new *IfcSensorTypeEnum* values: *EARTHQUAKESENSOR, FOREIGNOBJECTDETECTIONSENSOR, OBSTACLESENSOR, RAINSENSOR, SNOWDEPTHSENSOR, TRAINSENSOR, TURNOUTCLOSURESENSOR, WHEELSENSOR.*

Another advance is *IfcAlignment*. It serves two roles. First, given the importance of the alignment course for bridges and roads, it encodes crucial project data. Second, as the *IfcLinearPositioningElement* subtype, it enables placing other elements in relation to the *IfcAlignent* instance, simplifying (or, to some extent, even parametrizing) modeling IFC geometries.

3.2.2.5 Modeling IFC

This section presents an example of modeling IFC data. The example regards spatial hierarchy, geometry, semantics, and external sources. The model is based on IFC 4x3 schema and has been generated using the Xbim toolkit [164], a library for creating and analyzing IFC programmatically.

Listing 3-1. Modeling IFC: header

```
ISO-10303-21;


HEADER;
FILE_DESCRIPTION ((''), '2;1');
FILE_NAME ('Bridge model example', '2023-01-14T17:42:13', 'Kamil Korus',
'Silesian University of Technology', 'Processor version 5.1.0.0', 'Xbim.Common', '');
FILE_SCHEMA (('IFC4X3_ADD1'));
ENDSEC;
```

Listing 3-1 presents the IFC header. It contains basic information about the file: name, time stamp, creator and a related organization, used software/processor, view definition, and data schema. The schema is an essential information for processing software, e.g., viewers.

Listing 3-2. Modeling IFC: project initiation and units

```
DATA;
#1=IFCPROJECT('0vGCH5MKb7IxXhcP$Yz3$a',$,'Bridge project',
'This project is a simple example of modeling a bridge',$,$,$,(#14,#17),#2);


#2=IFCUNITASSIGNMENT((#3,#4,#5,#6,#7,#8,#9,#10,#11));
#3=IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#4=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#5=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#6=IFCSIUNIT(*,.SOLIDANGLEUNIT.,$,.STERADIAN.);
#7=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#8=IFCSIUNIT(*,.MASSUNIT.,$,.GRAM.);
#9=IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
#10=IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
#11=IFCSIUNIT(*,.LUMINOUSINTENSITYUNIT.,$,.LUMEN.);
```

IFC data is coded with entities. One of the basic entities is *IfcProject* (Listing 3-2). *IfcProject* instance attributes start with the global identifier (the global identifier is attached to all the entities inheriting from *IfcRoot*). Next are properties like name or description. The numbers preceded with hashtags (e.g., #14) refer to other entities. ($14, #17) links the project with its geometric representation contexts (Listing 3-3). #2 attaches declared project units.

Listing 3-3. Modeling IFC: geometric representation contexts

```
#12=IFCCARTESIANPOINT((0.,0.,0.));
#13=IFCAXIS2PLACEMENT3D(#12,$,$);
#14=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.E-05,#13,$);
#15=IFCCARTESIANPOINT((0.,0.));
#16=IFCAXIS2PLACEMENT2D(#15,$);
#17=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Plan',2,1.E-05,#16,$);
```

Geometric representation contexts (Listing 3-3) can be perceived as containers for objects' representations. A project can comprise several contexts, e.g., 3D ("3" in #14) or 2D ("2" in #17). A context has coordinate systems (declared with IfcAxis2Placement) and declared precision (1.E-05).

Listing 3-4. Modeling IFC: basic geometry primitives

```
#18=IFCLOCALPLACEMENT($,#13);
#19=IFCDIRECTION((1.,0.,0.));
#20=IFCDIRECTION((0.,1.,0.));
#21=IFCDIRECTION((0.,0.,1.));
```

Listing 3-4 presents basic geometric primitives. #18 defines the local placement referring to the point "0, 0, 0" (#13 and #12). #19, #20, and #21 define basic axes. The basic primitives are useful in further modeling.

Listing 3-5. Modeling IFC: spatial elements

```
#22=IFCFACILITY('3eRo4UYs96VvgzHYlN5rwP',$,'Facility',$,$,$,$,$,$);
#23=IFCRELAGGREGATES('1zjaZZpoz2EwQTKxCE3dQj',$,$,$,#1,(#22));

#24=IFCBRIDGE('2pyXrKuJX2hROBEU2KQ1rq',$,'Bridge',$,$,$,$,$,$,.GIRDER.);
#25=IFCRELCONTAINEDINSPATIALSTRUCTURE('2PeVAh2uH2phENraW6lvHj',$,$,$,(#24),#22);

#26=IFCBRIDGEPART('2N8pyojwb4Bgw07KZBEF3O',$,'Superstructure',$,$,#36,#55,$,$,.USERDEFINED.,
.SUPERSTRUCTURE.);
#27=IFCRELCONTAINEDINSPATIALSTRUCTURE('1LqnXVBdfEMwYGj9_UUshh',$,$,$,(#26,#30),#24);
```

Listing 3-5 presents the hierarchy of spatial elements. *IfcFacility* (#22) is the basic spatial container in this project. *IfcRelAggregates* and *IfcRelContainedInSpatialStructure* codes relationships. #23 attaches *IfcFacility* to the *IfcProject*. Next in the hierarchy are *IfcBridge* (attached as part of the *IfcFacility*) and *IfcBridgePart* (attached as a part of the *IfcBridge*), representing the bridge's superstructure.

The spatial entities have various attributes. Some of them are free-definable strings (e.g., "Bridge" in #24), and others are enumerators of pre-defined values (e.g., ".SUPERSTRUCTURE." defining the *PredefinedType* of *IfcBridgePart*).

Listing 3-6. Modeling IFC: attaching material

```
#28=IFCMATERIAL('Concrete',$,$);
#29=IFCRELASSOCIATESMATERIAL('295_zxIQT9QBPS_FCpe59Y',$,$,$,(#26),#28);
```

Listing 3-6 comprises declaring a material and attaching it to the superstructure's *IfcBridgePart*. The material is an example of semantic data. The material (similar to other objects) can have properties (omitted in the example), further increasing the data-richness.

Listing 3-7. Modeling IFC: alignment

```
#30=IFCALIGNMENT('2eDu6iXCn6qfh2qIv8t0P2',$,$,$,$,#18,#35,$);
#31=IFCPOLYLINE((#32,#33));
#32=IFCCARTESIANPOINT((0.,0.,0.));
#33=IFCCARTESIANPOINT((20000.,0.,400.));
#34=IFCSHAPEREPRESENTATION(#14,$,'Curve3D',(#31));
#35=IFCPRODUCTDEFINITIONSHAPE('Alignment shape definition',$,(#34));
```

Listing 3-7 present modeling the alignment. The alignment refers to the "0, 0, 0" placement (#18) and is equipped with an *IfcProductDefinitionShape* (#35) declaring the alignment's profile. Modeling the profile starts with *IfcCurve*, here in the form of *IfcPolyline* (#31) comprising two points (#32, #33). *IfcShapeRepresentation* (#34) attaches the *IfcPolyline* to the *IfcGeometricRepresentationContext* (#14) and states the shape's type ("Curve3D").

Listing 3-8. Modeling IFC: alignment

```
/* Option 1: placement with an unrelated point
#36=IFCLOCALPLACEMENT($,#37);
#37=IFCAXIS2PLACEMENT3D(#38,$,$);
#38=IFCCARTESIANPOINT((0.,0.,-80.));

/* Option 2: placement with a point on the alignment curve
#36=IFCLOCALPLACEMENT($,#37);
#37=IFCAXIS2PLACEMENT3D(#38,$,$);
#38=IFCPOINTONCURVE(#31,0.);

/* Option 3: placement with a point in relation to the alignment curve
#36=IFCLOCALPLACEMENT($,#37);
#37=IFCAXIS2PLACEMENT3D(#38,$,$);
#38=IFCPOINTBYDISTANCEEXPRESSION(IFCNONNEGATIVELENGTHMEASURE(1000.),0.,-80.,0.,#31);

/* Option 4: linear placement in relation to the alignment curve
#36=IFCLINEARPLACEMENT($,#37,$);
#37=IFCAXIS2PLACEMENTLINEAR(#38,$,$);
#38=IFCPOINTBYDISTANCEEXPRESSION(IFCNONNEGATIVELENGTHMEASURE(0.),0.,-80.,0.,#31);
```

The superstructure's *IfcBridgePart* (#26) refers to a placement (#36; shown on Listing 3-5). IFC schema gives numerous options for coding the placement (Listing 3-8). The most basic is placement in an unrelated point (option 1). The next options are also based on points but in relation to the alignment curve: option 2 uses *IfcPointOnCurve* to get a point on the alignment, and option 3 uses a new schema resource, *IfcPointByDistanceExpression,* to get a point with possible offsets from the alignment. Option 4 presents a recent schema concept of linear placement relating directly to the alignment curve. Unfortunately, not all the options are yet implemented by IFC software (e.g., viewers).

Listing 3-9. Modeling IFC: profile (superstructure cross-section)

```
#39=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#40);
#40=IFCPOLYLINE((#41,#42,#43,#44,#45,#46,#47,#48,#49,#50,#51,#41));
#41=IFCCARTESIANPOINT((0.,0.));
#42=IFCCARTESIANPOINT((-3800.,-76.));
#43=IFCCARTESIANPOINT((-6750.,-17.));
#44=IFCCARTESIANPOINT((-6750.,-237.));
#45=IFCCARTESIANPOINT((-4300.,-356.));
#46=IFCCARTESIANPOINT((-3900.,-1100.));
#47=IFCCARTESIANPOINT((3900.,-1100.));
#48=IFCCARTESIANPOINT((4300.,-356.));
#49=IFCCARTESIANPOINT((6750.,-237.));
#50=IFCCARTESIANPOINT((6750.,-17.));
#51=IFCCARTESIANPOINT((3800.,-76.));
```

Listing 3-9 regards modeling the superstructure geometry, starting with the cross-section profile. The profile (#39) is defined by a closed polyline (#40) based on points (#41-#51).

Listing 3-10. Modeling IFC: geometry representation (superstructure geometry)

```
#53=IFCFIXEDREFERENCESWEPTAREASOLID(#39,$,#31,$,$,$);
#54=IFCSHAPEREPRESENTATION(#14,'Body','SolidModel',(#53));
#55=IFCPRODUCTDEFINITIONSHAPE('Superstructure shape definition',$,(#54));
```

Listing 3-10 continues modeling the superstructure geometry. *IfcFixedReferenceSweptAreaSolid* (#53) is a geometric representation sweeping a profile along a curve; here, the cross-section (#39) is swept along the alignment curve (#31). The resulting geometric representation is *zipped* with *IfcShapeRepresentation* (#54), further referenced by *IfcProductDefinitonShape* (#55). The product definition shape is referenced as an attribute of the superstructure's *IfcBridgePart* superstructure (shown on Listing 3-5). Fig. 3-3 presents the superstructure's geometry and characteristics (visualized in usBIM.viewer+ [165]).

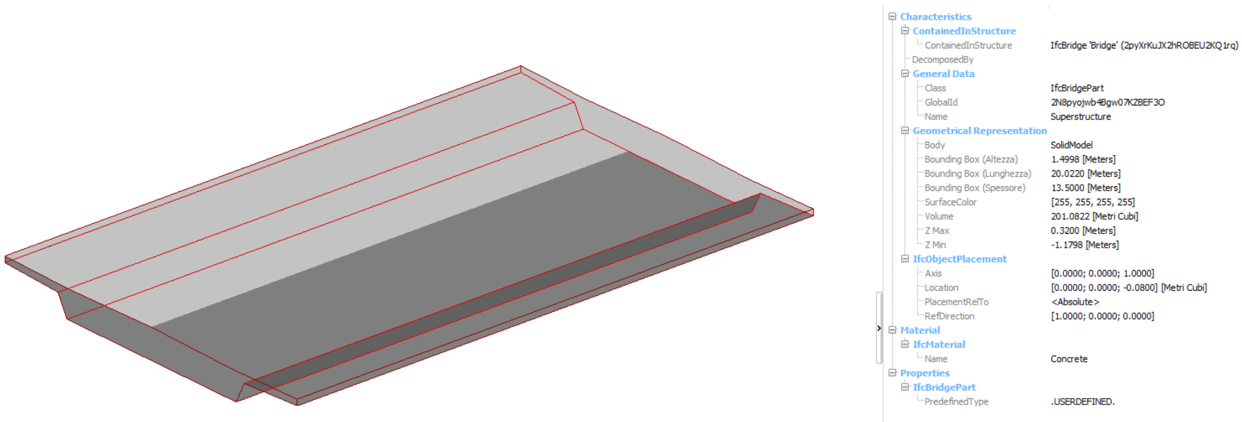


Fig. 3-3. Modeling IFC: superstructure's geometry and characteristics

Listing 3-11. Modeling IFC: external data source (performance history reference)

```
#56=IFCPERFORMANCEHISTORY('2AKe$ZJ1bCxunXTKWapVSF',$,
'Superstructure performance history',$,$,$,'OPERATION',$);
#57=IFCRELAGGREGATES('0dSmTizeDBMREGS$lvAfsh',$,$,$,#26,(#56));
#58=IFCLIBRARYINFORMATION('Superstructure performance database',$,$,$,'
https://www.digitaltwin-database.com/0vGCH5MKb7IxXhcP$Yz3$a/2N8pyojwb4Bgw07KZBEF3O',$);
#59=IFCRELASSOCIATESLIBRARY('01mMvMoyvFJwH8aYXLKCjV',$,$,$,(#56),#58);


ENDSEC;
END-ISO-10303-21;
```

The model ends with attaching an external performance history database to the superstructure (Listing 3-11). This is only one of the IFC possibilities beyond direct modeling geometric and semantic data. The whole model's code is presented in Listing B-1.

3.2.2.6 Horizon of advances

The Open BIM idea begins to get momentum due to industrial requirements and the benefits it provides. IFC is a developing project with a wide horizon of advances.

IFC is most commonly encoded as the STEP file, but it is continually being enhanced with additional compatible formats. Table 3-4 lists the formats and their status. Compatibility with the additional formats is a premise of convenient data exchange and wider interoperability. Another interoperability potential comes from *webifying* the IFC schema by ifcOWL (a developing IFC version of the Web Ontology Language) or OpenCDE API.

Table 3-4. IFC formats (based on [156])

Status	*Format*	*IFC extension*
Official	STEP	ifc
	XML (Extensible Markup Language)	ifcXML
	ZIP	ifcZIP
	Turtle (Terse RDF Triple Language)	ttl

	RDF/XML (Resource Description Framework)	rdf
Candidate	JSON (JavaScript Object Notation)	json
	HDF (Hierarchical Data Format)	hdf
Experimental	SQLite	sqlite

IFC is AEC-oriented, but as an open format, its data can be mapped to be compatible with other formats.

It will enable data exchange between models or including them as parts of bigger systems. Attempts to link IFC models with cities' data formats, like CityGML [159], lay the foundations for systems of future smart cities.

As BIM practice demands the formalization of further concepts, the IFC data schema is not the only representative of the Open BIM trend. Model View Definition (MVD) standard specifies IFC entities (and their data-richness) that should be included in a model to serve specific use cases; examples are *Design Transfer View*, *Quantity Takeoff View*, *Energy Analysis View*, *IFC4Precast* (focusing on data needed for precast automation), and *Bridge Construction View*. Information Delivery Specification (IDS) defines model exchange requirements in a computer-interpretable way; it automates concise validation of IFC export and import. BIM Collaboration Format (BCF) serves standardized communication. Information Delivery Manual (IDM) structures the information workflow between BIM project parties. buildingSMART Data Dictionary (bSDD) attempts to form a concise base describing elements and their properties. Construction Operations Building Information Exchange (COBie) specifies which data should be handed as relevant for facility management. The listed concepts are still being developed and enhanced. They structure BIM usage and widen its interoperability potential.

3.2.3 Structural Health Monitoring (SHM)

Structural health monitoring (SHM), a structure-dedicated Product Health Monitoring (PHM) subset, is a field of engineering that uses sensing techniques to monitor the *behavior* of a structure (reflected in, e.g., responses to loads). SHM aims to identify changes in a structure's condition, especially those potentially indicating damages and malfunctions. That enables alarming potential failures, which increases safety and reduces the total costs of a structure's lifecycle. SHM systems are realized with various sensing devices.

Structural health monitoring emerges in civil engineering use cases. SHM advent is enhanced by technological advances: sensing devices are cheaper and more accessible. It results in more SHM applications but mostly in non-standard cases: SHM systems are usually installed solely in complex untypical structures, structures already showing signs of deterioration, or in scientific approaches. Nonetheless, the SHM application trend uprises.

Bridges, as complex structures with high safety requirements, are a subject of SHM applications. SHM is being applied for various types of bridges regarding their structure (e.g., large-scale [166], long-span [167], cable-stayed [168]) and function [169]. Also, different damage causes are addressed, e.g., flooding [170] or seismic actions [171]. SHM systems are enhanced with new sensing devices [172] and data-acquiring tools, like unmanned aerial vehicles [173] (following the non-contact SHM techniques). The acquired SHM data of various types is being analyzed with emerging techniques like digital image correlation [174,175] or big data algorithms [176]. The bridge SHM research matures with long-term studies, like a 14-years analysis of a prestressed concrete bridge on the A100 Highway in Berlin [177]. SHM is also already utilized in linkage with digital twinning for, e.g., aerostructures [178], ships [179], offshore cranes [180], and also bridges [105].

3.2.3.1 SHM in the digital twinning context

Structural health monitoring enables acquiring extensive data. The data must be properly processed and analyzed. Modern big data techniques enable advanced analysis, but SHM data is insufficient to distill the knowledge and patterns of a facility's behavior – for that, also comprehensive data about the structure operation context, environment, and characteristics are needed. This kind of data can be sourced from digital twins. Therefore, in the digital twinning concept, SHM is a component delivering reliable data about the actual conditions of the physical counterpart. The data is further processed by the digital twin system to depict dedicated patterns leading to a better understanding of the structures' processes.

3.2.4 Artificial Intelligence (AI)

Artificial intelligence (AI) is a group of algorithms mimicking the processes (problem-solving, decision-making, pattern recognition) of living beings' brains (particularly humans). AI systems can automate tasks that are hard to automate with traditional solutions or surpass the traditional solutions' effectiveness. Artificial intelligence has become a standard automation paradigm of Industry 4.0. It is enabled by advances in computation power (mathematical foundations of a plethora of AI algorithms were developed decades ago, but the limitation of computer computation power hindered their practical use) and accessibility to comprehensive data.

AI contains a wide group of solutions. Machine learning (ML) is an AI subset comprising algorithms that are not explicitly programmed but rather optimize their performance with training. Machine learning can be divided regarding the approach to the algorithms' training and use cases [181]:

- supervised learning:
  algorithms trained with labeled data, e.g., classification (attaching distinct categories or labels) and regression (assessing continuous quantity).
- unsupervised learning:
  algorithms trained with unlabeled data, e.g., clustering (grouping) and anomaly detection.
- semi-supervised learning:
  Hybrid algorithms operating on labeled and unlabeled data.
- reinforcement learning:
  algorithms that automatically optimize their behavior to the conditions of the environment in which they operate, e.g., controlling systems.

Deep learning (DL) is a subset of machine learning. Deep learning, with its artificial neural networks (ANN), mimics the functioning of the human brain. Artificial neural networks are powerful algorithms that can perform complicated tasks, e.g., natural language processing, speech recognition, and image analysis.

The family of intelligent algorithms also comprises other meta-heuristics like genetic algorithms, particle swarm optimization, or ant colony optimization; many of these are inspired by nature.

3.2.4.1 AI in civil engineering

Intelligent algorithms have gained the increasing interest of scientific [182] and industrial civil engineering communities. [132] identified six main research topics of civil engineering AI: knowledge representation and reasoning (building knowledge-bases patterns system instead of algorithms or statistics), information fusion (merging data from various sources, e.g., SHM sensors, to distill information), computer vision (processing of images, videos, or point clouds), natural language processing (e.g., interpreting texts in unstructured forms, chatting), intelligent optimization (searching for the optimal solution given an objective function; in civil engineering, often with genetic algorithms [182]), and process mining (analyzing processes distilled from structured reports generated by systems). The variety of research areas is reflected in use cases; only for bridges have intelligent algorithms been applied for, e.g., geometric optimization [183], FEM model updating [184], cracks classification [185], and extracting condition information from inspection reports [186].

3.2.4.2 AI in the digital twinning context

A digital twin acquires diversified and reliable data about the characteristics and processes of a physical object in its entire lifecycle. With such a big data set, the digital twin becomes a data source for intelligent algorithms, which, after training, are utilized in the digital twinning systems. Digital twins comprise algorithms of various kinds: task-dedicated algorithms can be used for automated modeling, detecting malfunctions in the structure behavior, and proposing maintenance solutions; intelligent controlling agents (e.g., reinforcement learning-based) can orchestrate the data exchange between the digital twin’s modules. Such a portfolio of intelligent algorithms transforms the digital twin into an autonomous system.

3.2.5 Visual programming

The ability to develop user-defined algorithms to declare the logic of systems operation is beneficial, especially for domain experts whose requirements are highly specialized. This need for customization leads to the trend of non-code solutions. The non-code solutions are an alternative to traditional text-based programming. Their idea is to give the possibility of creating algorithms in an accessible manner, opening the customization for non-programmers. Visual programming is such a solution.

Visual programming is a non-code alternative to traditional text-based programming. Its accessibility increases its popularity, especially among non-programmers; even a little experience allows for developing scripts for complex tasks [187]. Visual programming techniques are commonly used in non-informatics industries, and many visual programming languages (VPL) serve specific fields [188].

Typical VPL scripts consist of methods graphically represented by blocks, also called nodes. The block input fields pass method parameters, while the output fields convey the results of the block operation. Both input and output fields often take the specific data type: numbers, strings, or instances of a defined class (e.g., Point, Line, Surface). Connected by wires, the blocks constitute a logic network of methods (Fig. 3-4).

The graphical form of the script and instant return of the methods' results provides simplified dataflow control and user-friendly debugging [189].

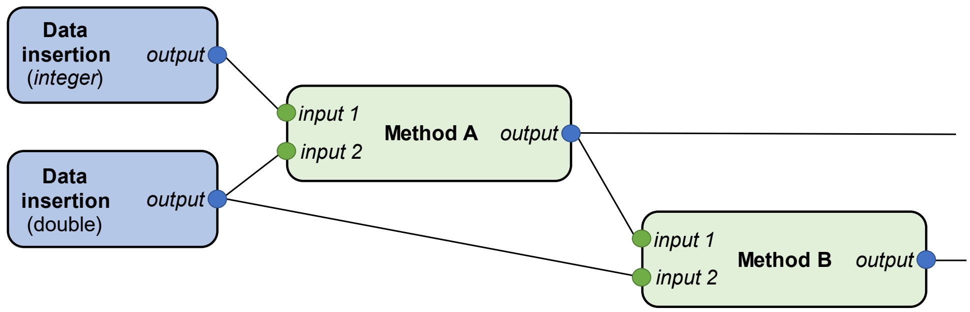


Fig. 3-4. Scheme of a visual programming language script

Visual programming is becoming popular also in architecture and civil engineering, especially in the BIM environment; Dynamo for Autodesk Revit, Grasshopper for Rhinoceros3D, Marionette for Vectorworks, Allplan Visual Scripting for Nemetschek Allplan, and Bentley Generative Components are effective additions to the modeling software. VPL tools are used mostly for parameterized geometry modeling but can also serve other engineering tasks. Hence, extensive research has been conducted in this field. [190] stressed visual programming utility for AEC students, e.g., integrating a generative design with CAD allows for experimenting with new forms and shapes [191]. [192] created VCCL (Visual Code Checking Language) to ensure code compliance of models, including IFC, while [193] checked code compliance of railway BIM design. [194] developed the visual Query Language for 4D Building Models (vQL4BIM) to retrieve BIM models' data. [195] used Grasshopper and EnergyPlus to predict and visualize energy consumption in buildings. A similar software combination predicted aggregate energy demand using GIS data and calibrated multi-zone energy models [196]. [197] used visual programming for energy and shading analysis. Dynamo helped to optimize building energy performance [198] and to perform thermal analyses [199], while Grasshopper allowed daylight simulations [200], design of nearly zero-energy building [201], and forming a variable beam section [202].

3.2.5.1 Visual programming in the digital twinning context

Computer algorithms, including visual programming ones, are based on variables, also called parameters. The parameters are also the basis of components forming semantic models. Both physical objects and digital models are packed with data, including varying parameters. Semantic digital models are based on parametrical geometries and parametrized semantic data. Also, the physical objects are based on parameters that control their operation. Visual programming can serve as a link between the complexity of the digital twins' systems, the need for automation, and domain knowledge. It can be a user-friendly interface to analyze and control digital twins in a customized manner.

3.2.6 Point clouds

A point cloud is a set of points sampled from the surface of a 3D geometry. The points are declared by their 3-dimensional position (x, y, z) and optional parameters (e.g., color as R, G, B; the intensity of the reflection;

surface normal as Nx, Ny, Nz) [203]. Point clouds can be produced by 3D laser scanning (terrestrial (TLS), airborne (ALS), and mobile (MLS)), photogrammetry, videogrammetry, or synthetic aperture radar (SAR) [204]. These techniques vary in practical features, tools requirements, and produced point clouds' properties. The fast development of hardware [205] and software [206] increases the quality of point clouds and results in many applications.

Point clouds are utilized in many sectors [207]: earth and environmental sciences, geodesy, reverse engineering, manufacturing, and forensics. They also are extensively utilized in AEC in each project phase [208]: planning and design (e.g., reconstructions, acquiring design data), manufacturing and construction (e.g., inspections, progress monitoring), and maintenance (e.g., structural health monitoring, performance analysis). These applications take advantage of accurate geometrical representation.

3.2.6.1 Point clouds automation

Point clouds undergo different processing procedures. Data cleaning and registration prepare a raw point cloud for further utilization. Further procedures, depending on the task, may include segmentation, object and shape recognition, model generation, comparing, and interpretation. Since performing these tasks manually is tedious and error-prone, much research effort is put into automation. This section reviews the tasks and approaches to automating them.

Point cloud segmentation is a process of clustering points into segments. It is often the first step of the point cloud automation pipeline. Point cloud segmentation is sometimes defined as object recognition, classification, or labeling.

Segmentation can be based on various principles, considering features and object boundaries. Segmentation into geometric primitives, e.g., surfaces, is the first technique, with Random Sample Consensus (RANSAC) [209,210] and Hough Transform [211,212] as the main representatives. So-called semantic segmentation groups points into semantic elements. The semantic segmentation can be class- or instance-oriented. For instance, having a point cloud of four elements (instances) of two types (classes): "column 1", "column 2", "beam A", and "beam B", class-based segmentation will divide points into two groups regarding the type ("column" or "beam") while instance-based segmentation will divide points into four groups, separating every element ("column 1", "column 2", "beam A", "beam B"). Examples of segmentation into geometric primitives, instance-based, and class-based semantic segmentation are presented in Fig. 3-5. Since the lack of grouping in raw point clouds is their significant drawback, segmentation is a crucial step in the automation pipeline.

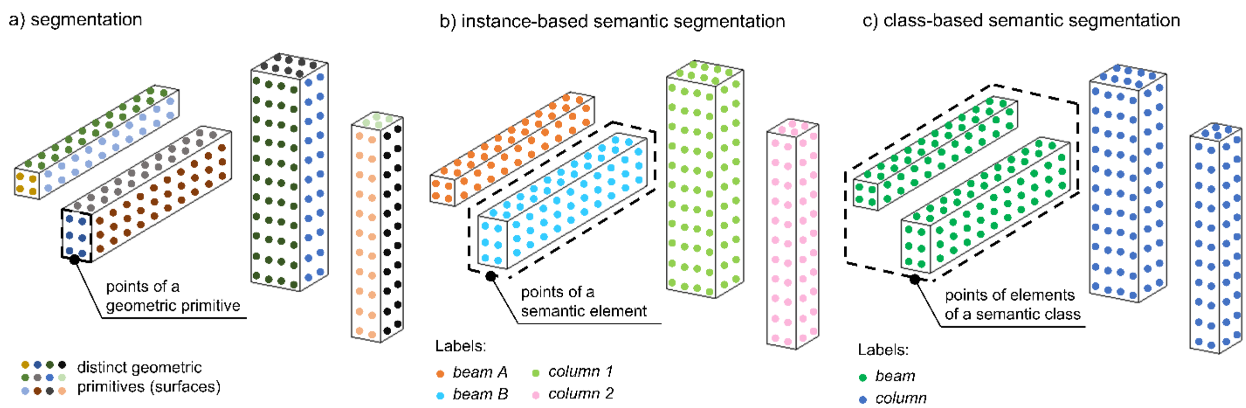


Fig. 3-5. Point clouds' segmentation types: a) segmentation, b) instance-based semantic segmentation, c) class-based semantic segmentation

The automated approaches to segmentation can be model- or data-driven [213]. Model-driven methods perform knowledge- and context-based matching using heuristics and expert knowledge [214] (e.g., recognizing objects based on predefined structures or topological relations [208]). These approaches can be successful, especially if the solution is dedicated to a structure type or even a single object. However, to generalize well on different types, the solution would require a lot of hard-coded rules and expert-defined patterns. Data-driven methods use pattern recognition algorithms to label points based on training data. This category is dominated by deep neural networks.

Data-driven methods can be divided into image-, voxel-, and point-based [215], depending on neural networks' inputs. For image-based methods, points are projected onto a surface to generate input images. Two shortcomings of the method are limitations and loss in geometrical details presented by projections and unsatisfactory performance on complex structures [204]. SqueezeSeg [216,217] networks are image-based representatives. Voxel-based methods take voxelized occupancy maps as inputs [218]. This approach has been used widely, especially before the advent of point-based architectures. Similarly to 2D projections, voxelization also implies a loss of data resolution. VoxelNet [219], VV-net [220], and VolMap [221] are voxel-based representatives. Point-based methods take points as inputs. The points are inputted as properties; besides coordinates, networks may operate on, e.g., color, intensity, or normal. PointNet [222] is the first solution of the kind, with PointCNN [223], PointCONV [224], and KPConv [225] as other representatives. Recently, Transformer-based methods (e.g., TransPCNet [226]) have also gained popularity.

Generating digital models from point clouds is often the purpose of the automation pipeline since semantically rich digital models enable further analysis and simulations. These procedures are often referred to as Scan-to-BIM or as-built BIM generation. With this idea, the generated model's geometry matches the real object's geometry, including manufacturing imperfections and maintenance deterioration.

Generating digital models from point clouds is a multi-stage process, with segmentation typically involved as the first step. Next, the segments are transformed into model elements. This step may be done with different strategies, e.g., matching the whole segments to an objects library (if the library contains parametrized objects, also the parameters can be adjusted) or modeling the segment's geometry with geometry primitives. Then, the geometrical model is semantically enriched. Due to the diversity of elements and systems in structures, as well as the diversity of their applications, no method has yet been able to generalize on creating semantically-rich BIMs of different types of objects [227]. However, there are promising dedicated attempts for, e.g., steel structures [228], MEP scenes [229], or indoor environments [230].

Updating existing BIM models is another point clouds' use. The process is similar to Scan-to-BIM, but since the solution can be dedicated to one structure, the segmentation algorithm can also regard the constant location of elements, making the segmentation task easier. If the model is parametrized, the algorithm may extract the geometrical parameter values of the points segments and update the model accordingly. The geometrical accuracy of an updated model depends on the solution performance and the level of model elements' parametrization. Automated shape and pose updating of parametrized footings [214] is an example of the procedure.

Point clouds can also be utilized for other tasks. Examples of automated solutions include both increasing the quality of point clouds for further use (e.g., point clouds' shape completion [231]) and increasing the effectiveness of processes on different stages of projects: progress monitoring [232], structures' change detection [233], defects classifying [226], parameters extracting [234], measuring similarities between 3D models and point clouds [235], assessing the quality of prefabricated elements [236].

Point cloud use cases are abundant because they accurately represent the geometry of existing infrastructure. Point cloud techniques fulfill the pressing need for automation and increasing the effectiveness of AEC processes. Traditional techniques, such as manual measurements, visual inspections, or manual modeling, are reported to be more time-consuming and error-prone than modern substitutes [208,237,238]. Thus, automation has great potential to enhance AEC.

3.2.6.2 Point clouds in the digital twinning context

With the advent of 3D scanning and photogrammetry, point clouds are becoming basic sources of geometrical data for physical infrastructure. The machine-learning research gives a premise of automated point-cloud-based modeling and extraction of semantic features. Point clouds are perceived as a potential base for creating digital twins [17,34,239].

3.3 Notions towards digital twinning of bridges

This section presents notions (in the form of assumptions and suppositions) towards the digital twinning of bridges. The notions complement techniques (section 3.2) and characteristics (section 3.4) and together can be perceived as an extended definition to guide creating, utilizing, and interpreting digital twins of bridges.

Some of the notions may seem to interfere or even counteract. Indeed, although all of the guides are indispensable, digital twins should be suited to the object – and so should be the focus on the presented aspects. In some cases, the data-synchronization rate is crucial; in others – the effectiveness of autonomous decision-making. Ideally, all the guides should be addressed completely – but then implementation costs must be regarded. Fortunately, the modular character of digital twins allows for benefitting from the concept while adapting it to economic realities.

3.3.1 Digital twins allow business thriving in the future

As stated in the *Motivations* for this dissertation (section 1.1), the world is changing at an increasing pace, giving new opportunities but demanding new requirements. Industries need to adapt to thrive in the business sense.

The digital era facilitates collaboration. Industries have become more interdisciplinary, adapting foreign practices and tools. The trend leads also to multi-industrial standardization. Cross-sectoral written standards and versatile data exchange formats are already used. As industries become more interdisciplinary, models of different assets will collaborate more, similar to their physical counterparts. Cooperation between digital twins will be based on standardized data exchange interfaces. The collaboration will open opportunities for the adapted – and exclude technical laggards.

Modern techniques increase the effectiveness of whole industries but are also beneficial on a smaller scale. BIM is already proven to provide commercial advantages for contractors. It has been noticed by the Centre for Digital Built Britain and reflected in "The Gemini Principles" [82]. BIM is expected to provide "new markets, new services, new business models, new entrants". BIM, a technical innovation, is perceived as an export product, increasing the competitiveness of the national market. The same reasoning, on the interdisciplinary scale, applies to digital twinning. To thrive economically, industries must adapt to and benefit from technical innovations – especially if the innovations are worldwide trends, like digital twins.

3.3.2 Digital twins' frameworks should be object-specific

Digital twinning encompasses various industries and types of objects – such different as bridges and livers [63]. Different industries have different requirements and aims; different objects have different characteristics. Digital twins need to be based on general standards to enable cooperation. At the same time, the standards must be flexible enough to enable reflecting the specificity of the twinned object.

Ongoing digital twinning research is double-track: specific frameworks are still being defined (for, e.g., shop floors [240], air force [22], cities [241], or civil engineering facilities [80,81]), while use cases also explored. The frameworks attempt to adapt digital twinning principles for specific objects. The frameworks also discuss the strategy of implementation. The strategy is crucial because digital twins need to be industry applicable to develop as a concept (section 3.3.11). Therefore, digital twin frameworks should regard not only the specificity of the object but also its industrial environment – for example, the industry's practices and ability and willingness to implement new techniques. All industries, including civil engineering, must declare dedicated digital twinning frameworks based on general principles but suited to their specificity.

3.3.3 Ultra-high fidelity is not always required

The early digital twins' definitions envisioned "ultra-high fidelity" [20] simulations of the physical counterpart; the physical object was to be reflected at the "micro atomic level" [6]. The definitions were established when digital twin was just an idea, unable to be implemented due to technical limitations. More recent definitions are less strict, orienting on practice.

Undoubtedly, micro-atomic fidelity opens new scientific possibilities; for instance, the simulations regarding the micro-atomic structure of a material, instead of its averaged properties, may clarify unexplainable phenomena. But current modeling techniques are not prepared for such fidelity, nor simulation tools to consume such data. BIM projects in practice are in LOD 200-350, frequently starting on a lower level to gradually increase during the project. It would be hard – and unproductive – to demand LOD 500 if a lower level enables effective performance.

"A digital twin must represent a physical reality at a level of accuracy suited to its purpose [82]." Ultra-high fidelity is the ultimate aim for the future. However, the approach of gradually increasing the fidelity of digital twins, following the abilities of modeling and simulation tools, is practically beneficial today.

3.3.4 Data synchronization does not need to be perfect

Similarly to "ultra-high fidelity", the early digital twin visions demanded "real-time synchronization" [27]. Indeed, latency is crucial in some cases, but others may operate properly with a lower synchronization rate. In practice, the synchronization rate should be established regarding how dynamically the decisions based on in-time data must be made.

Consider examples of an autonomous car and a bridge. The car is continually affected by its environment: the temperature affects tire pressure and the windshield misting. The environment also drives the car's operating parameters: the weather and road conditions are factors for setting the driving speed. Most importantly, for the car, dynamic changes in the environment require dynamic decision-making. When a pedestrian unexpectedly enters the road, the one-second latency between registering the event and making a decision can result in an accident. The bridge is also affected by its environment, continually responding to the vehicle, wind, or temperature loads. However, typically, the bridge does not dynamically adjust its parameters to the dynamic events – decisions about bridges are typically not made in an in-time fashion. Thus, synchronization latency can be bigger.

Digital twins comprise different data. In civil engineering, point clouds became a primary source of reliable geometric data. Periodic actualization based on laser scanning or photogrammetry is beneficial for the actuality of digital twins' geometry. Acquiring the point clouds is still expensive and time-consuming, so it is unlikely to be a day-to-day task. But given the pace of geometry changes in civil engineering facilities, typically, monthly to annual actualization is sufficient. The source and type of data affect the synchronization rate.

Undoubtedly, the lower latency the better since it gives new opportunities; in-time decision-making, possible with in-time data, is beneficial for all kinds of objects. But, in some cases, it is not essential. The synchronization rate affects the digital twin's implementation costs. High upfront costs of in-time synchronization may result in neglecting the digital twinning concept. Therefore, starting with lower synchronization is acceptable from a practical perspective.

3.3.5 Digital twin should not neglect any data freely

After two notions softening the strict rules (*ultra-high fidelity is not always required,* and *data synchronization does not need to be perfect*), this one incline into thorough implementations.

Data is the raw material of the XXI century [50]. Indeed, as in the Bronze and Iron Ages natural resources allowed to create tools giving technical dominance, today the data gives advantages to its possessors. Data-driven techniques have vast applications in business and industrial practice. In science, they complement or displace traditional calculating theories. The data-driven algorithms need data to train and tune. But not only the amount and quality of data affect algorithms operation – also the diversity of data features. GPT-3, a natural language processing algorithm, is based on 175 billion input features [242,243]. Many AI practitioners are even stating that proper diversified data is more important for the effectiveness of the AI systems than the algorithms architectures [244] (i.e., a *weak* algorithm trained on *strong* data is preferred over a *strong* algorithm trained on *weak* data).

Existing calculating theories enable engineering practice, but it is naive to think that we already have understood the majority of phenomena. Perhaps new calculating theories based on data-driven techniques will disclose new, currently omitted patterns. Perhaps the new theories will be based on data that today seem useless.

Digital twinning prepares for the future not only in a business sense (section 3.3.1). It also prepares to solve engineering tasks in the future phases of the object lifecycles. Due to stored data and learned patterns, the tasks can be of a type that has not been initially thought of [5]. Therefore, data prepares for the expansibility of digital twins to match future industrial requirements.

Storing all data is practically not possible, but continually cheaper storage and new analyzing algorithms help in extensive data-collecting [244]. In the design of the DT systems, data should not be neglected freely – if it is possible to collect additional data with the designed methods, it should be done.

3.3.6 Digital twin provides data interfaces

The multitude of data is beneficial for data-driven algorithms but can be overwhelming for humans. The famous Buckminster Fuller's knowledge-doubling curve predicted the increasing pace of knowledge production, eventually leading to knowledge doubling every 12 hours [245,246]. Humans are not adapted to that pace. The "drowning in data" [247] syndrome affects people's effectiveness – the popularity of books like "Deep Work: Rules for Focused Success in a Distracted World" [248] or "Digital Minimalism: Choosing a Focused Life in a Noisy World" [249] reflects the need for aid in the data-rich environment.

Digital twins are data-rich models. The data is utilized by algorithms, but people also should have access to the insights. It should be provided with user interfaces.

User interfaces are essential in industrial practice. They are the linkers between users and complicated systems. User interfaces perform various roles: on commercial websites, they attract visitors to convert them into customers; in specialized systems, they provide access to operation details and enable management. In digital twins, the user interfaces should present historical, current, and predicted performance in an easily interpretable way. Due to data-processing techniques, the performance can be presented not as raw data but as more consumable information or knowledge. Based on the data, intelligent algorithms may even suggest a decision in the ultimate step of automated data interpretation.

Also, user interfaces allow segregating data access. Different participants are focused on different areas, e.g., not the same data should be provided for designers, construction site workers, managers, or users of bridges. It is important both for security and productivity.

3.3.7 Digital twin includes processes

Models are typically expected to store data about objects. The data-richness depends on the model type: CAD models store geometric data, while BIM models also include semantics. The semantics enable advanced analysis but are insufficient for a thorough description of a reflected object. For civil engineering facilities, knowledge about the expected response to loads or environmental factors is crucial for management. But just like the behavior of a person cannot be predicted based merely on appearance, the response of an object cannot be predicted based only on semantic data. To utilize its potential, a digital twin must include processes.

Digital twins include past processes by storing historical performance data. The data is processed by machine learning algorithms to distill patterns. The patterns represent the object's behavior and act as the “process knowledge” – based on that, the algorithms predict future performance.

The historical performance of a bridge is driven not only by explicit parameters (e.g., geometry, materials) but also by elusive factors (e.g., environment). The one-object-dedicated patterns implicitly comprise also this elusive data. Such a dedicated approach leads to more accurate predictions than those based on generalizations [16,61]. Eventually, the sensing techniques will allow to measure the today-elusive factors and implement them as explicit features of global machine learning algorithms. However, even then, the dedicated patterns will be beneficial to understand the responses of an individual object. Processes – both historical performance, represented by data, and future forecasts, represented by patterns – are indispensable components of digital twins.

3.3.8 Digital twin is intelligent, autonomous

Artificial intelligence is becoming a standard automation solution, also in civil engineering [250]. Digital twins are compatible with various machine learning algorithms. As a big data system, DT allows the training of supervised algorithms, the most common type of ML algorithms so far. As a simulation environment, DT also allows the development of unsupervised algorithms, e.g., reinforcement learning types, that are increasingly more popular and powerful, especially for autonomous control. However, from the utilization perspective, the type of algorithms is an implementation detail – the important are the algorithms-provided benefits.

With intelligent algorithms, the digital twin gains autonomy. The autonomy can be on various levels. The first is giving suggestions. A digital twin can analyze physical counterpart conditions and offer improving suggestions, e.g., a need for additional inspection or maintenance works. On this first level, the suggestions enhance human decision-making. On the next autonomy level, the suggestions are transformed into executable decisions. A digital twin of an autonomous car interprets the physical counterpart's data and forms decisions. The decisions are executed instantly and autonomously, with no human operator confirmation.

The digital twinning concept reveals an additional level of autonomy. A digital twin cooperates with other digital twins (Social Internet of Things [251,252], Web-of-things [253], or Social Internet of Vehicles [254] attempt to establish rules for this cooperation). With such partnership abilities, intelligent algorithms may consider the actual conditions of the collective, not only a single instance. This is beneficial for the whole group – if two cars spot a danger, but both choose the same escape path, the crash is inevitable. Therefore, digital twins communicate to make shared decisions. On this next level of autonomy, a digital twin, besides managing itself, influences others.

3.3.9 Digital twin is a system

A digital twin is often perceived as a single model. In civil engineering, this perception can emerge from BIM. BIM shares many features with digital twinning, so the digital twin is sometimes not quite correctly perceived as a more advanced version of the BIM model. But although the model is the center of the digital twinning concept, a digital twin is more than a model – it is a system. This statement is validated on various levels.

A digital twin is modular (section 3.4.5). The digital twin's modules are chosen regarding the specificity of the object and technical maturity. The modules may require specific submodels to perform analyses (e.g., a bridge structural response module may need a FEM model). The submodels derive from the central model and data storage, so they are based on concise, actual data. The submodels expand the digital twin's abilities and are its indispensable parts. The digital twin is a system of modules, and the central model and its derivates form a complementary system of models.

Digital twins operate on data beyond the typical model-stored type (e.g., big data from SHM sensors is likely to be stored in dedicated databases). Digital twins also include interfaces (section 3.3.6). Digital twins' submodels cooperate with themselves and the central model. All of these factors lead to extensive infrastructure orchestrating digital twins functioning. The infrastructure links the components and establishes the digital twin system.

Digital twins, with their interoperable abilities (section 3.4.4), form networks with other digital twins. The cooperation can be at parallel or subordinate levels. A digital twin can represent an object big as a planet [69] or small as a microchip. The complex digital twins comprise subordinate digital twins: a digital twin of a smart city may contain a bridge digital twin, which may contain digital twins of its sensors. The complexity of these connections transforms a digital twin into a system of systems.

3.3.10 Digital twins widen empirical science

The traditional mission of science is to explain *why* something happens; answering the *why* brings closer to understanding the world and its phenomena. We create and validate explainable theories, determine inputs, and, relying on the theories, calculate outputs. However, the data-driven approach shifts the emphasis. The most powerful artificial intelligence algorithms are not interpretable [255], obstructing following the calculations. Human operators focus on providing high-quality input data and then rely on the black-box-produced predictions or decisions. Not knowing *why* is somewhat uncomfortable, but the effectiveness of artificial intelligence is alluring, especially when generating the best predictions and decisions is crucial. In the case of bridges, for which detecting signals of catastrophic malfunctions is essential for user safety, focusing first on delivering reliable outputs and then discovering the explanation may be valuable.

Currently, data about bridges' behavior (e.g., load responses) is rarely collected in their normal state; monitoring is usually applied when a structure already shows signs of malfunctioning. Having insufficient data about the particular object, we cannot establish dedicated patterns – the object's state is assessed based on a generalization determined by other objects' data. But the *normality* of one object does not necessarily match the *normality* of the other. This statement is a basis for the digital twinning approach to personalized medicine: discovering individual patterns, varying with biophysical and lifestyle factors, allows determining optimal ranges of health indicators (e.g., blood pressure), which may differ from standard population-based values [60]. Bridges, even of similar parameters (geometry, materials), are affected by different conditions (environment, loads). Relying on generalizations may lead to omitting factors and dependencies affecting the particular object.

Digital twinning establishes unique big-data systems for every object. It enables a dual approach. Individualized intelligent algorithms, looking for dependencies and patterns of the particular object, enable analyzing and forecasting. Global algorithms, operating on multiple objects' data, are an additional layer of security. If the *normality* of one object strongly differs from the global pattern, it will be detected; it can be an indicator of the initial malfunction. Digital twinning enables learning objects empirically – observing how inputs affect outputs and understanding better the processes.

3.3.11 Digital twins must provide practical benefits

Siemens's report on building digital twins [77] states: “a pressing concern for possibly any company that starts on the journey to digitization could be the clear ability to show benefits and realize value from its investment in creating a digital twin.” The importance of transparent, and preferably instant, business benefits is highlighted by the industrial tycoon and must be addressed in establishing digital twinning frameworks.

Digital twins have been *born and raised* in the scientific environment, but only wide industrial applications can reveal their full potential. Digital twinning frameworks should address *academic* requirements (e.g., proper fidelity and synchronization rate) while also balance them with implementation costs. Indeed, there is no point in restrictive demands if they lead to academic-only applications and neglecting digital twinning in industrial practice.

Digital twinning frameworks must show clear premises of returns on implementation investment. The returns can be in various forms: limiting economic and social costs in the lifecycle; automating design, manufacturing, and maintenance; increasing the effectiveness of management. Also, the human factor should be considered: digital twinning will be demanded if it will enhance the day-to-day tasks of designers, construction site workers, engineers, and managers. Therefore, besides theories and standards, digital twinning research – both academic and industrial – must provide practical use cases, e.g., advanced simulation and modeling, automation of manufacturing, and effective management decision-making. This *bottom-up* approach fuels the practical development of the digital twinning concept.

3.3.12 It is the right time to establish digital twinning frameworks

Technical advances come from demands and opportunities. Digital twins have been conceptualized to handle the pressing requirements of digitization, but initially, due to technological limitations, the concept was impossible to implement. Today, however, technological advances open gates for digital twinning practice – industrial companies have already started implementations.

Civil engineering is still in the early stages of digital transformation. Nonetheless, BIM, one of the main civil engineering transformation trends, is already utilized. BIM implementation comes from both *top-down* and *bottom-up* approaches. In top-down, BIM is required by national laws (often for public investments, as in the United Kingdom, Norway, or Singapore [256]), and companies must adapt. In the bottom-up approach, BIM is not required, but its benefits are spotted by commercial companies and implemented to increase effectiveness. For instance, some design offices claim to use BIM for internal works, even though they hand projects in 2D drawings; just, for them, it is more effective to produce drawings from BIM models than to create them traditionally. This trend is promising. Potentially, technological possibilities coming from digital twinning may convince practitioners and fuel digital twins' implementations.

The need for the technological step – from BIM to digital twins – has been spotted by countries leading in civil engineering digitization. Centre for Digital Built Britain announced the idea of National Digital Twin [82], an ecosystem of digital twins connected to share data. The National Digital Twin comes from expected benefits: data-sharing in construction is expected to save 20-30% on industrial costs, holistic management strengthens the resilience of infrastructure, increasing innovation opens new markets. Centre for Digital Built Britain forecasts almost 100 million IoT connections in the infrastructure sectors by 2024; these devices will need to cooperate based on multi-industrial standards.

BIM demand for standardization resulted in ISO 19650 [150,257–261] or National BIM Standard [262]. It is not long before digital twins settle sufficiently for standardization. It is better for the whole civil engineering industry first to forge digital twinning frameworks in practice and then consciously involve in establishing the multi-industrial standards. The alternative, passive implementation of the standards created by others may result in ineffective and inconvenient solutions.

Civil engineering needs digital twinning to not stay a technical laggard. Digital twinning will open civil engineering to the best multi-industrial practices and the digital partnership of civil engineering facilities and other objects. This is especially important for bridges that in the smart cities will act as connectors of smart infrastructure. Therefore, it is the right time to research civil engineering digital twinning possibilities, form *our* practices and establish *our* frameworks.

3.4 Characteristics of digital twins of bridges

This section describes characteristics that complement notions (section 3.3) in defining the digital twinning framework for bridges. Actuality, intelligence and autonomy, as well as interaction, are the foundation of the digital twinning concept. Interaction (the ability to cooperate with other objects in the digital dimension) is enabled by interoperability (the compatibility with general principles and standards of data exchange). Modularity enables practical implementations; not all the functionalities of digital twins must be implemented from the very beginning. The expansibility of functionalities and scalability are the results of previous characteristics. Accessibility for users and other instances, but with the security of data, as well as uniqueness (e.g., of identifiers), are the indispensable factors for effective implementations.

Although a digital twin can reflect various instances (e.g., processes), the following characteristics are bridge-oriented, so the digital twin is described as the physical object counterpart.

3.4.1 Actual reflection: fidelity and synchronization

Among different views on digital twins, the core is indisputable: digital twin is the actual reflection of the *twinned* object. The actuality is achieved with fidelity and synchronization.

Fidelity is the ability to reflect the physical object at a sufficient level. In civil engineering, fidelity is most often linked with geometrical accuracy and semantic data completeness. BIM methodology describes model accuracy with a Level of Detail or Level of Development (LOD; Table 3-1). LOD of BIM models in practice depends on the expected utilization and project stage. Different stages have different fidelity demands: LOD100 is sufficient for preliminary quantities, but much higher LOD is expected for as-build documentation. Typically, the model starts with a lower LOD and increases with project development. This approach is practical, especially with current modeling techniques, which are dominantly manual – lower LOD at the initial design gives more flexibility in model adjustment. Automated modeling techniques, like visual programming or generators, enable efficient modeling in higher LOD from the beginning. Therefore, in practice, LOD depends also on technical maturity.

View on digital twins fidelity evolved. Early definitions, established when digital twinning was a futuristic idea, demanded ultra-high fidelity. More recent ones, however, orient on practice and tend to adjust the fidelity to technical needs. Digital twins should have the fidelity sufficient to serve current utilization and match technical abilities – especially the digital twins originated in the design stage can start with lower detailing. However, it is crucial to enable the increase of the fidelity level in the future. When a digital twin fidelity is to be increased, it should not be replaced by a new model, but the existing model should be enhanced with new details and data. In that, the culture of modeling is essential. For IFC-based digital twins, it is essential to keep the same GUIDs for elements during models' enhancement (this is not always the case in the modeling software).

Synchronization is another factor of the digital twin's actuality rate. Similar to fidelity, primary digital twinning visions demanded continuous in-time synchronization between physical and virtual counterparts. Perfect synchronization extends digital twins' potential and should be the ultimate goal. Nonetheless, in practice, the synchronization does not need to be in-time for successful digital twinning. The synchronization rate should be established regarding the object's context of operation, technical abilities, and data type.

Bridge digital twins store various types of data, e.g., geometry and load responses. The data can have different sources: geometry can be sourced from point clouds and load responses from SHM sensors. The data can have different sampling rates: laser scanning can be performed every few months, while load responses are sampled every second. The data can have varying updating labor: sensors outputs can be transferred automatically to a database, but modeling from point clouds is still dominantly manual [227] (although promising Scan-to-BIM and as-built generation attempts). Finally, the data have different usage: geometry may be used for visualization, while sensors data for simulations. All these factors lead to the conclusion that various data types stored in a digital twin should have various synchronization rates – not necessarily in-time.

Actual reflection is the core characteristic of digital twins, but the level of fidelity and synchronization can be adjusted to serve specific needs. Beginning with a lower actuality rate is beneficial from a practical perspective, but increasing the rate is crucial for extricating the DT potential. Firefighters using virtual and augmented reality tools in smoke-full buildings [263] is an example of extending the DT functions when it is actual – only the actual information lets the rescuers rely on digital models to find paths in an opaque environment. The same is with advanced, on-time simulations. When designing digital twins, it is important to declare the level of fidelity and synchronization rate and embed enablers to increase them. It is also crucial to contain only trustful data – it is better to create DT on lower fidelity than to declare high fidelity and model unreliable details.

3.4.2 Intelligence and autonomy

Digital twins are lifecycle management systems. The first digital twinning contribution to holistic management is unified, concise data storage. It is especially important in civil engineering, where data loss occurs at each lifecycle stage [134]. In digital twinning, however, the data is not merely stored – it is actively used to arouse and enhance intelligent algorithms. The intelligence leads to autonomy.

Big data and artificial intelligence are key blocks of digital twins [77]; intelligent algorithms are included in DT [71]. It differentiates digital twins from BIM models. In BIM, the algorithms are only one of the optional use cases. In digital twinning, however, intelligent algorithms are indispensable parts of digital twins' systems. Even if the algorithms are not operating at the origins of a digital twin, it must be designed to store data in a machine-learning input manner and include enablers for future algorithms' initiation.

Intelligent algorithms have numerous applications in civil engineering (section 3.2.4). They can automate processes and enhance or take over decision-making. Deferring decision-making to the digital twin makes it an autonomous system able to manage itself. The autonomy can be on different levels: from giving suggestions to a human operator to executing own decisions and affecting other digital twins (section 3.3.8).

With digital twinning interaction abilities, the algorithms can operate on different dimensions: analyze single digital twin data and affect the single instance or analyze data from multiple digital twins and affect single or multiple instances. An article describing the human body as an inspiration for digital twinning solutions [264] highlights another, more granular, dimension of digital twins' intelligence. In the body, the control is not always directly from the brain – some human body parts perform operations without conscious decisions (e.g., heart beating). The same phenomena can be introduced in digital twins' systems – some "local" operations (e.g., adjusting SHM sensors) can be performed without central systems authoring. This kind of diffused autonomy may effectively decrease the computation efforts of the central system.

Excluding humans in management decisions still encounters a social barrier – people are scared about potential harmful defects in algorithms operations [265,266]. The scare is non-trivial since the black-box machine learning solutions can perform unpredictably on edge cases. On the other hand, the algorithms are repeatable in dominant cases, being non-subjective and immune to factors influencing human decisions. The visual inspection rating of a bridge can vary significantly among different inspectors [136] – the inspectors' subjectiveness leaks into the rating. The weather is reported to affect retail purchasing decisions [267] and even stock returns [268] – it is likely to impact also the bridge inspectors' assessment. These elusive factors are imperceptible for algorithms, so their assessment and decisions are more unified and concise.

The level of a digital twin's autonomy should be based on the effectiveness of the algorithms. Higher levels of automated decision execution should be applied only after careful evaluation of the algorithms' performance, with a particular focus on edge cases. The performance of most types of machine learning algorithms depends on available data. With digital twinning systems, databases are continually enhanced with trustful examples, so the algorithms' effectiveness will increase; the algorithms self-evolve. Diversified datasets will enable establishing of both single-object and global algorithms, generalizing well on many objects. The algorithms may also be derived with transfer learning. In that manner, the algorithms do not have to be trained from scratch but only tuned with the object's data. It enables effective operation even in the initial phases of a digital twin existence when this instance dataset is small.

3.4.3 Interaction

The need for – and possibilities from – interaction should be a key motivator for the evolution into digital twinning. Interaction is the ability to communicate and cooperate. In digital twinning, interaction occurs on different levels. The interior level is the cooperation between a single digital twin's modules; the next is the cooperation between digital twins.

The interaction has three dimensions: physical-physical, virtual-virtual, or physical-virtual [34]. The physical-physical is already enabled – it is the natural interaction between physical objects (e.g., a car crossing a bridge interacts with the bridge and affects it). The physical-physical relation can also occur with the diffused autonomy (section 3.4.2) – some decisions can be made and executed by physical systems without interfering with digital counterparts. Virtual-virtual relation is the cooperation of digital modules in sharing data or performing collective simulations, but without affecting the physical counterpart. The bi-directional physical-virtual cooperation is the pinnacle of the digital twinning concept. The physical counterpart fuels the virtual counterpart with actual data; the virtual counterpart analyzes the data and steers the physical object.

The digital twinning interaction constitutes relations on different levels (Fig. 3-6). In the parallel relation, partnering systems are independent (like a car and a bridge); in the superior-subordinate relation, one model contains another. A bridge digital twin can contain, e.g., digital twins of measurement devices. But on the other hand, a bridge digital twin can be a part of a superior system, such as a digital twin of a smart city. The relations will be based on physical dependences but also negotiated by digital twins existing in networks [45].

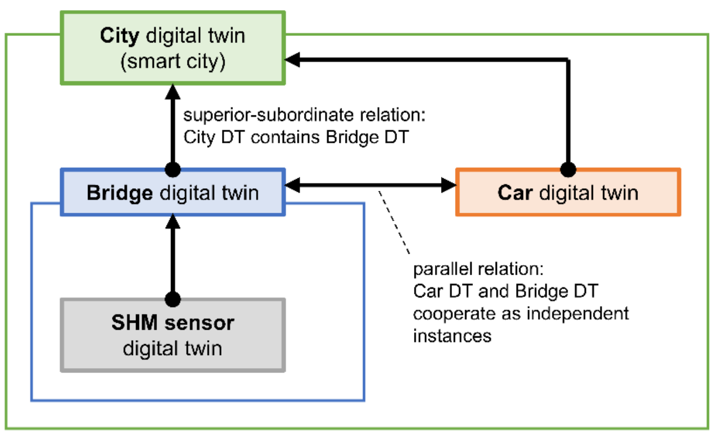


Fig. 3-6. Parallel, superior, and subordinate relations between digital twins

The ways of affecting the physical object by virtual counterpart depend on the nature of the object's operation and level of autonomy. [269] uses digital twins for real-time geometry assurance of produced objects and describes the virtual trimming concept. The newly produced components can have geometrical inaccuracies that traditionally are trimmed manually and iteratively. With a digital twin, however, the data about inaccuracies can be processed by the virtual model to simulate the iterative trimming. Then, the physical counterpart executes the trimming plan. The bi-directional link enables dedicated solutions based on actual data.

Bridges are much more static in their operation than production machines, so the virtual counterpart decisions are typically not so dynamic. Still, the virtual counterpart can perform expert analysis based on actual data to, e.g., select and plan maintenance operations. Depending on the level of autonomy and technical abilities, the suggestions may be presented to a human operator, so the virtual counterpart is not affecting the physical object directly and autonomously – yet. The future, when a bridge digital twin can order a drone-performed inspection or even an android-performed repair, seems to be more realistic with ongoing technological advances. Also, we may imagine an approaching car sending an introductory data package to the bridge; if, after virtual counterpart analyses, the car is suspected of causing damage to the structure, the bridge may obstruct the car's passage. These examples show that, in the digitally interconnected world, digital twins of also *static* bridges will perform extensive, autonomous, dynamic actions.

The interaction between multiple digital twins enables holistic management. It leads to global optimization regarding not only singular elements but bigger collective; optimization at the system-of-systems level, instead of individual systems, is more efficient [25] since it addresses global conditions. It guides the concept of smart cities, where bridges will be the links between intelligent infrastructure. But the interaction is beneficial already, in current technical conditions. The bridge operation is continually affected by its environment: environmental factors [270] (e.g., temperature, humidity, insolation), surrounding elements, and loads (e.g., type and periodicity of vehicles). Collecting data about that factors, enabled by interaction, and including them in analyses are crucial for interpreting bridge responses.

3.4.4 Interoperability

Interaction between digital twins is enabled by their interoperability. The interoperability comes from unified standards. The standards can have different forms: rules of implementations, data-exchange formats, programming interfaces. Compatibility with the standards opens the interaction for the compliant instance. It is the indirect reason for the need to introduce digital twinning to civil engineering: BIM, *our* digitization technique, is only civil engineering oriented; digital twinning, a multi-industrial trend, opens new collaboration opportunities.

Digital twinning is not yet standardized, and the need for standardization is declared as one of the main challenges for implementation [76,151,152]. Therefore, presumably, digital twinning will be a subject of similar processes as BIM, which eventually has been formally standardized. Nonetheless, the current non-standardization has positive sides: the ongoing formation stage gives a chance for industries to declare their own techniques and clarify needs that should be addressed in general standards.

Industry Foundation Classes [160] is the BIM standard, a core brick in the OpenBIM concept. In the proposed solution, IFC models are the base for civil engineering digital twinning. Dedicated digital twinning frameworks can be based on internal-industry practices but must comply with general standards. IFC, as a structured, open-source data schema, can be mapped to multi-industrial data exchange formats. Using IFC enables initiating the civil engineering digital twinning already while still being prepared for the future requirements of compliance.

Digital twinning standards must be general enough to enable multi-industrial cooperation but specific enough to enable successful applications of specific objects. Therefore, I propose a nested inheritance structure of standards. Listing 3-12 presents a scheme of the nested inheritance standardization concept with exemplary division. The outer standards, more general ones, would state common rules, definitions, and interfaces; delving deeper into the hierarchy, more dedicated standards would provide more specific guidelines.

Listing 3-12. Concept of a digital twinning standardization nested inheritance scheme

```
Digital twin
        DT of an object
                DT of a static object
                        DT of a civil engineering facility
                                DT of an infrastructure facility
                                        DT of a bridge
                DT of a dynamic object
                            […]
                                        DT of a car
        […]
```

3.4.5 Modularity

Digital twins are systems with complex interconnections. Modularity is the approach to constitute and effectively manage the connections. In the proposed framework, a digital twin is based on a central model linked with modules. The central model delegates tasks to the modules and comprises results. The modules can be infrastructural (i.e., enabling effective operation, e.g., database), provide user interfaces (e.g., visualization of geometry and performance), enable compatibility, or result from utilization (e.g., monitoring, simulation). The central model supervises modules based on inner interoperability, ensuring the modules operate on actual data.

The module set is suited to an object's specificity and digital twinning aims. The proposed bridge digital twin centrum is an IFC model. IFC-STEP models are not suited to store all types of data, so they should be aided by databases (although IFC comprises many rarely explored classes for storing various data, text-based storage is not always efficient). The database is an example of an infrastructural module. IFC viewers, data visualizers, and visual programming are user-interfaces modules. Interoperability modules enable communication and data exchange with other systems (e.g., a module mapping IFC schema to general standards). Finally, the utilization modules comprise, e.g., structural analysis, monitoring (SHM), maintenance planning, economic analysis. Fig. 3-7 shows an exemplary modular structure. For practical purposes, not all the modules must be launched at the digital twin initialization but can be integrated in later phases.

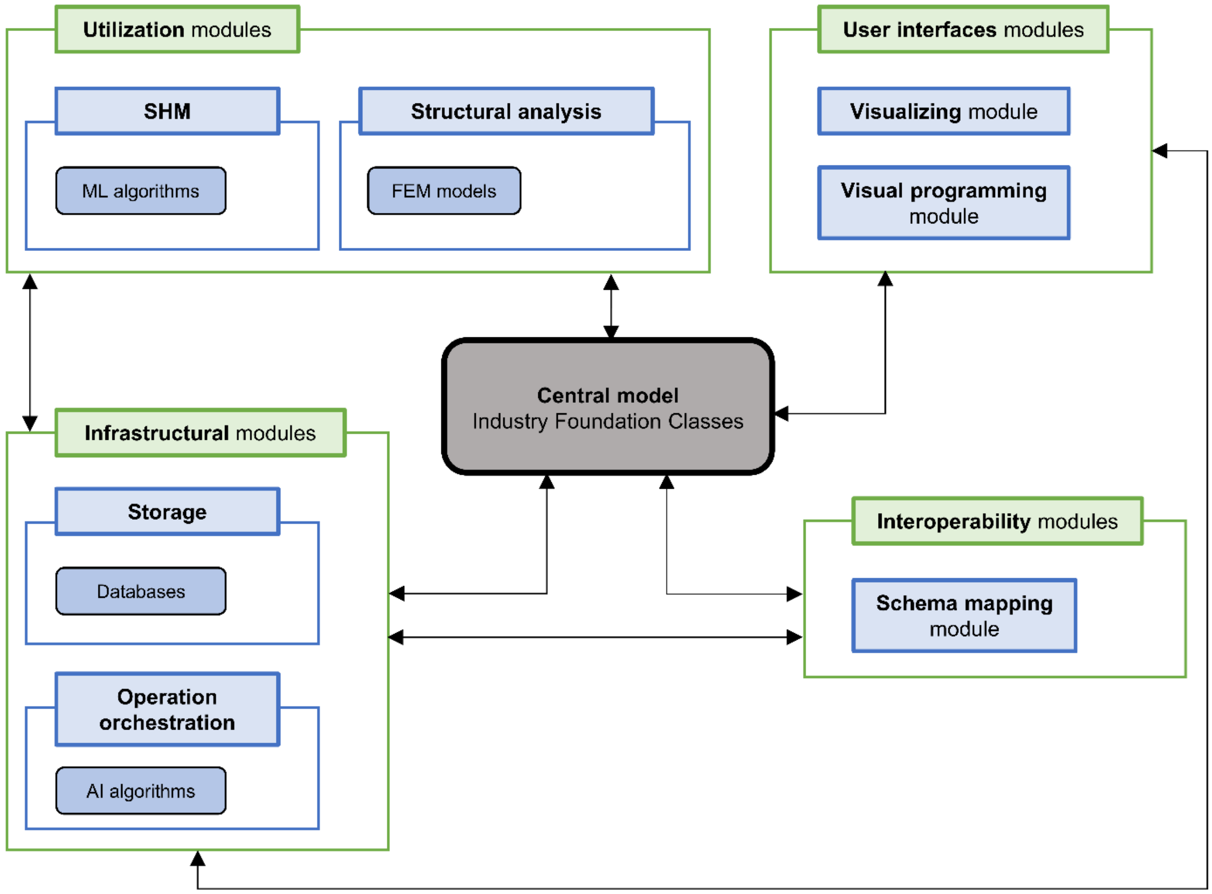


Fig. 3-7. Modular structure of a bridge digital twin

The listed modules are complex systems themselves, characterized by various granularities. For instance, a structural analysis module may comprise numerous FEM models constituting submodules. This complex network should be orchestrated by intelligent algorithms, driving communication between the central model, modules, and their submodules.

Modularity is effective in various cases. The famous SOLID object-oriented programming rules [271,272] begin with the Single Responsibility Principle. It advises structuring software code in a modular manner with specified, task-dedicated modules. Modularity exists also in the human body with its 11 distinct systems. Cardiovascular, digestive, endocrine, immune, integumentary, muscular nervous, reproductive, respiratory, skeletal, and urinary systems operate synchronously to enable human existence [264]. The modular approach can also be beneficial for managing complex digital twins' systems.

3.4.6 Expansibility and scalability

The interoperable and modular design of digital twins enables their expansibility: digital twins can be enhanced with new modules at each lifecycle phase. Therefore, digital twins prepare for the future in the technical sense, being open to integrating technological advances. New infrastructure modules (e.g., more efficient databases) may increase the effectiveness of the DT orchestration. New user interfaces may help in the human operations. New interoperability interfaces may enable consuming new data. New utilization interfaces may extend DT functions. New functions, especially those driven by intelligent algorithms, may be based on data stored in digital twins. Therefore, as discussed (section 3.3.5), storing different types of data, even those that initially seem useless, may be essential for future expansibility. The full potential of digital twinning is not yet known [101] – additional use cases will be continually revealed with time and usage. Digital twin design ensures benefitting from expected technical advances but also prepares for future demands initially not thought of [5].

The premise of expansibility requires a practical consideration of scalability. A digital twin must be technically ready to scale its operation. With superior-subordinate relations, a digital twin may comprise a system of instances – the ideas of smart cities or even digital twinning of Earth [69] show how complex this system can be. Also, digital twins are continually growing in data – especially since DT is not only a data container but also a data producer; outcomes of modules' operations are stored next to data from physical counterparts. To be prepared for these expansions, a digital twin must be scalable in terms of storage and computational power. It should be regarded in the design. A bridge digital twin, a relatively complex system, theoretically can start on local servers but eventually will need scalability currently offered by cloud providers. Initializing digital twins in the cloud should be considered to omit the practical burdens of migration.

3.4.7 Accessibility and security

Digital twin stores extensive data and its higher-order products: information, knowledge, intelligence [273]. These must be accessible to people and other cooperating digital twins. On the other hand, such value cannot be released unconsciously, so digital twins must have systems of data access management to ensure data security. Accessibility and security result from proper DT infrastructure architecture.

The central model format is essential in DT architecture. The proposed central model is Industry Foundation Classes. IFC is dedicated to AEC, so it can store industry-specific data (with the newest IFC 4.3, also infrastructure-specific). Nonetheless, it is not limited to the AEC-domain compatibility; as the open format, it can be mapped and linked to other schemas for interoperable data exchange. The open nature of the format also brings additional accessibility benefits. In the Open BIM idea, IFC is envisioned as the format that can be exported and imported by BIM software (currently, IFC import-export process still loses data, but industrial demands and software advances should fill this gap). Also, with open documentation, the data is ensured to be accessible through the facility lifecycle; given the bridges' lifecycle ranges (often exceeding 100 years), relying on vendor-dependent formats is uncareful.

Besides capabilities, placing IFC in the center of a digital twin also brings concerns. IFC, especially in its basic STEP file form, is ineffective in storing all data types. For instance: theoretically, IFC entities can store SHM data about historical performance; in practice, however, introducing all the historical data directly into IFC could make it an obese, unmanageable file. For that kind of data, databases are much more convenient – the digital twin system architecture should aid the central IFC with proper infrastructure modules. Another concern comes from perceiving IFC as static files. The perception potentially results from the practices on lower BIM adoption levels: IFC is rather a final handover model than a data exchange approach during design and construction. The cure to the dynamization of IFC data management comes with *webifying* the IFC. Activities like developing ifcOWL, an IFC version of the Web Ontology Language, give a premise for placing IFC as the center of an online system. Creating digital twins as online systems will ensure accessibility for people, physical counterparts' components (e.g., IoT devices), and other digital twins.

The benefits of accessibility should not obscure the need for security. Digital twin data should be accessible but not unconsciously and uncarefully. Digital twins must be equipped with data-sharing rules regarding people and other DT instances. Digital twin systems must also be developed on reliable cloud providers or dedicated online platforms. Data security is a key challenge of the digital transformation [37] – modern, reliable cybersecurity solutions are in demand.

3.4.8 Uniqueness

A digital twin originates with stating its system's architecture. It can be based on architecture frameworks preprepared with required modules (e.g., databases, orchestrating algorithms, parametrical geometry). At first, the system is *clear*, a digital twin does not differ from *newborns* of the same kind. However, in the lifecycle, the digital twin is enhanced with data specific to its physical counterpart. The data make the digital twin a unique instance.

The digital twin's uniqueness enables dedicated/personalized approaches. Big data of one object enables spotting specific patterns. It differs from relying on calculation generalizations, which may omit some factors – and, since physical phenomena are still wide to explore, we may not even know which crucial factors are neglected. The dedicated data-driven approach widens the understanding of the physical counterpart's processes.

The uniqueness of the digital twin is not only a data-resulting characteristic but also a practical requirement. Digital twins exist in the virtual environment of data exchange. In this communication network, it must be clear which instances send and receive the data. Digital twins must have their identifiers. The identifiers must be globally unique regarding also other domains, so the rules of identifiers attaching should be standardized. The identifiers system should be scalable to regard the potential explosion of digital twins' count.

Uniqueness is also required in the boundaries of a single digital twin: its modules, models, and models' elements. The need for unambiguous identification is reflected in the IFC specification; the majority of non-geometrical data-modeling entities (representing, e.g., elements, relations, processes) have Global Unique Identifiers (GUIDs). It enables reliable referring to the entities in external sources.

For reliable references, an element must stay with the same identifier, even if it has been modified. It is linked with modeling culture. Currently, IFC models are usually exported from commercial software models. It is essential to be careful about the way of applying changes to the model. For instance, if an element is deleted and inserted as new, its GUID will change. Sometimes it is desirable: the new element might be meant to reflect a new element, not just a modified version of the previous. But if the model element is logically still the same, it should be modified to stay with the same identifier. The issue of identifiers management has been spotted and applied in BIM software. Fig. 3-8 shows the ability to manage IFC GUIDs in Revit to keep consistency after modifications.

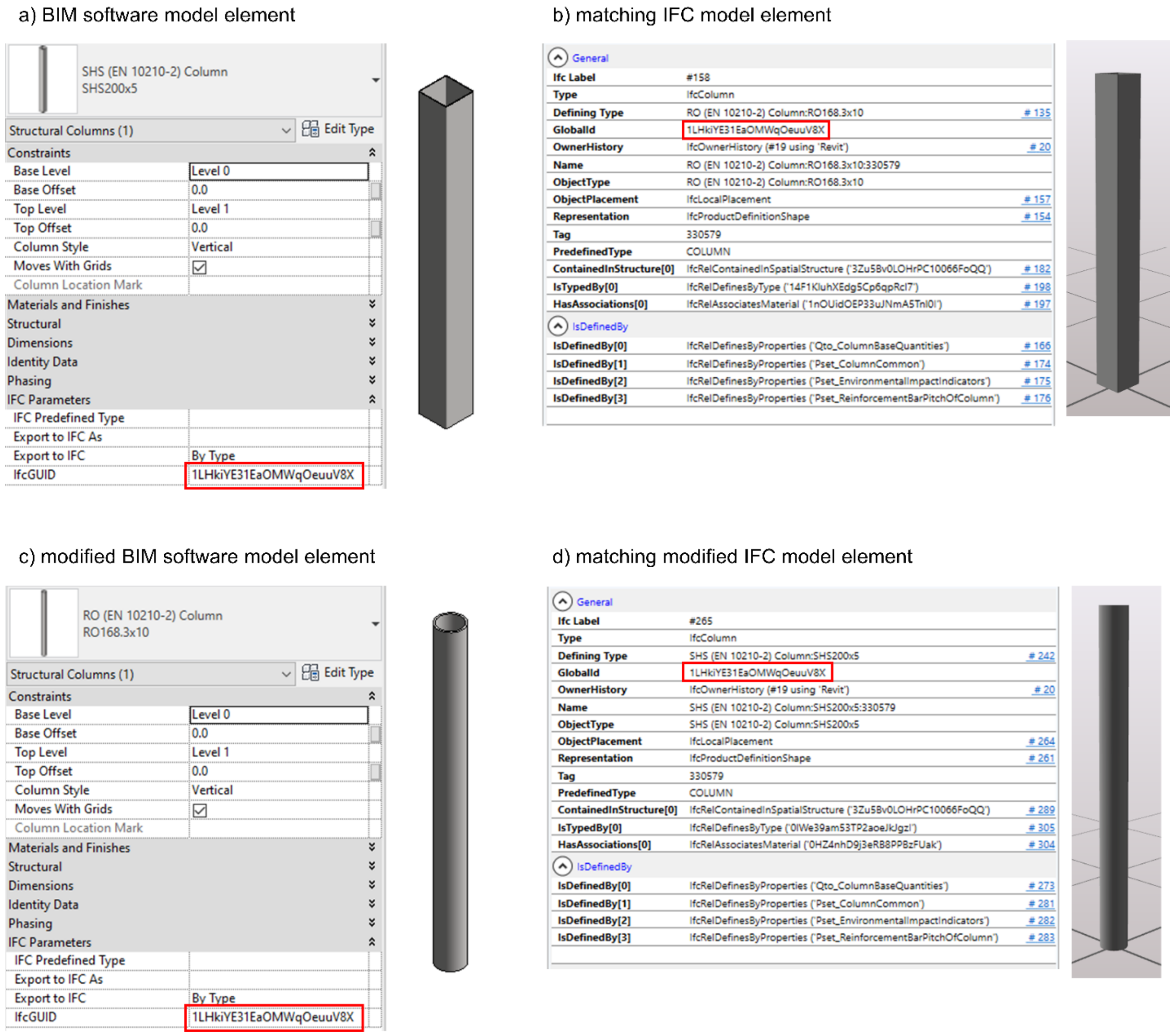


Fig. 3-8. Management of model elements identifiers in BIM software (Revit): a) BIM software model element; b) matching IFC model element; c) modified BIM software model element; d) matching modified IFC model element

3.5 Conclusions: the need for practical use cases

Bridges are complex, unique structures of logistic and strategic importance. The problems of aging infrastructure and the margin of civil engineering productivity motivate the introduction of modern technologies for data management, analysis, and automation. The need for transformations is demanded and expected by practitioners. Bridges, like the entire civil engineering, must adapt to fulfill new demands and benefit from technological advances. Digital twinning is the enabler of digital transformation.

Applicability is crucial in crafting solutions to be used not only in academia but also in industrial practice. Therefore, the here-proposed bridge digital twinning framework promotes the evolutionary approach. Civil engineering digital twinning should not be a forced revolution but a natural adoption of beneficial techniques. The techniques like BIM (with IFC), SHM, and AI, mixed and enhanced in an interoperable manner (with, e.g., visual programming and point clouds), can be a base for bridge digital twins. On the other hand, the framework must regard the general principles of the digital twinning concept. These forces are reflected in proposed notions towards digital twins of bridges and their characteristics. Bridge digital twin should be a virtual counterpart of the bridge in its entire lifecycle characterized by actuality, intelligence and autonomy, interaction, interoperability, modularity, expansibility and scalability, accessibility and security, and uniqueness. The proposed framework prepares bridge engineering to contribute to multi-industrial cooperation, thrive in the global market, and adapt to future tasks.

Digital twins, as an emerging concept, still requires extensive research. Stating the object-suited principles is the first step. However, for development, digital twinning must provide practical use cases giving clear benefits to designers, engineers, and managers of facilities.

This dissertation presents two solutions created by the author, utilizing techniques identified as components of civil engineering digital twinning. The solutions regard various lifecycle phases and provide benefits on the design and operational phases of the objects' lifecycle.

The first proposed solution is the optimization using visual programming and genetic algorithm [183]. Visual programming is used to link BIM modeling, FEM analysis, and optimization algorithm. The solution is a step towards utilizing digital twins' techniques in the design phase. The system is an effective geometrical optimization approach, assisting designers in their day-to-day tasks. Moreover, the integration of BIM and FEM models is crucial for creating a fully-functional digital twin of a bridge.

The second proposed technique is the generation of synthetic point clouds' datasets. Point clouds can be a base for modeling actual geometrical conditions in both operational and building phases. To automate the extraction of point clouds' information or even the generation of geometrical models, often machine learning algorithms are used. However, the available point clouds' datasets are not sufficient for the training of machine learning algorithms. The alternative is synthetic data. A scanning simulator created by the author allows for generating synthetic point clouds from BIM models. Acquiring the data is the first but indispensable step in developing machine learning solutions automating the utilization of point clouds, including systems to monitor geometrical changes in the digital twin models. The scheduled scanning of the physical object and then updating the model with trained algorithms will create a model-based history of geometrical changes. Monitoring, identifying, and comparing the geometrical changes can help identify structural malfunctions. As for bridges, monitoring the geometrical changes is crucial, especially for objects on the ground deformation areas (e.g., mining or post-seismic activities, tunneling). The model updating can therefore lead to expert systems alarming about hazards.

4 Optimization using visual programming and genetic algorithm

4.1 Introduction

The multitude of geometric parameters makes bridges a challenging task for designers. Traditional optimization is time-consuming and usually consists of an insufficient number of analyzed variants, leading to non-optimal solutions. Optimization techniques for bridges are the subject of extensive research. Many of the approaches use text-based programming languages. However, they are not easy to use by non-programmers. The alternative can be visual programming (VL), a more accessible form of developing algorithms. Visual programming is closely linked with BIM environments, where parametricism and quick modifications are essential. Visual programming scripts typically automate parametric geometry modeling but can also serve other engineering tasks. Merged with an optimization algorithm, they constitute an automated design process – generative design. In this approach, a user sets the constraints and ranges of selected parameters, and the algorithm optimally adjusts the values.

A generative design system based on visual programming can perform iterative, computationally-expensive tasks. However, as revealed within the literature study, its utility for bridges is still not sufficiently recognized, especially in the field of finite element analysis; not all civil-engineering visual programming environments provide proper FEM (finite element modeling) analysis tools.

To address this gap, the author proposes [183] a generative design geometry optimization process for bridges using Dynamo, an open-source visual programming language popular among civil engineers. BIM and FEM models have been integrated with an optimization algorithm in an automated manner. To enable FEM analysis, Dynamo has been enhanced with a FEM package created by the author. It empowers construction analysis entirely in the visual programming environment, which additionally automates BIM modeling. The created Dynamo script generates arch bridge models; then, they are computed with the FEM package methods. A linked genetic algorithm takes the analysis results as input. It steers the geometrical parameters adjustment and optimizes the geometry of the structure. The solution has been evaluated on a complex reference construction.

The use case presented in this chapter shows exemplary benefits of digital twins' techniques in the design phase. The algorithms of automated optimization can assist designers in day-to-day tasks. And the integration of BIM and FEM models is crucial for creating fully-functional digital twins of civil engineering facilities.

4.1.1 Optimization techniques for bridges

Structural and architectural optimization has been the subject of advanced studies for over four decades [274]. They incorporate various mathematical and programming methods, particularly iterative procedures based on a gradient or probabilistic approaches. The probabilistic aspect is essential for global search algorithms, e.g., particle swarm optimization [275], simulated annealing [276], swarm intelligence, or genetic algorithm [277].

In bridge engineering, extensive studies have been carried out on volume, weight, and cost minimization. [278] developed a genetic algorithm system in a Visual Basic environment to minimize the total cost of bridges' prestressed concrete beams by varying their shape, prestressing, and arrangement. [279] used a genetic algorithm to optimize high-strength steel girders: their weights, cost, and CO_2 emission. [280] presented a two-step material cost optimization by the example of simply supported composite girders, analyzing five algorithms using Matlab.

Optimization techniques' advantages – universality and robustness for non-convex and multi-modal structures – are especially beneficial for complex systems. Numerous studies concern the optimization of truss, cable-stayed, suspension, and arch bridges.

Optimization of cable-stayed bridges mainly involves cable forces distribution and cost minimization [281], often with the automated generation of numerical models for constraints' impact evaluation. [282] optimized cable-stayed bridges using 2D models and also compared the influence of nonlinear and linear analysis. [283] provide a detailed description of 3D model implementation for similar structures. [284] analyzed the optimal quantity of cables for the La Coruña cable-stayed bridge, using Matlab to generate FEM models. [285] presented an extensive summary of the optimal cable-stayed bridge design for various span lengths with a nonlinear FEM analysis and genetic algorithms. The cables' cross-section areas have been reduced with the B-spline curve distribution function [286]. A similar approach coupled with a simulated annealing method was proposed in [287]. Also, a micro-genetic algorithm [288], Support Vector Machine (SVM) [289], and artificial neural networks [290] increased the efficiency of the cable-stayed bridges' optimization. Machine learning techniques reduced the FEM model calls to the number required to create the training set; further optimization is based on iterative input-output generalization, bypassing time-consuming FEM computations.

In terms of arch bridges', [291] analyzed a steel truss arch bridge with a hybrid genetic algorithm: a combination of genetic algorithm and finite element analysis. [292] and [293] studied the reliability-based optimization of arch bridges with probabilistic and deterministic constraints using 2D finite element models. [294] developed a network arch bridges' optimization interface based on evaluating geometric parameters with a global evolutionary operation (EVOP) algorithm. [295] examined one of the swarm intelligence algorithms, the bee colony, in arch bridges' design. [296] analyzed a concrete, open-spandrel arch bridge using the Simultaneous Perturbation Stochastic Approximation algorithm (SPSA), Ansys, and Matlab software. [297] proposed the optimal design of stone arch bridges implementing limit state plastic analysis based on a hinged mechanism for the collapse mode. [298] assessed arch bridges' aesthetics and compatibility with landscape merging a psycho-vector concept and genetic algorithm-based shape optimization. A genetic algorithm was also used to reduce arch bridge concrete deck material [299], optimize arch bridges made of high-performance steel [300], evaluate the optimum hangers configuration in network arch bridges [301], analyze soil-structure interactions [302], and perform structural identification [303].

4.1.2 Genetic algorithms

A genetic algorithm is a nature-inspired global search heuristic method used to generate optimal solutions. It imitates biological reproduction and evolution processes executing repetitive genetic operators: selection, crossover, and mutation. Genetic algorithms have been modified in various ways to handle binary-coded, real, and discrete variables [304]. They have been applied to unconstrained and constrained optimization problems [305], including combinatorial, linear, and nonlinear functions described by one or more competitive objectives (single- and multi-objective optimization) [306].

At first, an initial population with a fixed size is randomized. Each element stores a set of design variables' values related to a given cost or fitness function. The function is essential for selecting: the elements with the best fitness/cost function value are most likely to be passed to the parent set. Since the genetic algorithm concept is based on a probabilistic approach, candidates with worse function values can still be passed to the parent set, but it is less likely to occur. The selection can be based on different methods, e.g., the so-called tournament or roulette wheel.

A crossover operator matches the parent set's individuals to produce offsprings. For binary-coded variables, a simple one-point crossover is typically used. For real-coded variables, the simulated binary crossover (SBX) operation [307] flips two binary strings between two parents. A mutation operator is applied to a limited number of recombined individuals; it alters variable values to diversify new generations and minimize the chances of convergence to local extrema [308]. The selection, crossover, and mutation are repeated to meet the termination condition: the maximum number of generations or time limit is reached, or the evaluated function values are stagnant.

The optimization problem formulation requires a definition of design variables, constraints, and objective function. The constraints form a boundary between feasible and infeasible solution regions. The objective function can be a cost function, which is to be minimized by algorithm, or a fitness function, which is to be maximized. They can be interconverted in most cases.

4.2 Methods

4.2.1 FEM and visual programming: deploying a package

Visual programming software allows third-party additions that extend their functionalities. The supplementary packages of methods provide advanced geometry creation, BIM models-2D drawings linkage, Excel spreadsheets manipulation, and specific simulations.

Dynamo is an open-source visual programming language for parameterized geometry modeling. It offers solutions for exporting created geometries to third-party computational applications (e.g., Autodesk Robot Structural Analysis Professional, StruSoft Fem-Design). However, it lacks methods to perform finite element method analyses without external software. To address this gap, a finite element method package for bar models' static analyses has been developed as part of the proposed solution. It enables performing the simulation entirely in the Dynamo environment establishing an effective and consistent design method.

The package consists of so-called zero-touch nodes. It functions as a C# programming language dynamic-link library (DLL). Its development requires knowledge of text-based programming, but once created and imported to Dynamo, its methods can be ordinarily used in visual scripts. Zero-touch development provides access to the software API (Application Programming Interface) and the ability to use external text-programming language libraries. The open-source BriefFiniteElement.NET library [309] has been used in the FEM package development. BriefFiniteElement.NET enables the static and linear analysis of solids and structures using the finite element method. The library development is ongoing, so new features and capabilities are continually added.

The Dynamo FEM package introduces a set of classes whose methods operate as nodes in the visual programming script. The *AnNode* and *AnBar* classes represent analytical elements of the computational model. Their constructing methods take basic Dynamo geometries, *Points* and *Lines,* as inputs. The geometries constitute analytical elements' locations. They are also characterized by several features. Cross-section (represented by *AnSection* class) and material (*AnMaterial*) are fundamental parameters of analytical bars. The *AnSection* instance is declared by the cross-section area and central moments of inertia, while the *AnMaterial* by Young's modulus, Poisson's ratio, and material weight. The *IsTruss* parameter sets bars' connection: fixed or released. The supports are added as the optional parameter of *AnNodes*.

The package allows applying concentrated forces and uniform loads. Concentrated forces, represented by *AnNodalLoad* class, can be added as *AnNodes* optional parameter, while uniform loads (*AnBarLoad)* as an optional parameter of *AnBars*. The loads form load cases with *AnLoadCase* class. Self-weight can be automatically applied to all model elements. The *SolveModel* method of the *AnModel* class runs the constructed model analysis. Its results can be obtained with *AnResults* class methods, of which *ForceAtParameter* is the most versatile. It returns the internal forces at a point on the bar, determined with the 0 to 1 parameter value, describing the relative location of the point on the bar length. Fig. 4-1 shows a fragment of the script constructed with the FEM package nodes. The script processes both input and output values with no units, a common visual programming approach. To ensure the analysis's correctness, the user must declare a consistent unit set and convert the values accordingly.

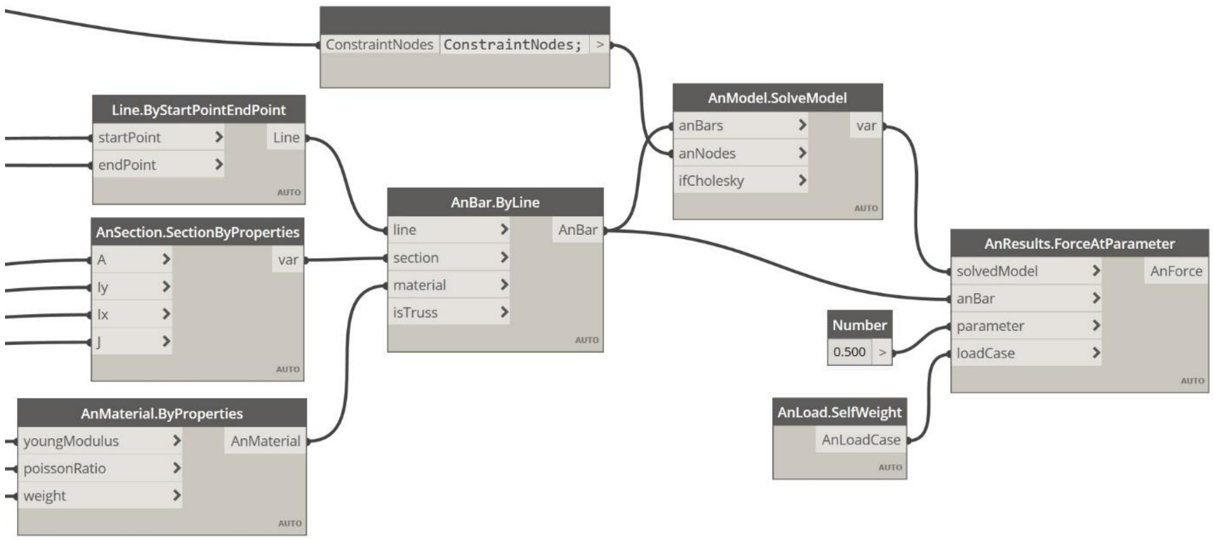


Fig. 4-1. Fragment of Dynamo script with Dynamo FEM package methods

Visual programming environments have a specific set of operation rules. Typically, methods nodes store script data as separated outputs. The Dynamo FEM package additionally holds instances of analytical classes in its operating memory. This approach is unusual for visual programming but enables advanced data manipulation, such as implemented element duplication prevention. The script creates new *AnNodes* and *AnBar* instances only if geometrically-identical elements are not already present in the model.

The execution order of the methods is another visual programming issue. It is clear for simple, one-branched scripts, but they are rarely used. For multi-branched scripts, the order of execution may vary, but nodes whose outputs are the inputs of other nodes are always executed first. Therefore, to ensure that the script obtains analysis results after the model is computed, *AnResults* nodes use the output of *AnModel.SolveModel* as one of its inputs. Together with the elements duplication prevention system, this approach provides faultless computational model generation and operation.

4.2.2 Generating models

To automate arch bridge design, a Dynamo visual programming script has been developed. The basic Dynamo methods create a geometrical model and the developed FEM package analyze it. In this way, geometry modeling and FEM analysis are performed in one environment without additional software.

Fig. 4-2 shows the flowchart of the script operation. A script user inserts values of the input geometrical parameters. Based on them, the script generates a geometrical model, further transformed into an analytical model. Both models contribute to output parameters calculation: *MaterialCost* results from the geometry model and *use* indices from computational analysis. Based on the outputs, the user evaluates the construction solution.

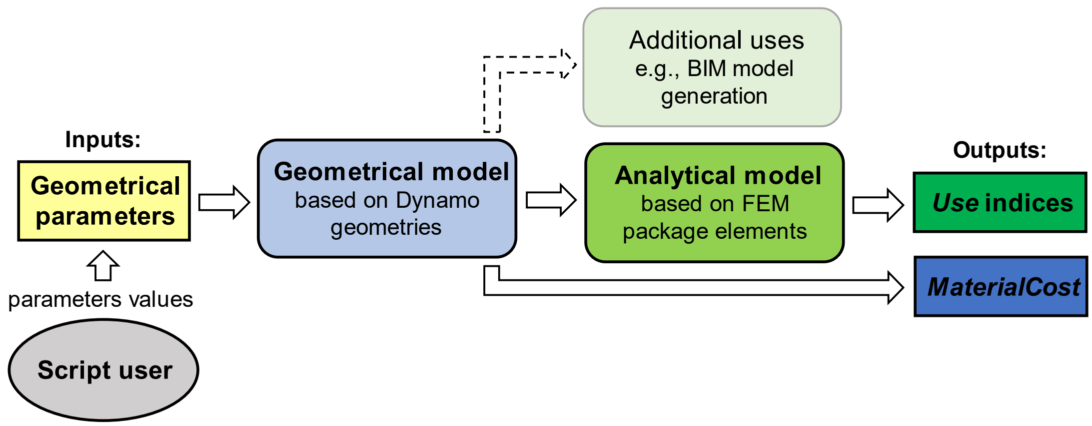


Fig. 4-2. Script operation flowchart

The script generates a bar model of the arch bridge composed of steel elements (Fig. 4-3). According to their function, the bars form groups: arches, girders, crossbars, hangers, and braces. Crossbars are further divided into main (connected to hangers), additional (between the main ones, optional), and external. The number of the groups' elements varies depending on the construction geometry. Each group has a fixed type of cross-section: I-beam, rectangular box, tube, or solid circle. Their dimensions are adjustable with parameters. Also, the material properties can be set separately for each group.

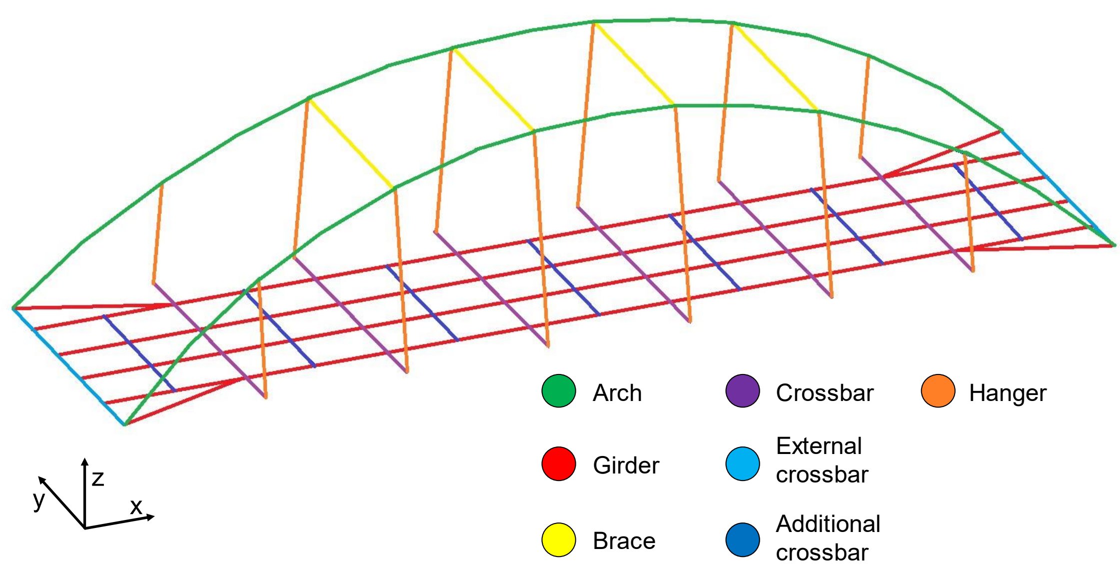


Fig. 4-3. Bar model and its bars groups

The script provides four hangers' layout systems: vertical, radial, oblique, and network. The model support system consists of fixed, free sliding, and guided bearings (Fig. 4-4).

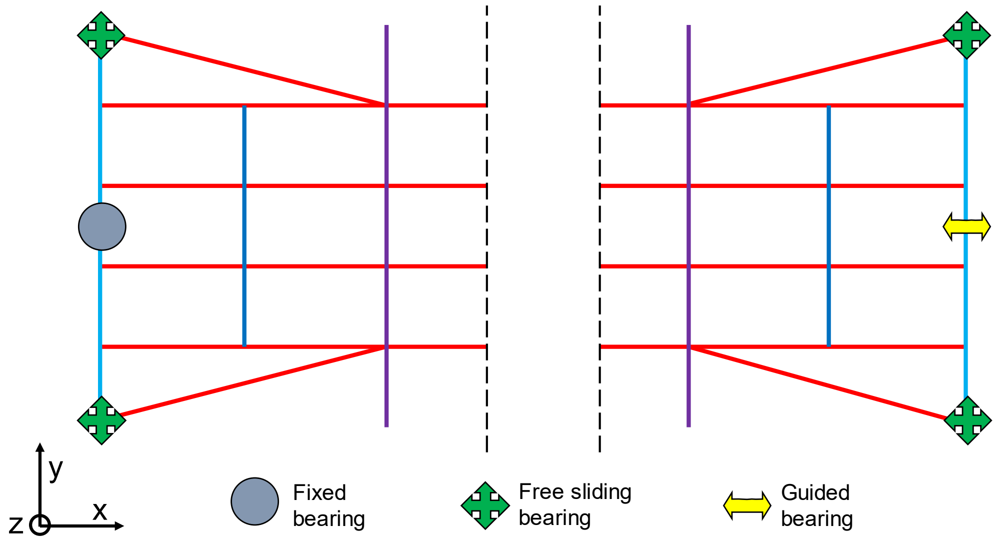
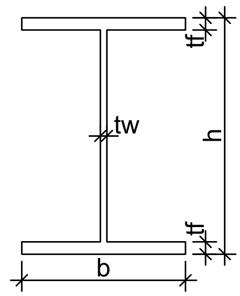
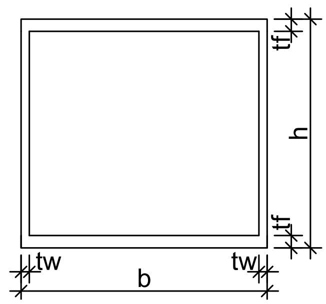
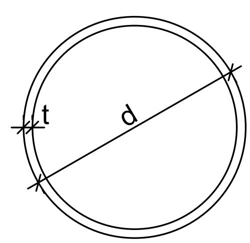
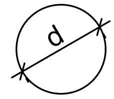


Fig. 4-4. Support system schema

The model geometry is based on input parameters (Table 4-1). They control the construction form and cross-sections' dimensions. All the geometrical parameters are number types, including the one related to the hanger system selection: the system is defined by an integer associated with a type. The parameter values are inserted with special slider blocks; they set the values' ranges and steps. The script operates on an established unit set: global parameters in meters and cross-sections' dimensions in millimeters.

Table 4-1. Geometrical input parameters

Parameter	*Unit*	*Description*
Length	m	Axial length of the construction.
Width	m	Axial width of the construction, measured between arches' bars axes on the girders' bars axes level.
Use width	m	Summed width of lanes and pedestrian walkways.
Center girders spacing	m	Spacing between two girder bars.
External girders offset	m	Spacing between the first main and external girder.
Braces number	-	Number of braces between arches.
Arch rise	m	Axial rise of the arches.
Arch inclination	°	Lateral inclination of the arches. A positive value indicates an inclination towards the construction interior.
Hangers system type	-	Adopted hanger system type. 0 – Vertical, 1 – Radial, 2 – Oblique, 3 – Network.

Hangers number	-	Number of hangers fixed to one of the arches. Depending on the selected hanger system type, the number of hangers establishes the number of crossbars fixed to the hangers.
Additional crossbars number	-	Number of additional crossbars between each pair of main crossbars. The additional crossbars are not fixed to the hangers; their function is to increase the cross stiffness.
h_girder, b_girder, tf_girder, tw_girder; h_crossbar, b_crossbar, tf_crossbar, tw_crossbar; h_crossbarAdd, b_crossbarAdd, tf_crossbarAdd, tw_crossbarAdd	mm	Dimensions of the I-beam cross-sections of girders/crossbars. 
h_arch, b_arch, tf_arch, tw_arch; h_ crossbarExt, b_ crossbarExt, tf_ crossbarExt, tw_ crossbarExt;	mm	Dimensions of the rectangular box cross-sections of arches/external crossbars. 
d_brace, t_brace	mm	Dimensions of the tubular cross-section of braces. 
d_hanger	mm	Diameter of the hanger cross-section. 

After a user adjusts the parameters, the script generates the bridge bar geometry. It consists of Dynamo geometrical classes such as *Lines*, *Curves*, and *Arcs* (Fig. 4-6a). The model is the basis for further operations. Based on the bar geometries, the Dynamo FEM package creates *AnNodes* and *AnBars*, which form a computational model. It includes self-weight and three live load cases. The live loads are the uniform bar loads applied to girders (Fig. 4-5). Their magnitude *q* results from the *Use width* parameter, number of girders (4, constant in the script), and magnitude of the planar live load q_a. The planar live load amount is adjusted depending on the type of the analyzed bridge.

$$q = Use\ width \cdot q_a\ /\ Girders\ number \tag{4-1}$$

where:

q_a – planar live load [kn/m^2]

Girders number = 4

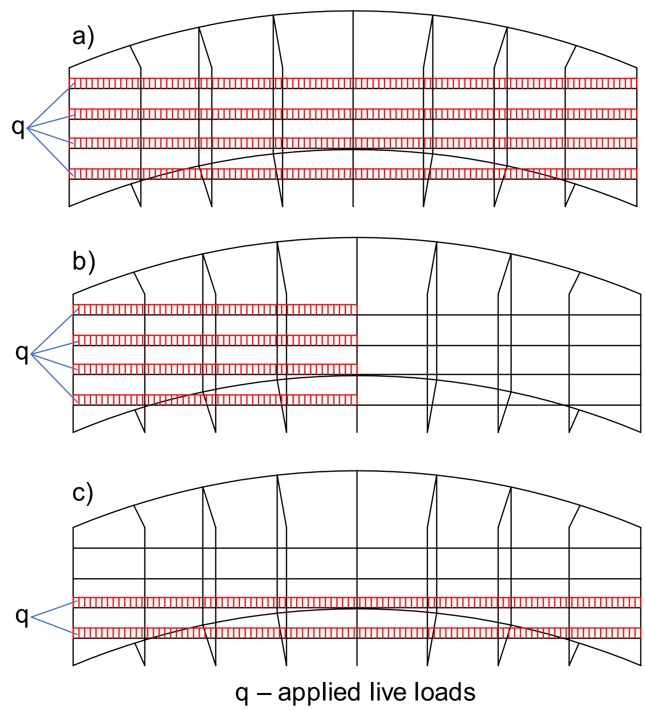


Fig. 4-5. The schema of live load cases applied to model

The computation analysis results are the structure's internal forces: the bending moments (M_y, M_z), torque moment (M_x), shear forces (F_y, F_z), and axial force (F_x). The values are obtained for each model bar in 10 points: two external (bar's start and end) and eight uniformly distributed internal points, separately for each load case. They are combined to get eight cases of extreme dominant force with its corresponding forces for each bar. The extreme dominant is the maximum value of a particular internal force in the bar; the corresponding forces are the other internal forces occurring at the point of extreme dominant. The cases are sole self-weight with dominant minimum F_x, maximum F_x, minimum M_y, maximum M_y, and self-weight with extreme live load (one of the three load cases) with dominant minimum F_x, maximum F_x, minimum M_y, maximum M_y. The values are multiplied by combination factors [310]: $\gamma_{G,sup}$ = 1.35 (self-weight increasing the extreme value), $\gamma_{G,inf}$ = 1.00 (self-weight decreasing the extreme value), $\gamma_{Q,sup}$ = 1.50 (live load increasing the extreme value), $\gamma_{Q,inf}$ = 0 (live load decreasing the extreme value).

$$F_{min} = \gamma_G \cdot G + \gamma_Q \cdot \min(Q_1, Q_2, Q_3)\,;\ \ \gamma_G = \begin{cases} \gamma_{G,inf}, & G \geq 0 \\ \gamma_{G,sup}, & G < 0 \end{cases};\ \gamma_Q = \begin{cases} \gamma_{Q,inf}, & Q \geq 0 \\ \gamma_{Q,sup}, & Q < 0 \end{cases} \quad (4\text{-}2)$$

$$F_{max} = \gamma_G \cdot G + \gamma_Q \cdot \max(Q_1, Q_2, Q_3)\,;\ \ \gamma_G = \begin{cases} \gamma_{d,inf}, & G \leq 0 \\ \gamma_{G,sup}, & G > 0 \end{cases};\ \gamma_Q = \begin{cases} \gamma_{Q,inf}, & Q \leq 0 \\ \gamma_{Q,sup}, & Q > 0 \end{cases} \quad (4\text{-}3)$$

where:

$\gamma_{G,sup}$, $\gamma_{G,inf}$ – combination factors for constant loads, supremum and infinium

$\gamma_{Q,sup}$, $\gamma_{Q,inf}$ – combination factors for live loads, supremum and infinium

G – effect of self-weight

Q_1, Q_2, Q_3 – effects of subsequent live-load cases

For each set of combined extreme dominant-corresponding forces of each bar, the normal stress values are determined. The longitudinal stress and capacity estimations include the plate buckling effects. In the case of arch bars, global buckling effects are also included [311–313].

For arches, longitudinal stress is calculated as follows:

$$\sigma_i = \frac{F_{x,i}}{\chi \cdot A_i} + \frac{M_{y,i} \cdot x_i}{I_{y,i}} + \frac{M_{z,i} \cdot y_i}{I_{z,i}} \quad (4\text{-}4)$$

For other bars, longitudinal stress is calculated as follows:

$$\sigma_i = \frac{F_{x,i}}{A_i} + \frac{M_{y,i} \cdot x_i}{I_{y,i}} + \frac{M_{z,i} \cdot y_i}{I_{z,i}} \quad (4\text{-}5)$$

In equations (4-4) and (4-5):

σ_i – normal stress at the point of the bar

$F_{x,i}$ – axial force at the point of the bar

$M_{y,i}$, $M_{z,i}$ – bending moments at the point of the bar

A_i – area of the bar cross-section

x_i, y_i – perpendicular distances to the neutral axes of the bar cross-section

$I_{y,i}$, $I_{z,i}$ – second moment areas of the neutral axes of the bar cross-section

χ – global buckling factor for arch bars

The cross-section properties are reduced with the effective cross-section method [311]. In the case of arches, ψ = 1.00 compressive stress distribution ratio along the internal parts of the cross-section was assumed to result in buckling factor k_σ = 4 [311].

$$h_{eff,arch} = \rho \cdot h_{arch} \quad (4\text{-}6)$$

$$b_{eff,arch} = \rho \cdot b_{arch} \quad (4\text{-}7)$$

where:

$h_{eff,arch}$ – effective height of the arch cross-section

$b_{eff,arch}$ – effective width of the arch cross-section

ρ – reduction factor; $\rho = \min\left[1.0;\ \frac{\bar{\lambda}_p - 0.055\cdot(3+\psi)}{\bar{\lambda}_p{}^2}\right] = \min\left[1.0;\ \frac{\bar{\lambda}_p - 0.055\cdot(3+1.00)}{\bar{\lambda}_p{}^2}\right] = \min\left(1.0;\ \frac{\bar{\lambda}_p - 0.220}{\bar{\lambda}_p{}^2}\right)$ (4-8)

$\bar{\lambda}_p$ – relative plate slenderness; $\bar{\lambda}_p = \frac{b/t}{28.4\cdot\varepsilon\cdot\sqrt{k_\sigma}}$

ε – partial factor, depending on the applied material

In-plane and out-of-plane buckling effects for arches have been assumed. Equations (4-9) and (4-10) calculate critical buckling forces *Ncr* [313].

$$N_{cr,y} = \left(\frac{\pi}{\beta\cdot s}\right)^2 \cdot EI_{y,i} \quad (4\text{-}9)$$

$$N_{cr,z} = \left(\frac{\pi}{\beta\cdot 0.5\cdot L}\right)^2 \cdot EI_{z,i} \quad (4\text{-}10)$$

where:

β – buckling length factor depending on arch rise to span length ratio in accordance with [313] table D.6

s – half-length of the arch for in-plane buckling

L – projection length of the arch, reduced by two in equation (8) due to transversal bracing

$EI_{y,i}$ – in-plane flexural stiffness of the arch

$EI_{z,i}$ – out-of-plane flexural stiffness of the arch

The global buckling factor χ used in equation (4-4) is calculated [312]:

$$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - \bar{\lambda}^2}} \quad (4\text{-}11)$$

where: $\Phi = 0.5 \cdot \left[1 + \alpha \cdot \left(\bar{\lambda} - 0.2\right) + \bar{\lambda}^2\right]$ (4-12)

$$\bar{\lambda} = \sqrt{\frac{A_i\cdot f_y}{N_{cr}}} = \sqrt{\frac{A_i\cdot f_y}{\min\left(N_{cr,y};\, N_{cr,z}\right)}} \quad (4\text{-}13)$$

α – imperfection factor, for hollow cross-sections: $\alpha = 0.49$.

The maximum value of stress in the bar group $\sigma_{BarGroup}$ is used to calculate its use index $Use_{BarGroup}$. The maximum value of the groups' use indices determines the construction use index *Use*.

$$Use_{BarGroup} = \sigma_{BarGroup}\ /\ f_y \quad (4\text{-}14)$$

$$Use = max(Use_{Arch}, Use_{Girder}, Use_{CrossBar}, Use_{ExternalCrossBar}, Use_{AdditionalCrossBar}, Use_{Brace}, Use_{Hanger}) \quad (4\text{-}15)$$

The script also calculates the *MaterialCost* parameter.

$$MaterialCost = \sum_{BarGroup} k_{material,BarGroup} \cdot V_{BarGroup} \quad (4\text{-}16)$$

where:

$k_{material,BarGroup}$ – cost factor of the material applied for the bar group

$V_{BarGroup}$ – material volume of all the bars in the bar group

The output *use* indices and *MaterialCost* parameters are the basis of model evaluation and further optimization.

The script's utility is not limited to computational analysis; the parametrical model is the basis for automating several engineering tasks. Besides expanding its functionalities with dedicated packages, Dynamo collaborates with external software. It is especially associated with Autodesk Revit. With that, the Dynamo basic geometries model can be transformed into an object-based BIM model (Fig. 4-6). The Dynamo-BIM software linkage provides a solution for the one-script design and BIM model generation.

a) Basic programming environment geometries

b) BIM model

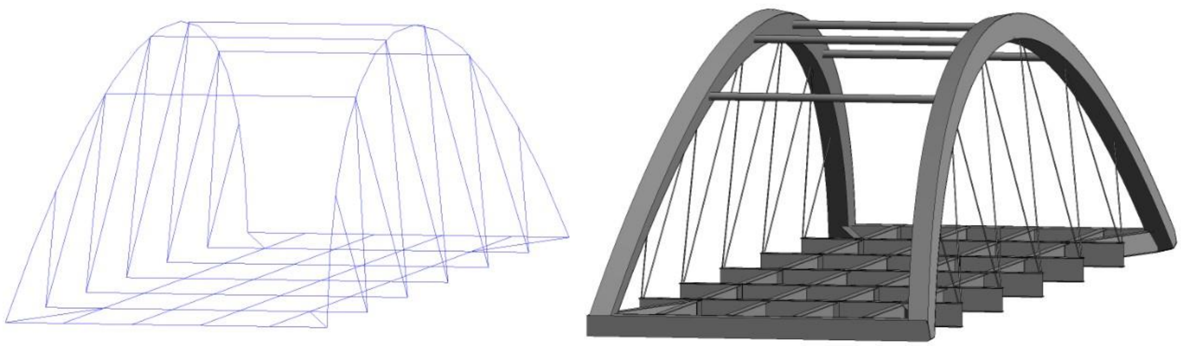


Fig. 4-6. Visualizations of a) basic programming environment geometries and b) generated BIM Model

4.2.3 Generative design optimization

Generative design is an approach to automate design processes [314]. It uses parametrical and relational modeling, similar to the computational design, but is enriched by the optimization algorithm. The computational design allows the rapid generation of multiple solutions by scripts. It improves the process but requires a user to adjust the script parameters' values manually; therefore, it is classified as a passive approach. With active generative design, the user sets ranges of parameters and the optimization objective, and the parameters' values are optimally determined by the optimization algorithm.

Here, a genetic algorithm-based generative design system has been implemented to automate the optimization of the arch bridge geometry. The genetic algorithm is linked with the geometry modeling script using the Dynamo Refinery tool. Fig. 4-7 shows the roles of a user and the genetic algorithm in the optimization system. The user sets parameters' ranges and determines the optimization objective. The genetic algorithm iteratively drives the visual programming script, adjusting input parameters' values depending on the optimization objective score.

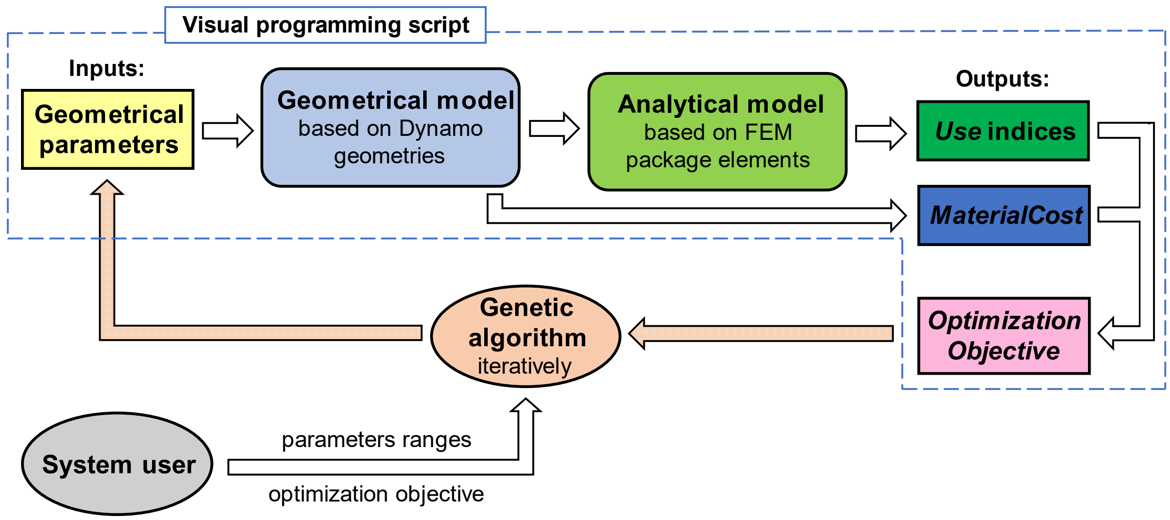


Fig. 4-7. Generative design optimization system operation flowchart

The genetic algorithm optimization objective function evaluates the subsequent iterations' solutions. A typical construction optimization goal has been assumed: minimizing the material cost while assuring the load-bearing capacity of the structure. The OptimizationObjective function is conditional. It equals MaterialCost if the static strength conditions are fulfilled; if not, the value is overstated with the Use parameter.

$$OptimizationObjective = \begin{cases} MaterialCost, \ Use \leq 1 \\ MaterialCost + 10^6 \cdot Use, \ Use > 1 \end{cases} \tag{4-17}$$

The optimization system task is to provide a solution with the lowest OptimizationObjective score. With the applied function, the genetic algorithm discards the solutions that do not fulfill strength conditions, promoting those that bear the applied loads and consume less material.

4.3 Optimization process and results

To evaluate the proposed system, reference construction optimization has been performed. A bridge used as a reference is a viaduct over the S3 expressway in Central-Western Poland. It is a steel through-tied arch bridge with a composite deck and inclined arch ribs. It has vertical hangers fixed to crossbars. Its load test was performed in 2011, and the results are presented in [60].

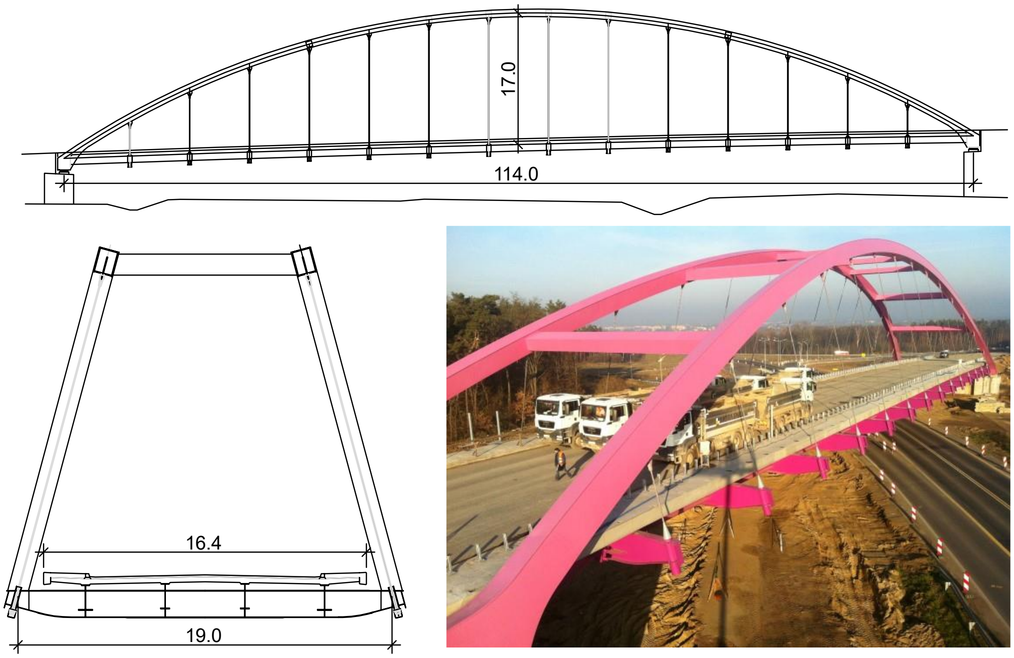


Fig. 4-8. Reference construction selected for the optimization process

4.3.1 Reference construction analysis

The reference construction analysis has preceded the optimization. Table 4-2 lists the values of input geometry parameters. The materials were S355 (f_y=355 MPa, ρ=79 kN/m^3, k=1) for all the elements except hangers (f_y=1570 MPa, ρ=79 kN/m^3, k=3). The applied cost factors *k* were used in the *MaterialCost* calculations.

Table 4-2. Input geometry parameters of reference construction

Parameter	*Value*
Length, m	114.0
Width, m	19.0
Use width, m	16.2
Center girders spacing, m	4.0
External girders spacing, m	3.4
Arch rise, m	17.0
Arch inclination, °	-15
Hangers system type	0 (Vertical)
Hangers number	14
Braces number	6
h_arch, mm	1320
b_arch, mm	950

tf_arch, mm	40
tw_arch, mm	30
h_girder, mm	1270
b_girder, mm	470
tf_girder, mm	35
tw_girder, mm	16
h_crossbar, mm	1400
b_crossbar, mm	650
tf_crossbar, mm	30
tw_crossbar, mm	16
h_crossbarExt, mm	1200
b_crossbarExt, mm	1500
tf_crossbarExt, mm	20
tw_crossbarExt, mm	30
d_brace, mm	450
t_brace, mm	20
d_hanger, mm	58

Self-weight and three live load cases were applied to the structure (Fig. 4-5). The basic planar live load q_a value that caused almost full use of the reference construction has been applied. This approach complicates the optimization task: the simple uniform reduction of all the cross-sections' dimensions would have led to the unfulfillment of strength requirements. The analysis determined q_a = 4 kN/m^2, which resulted in *Use* = 0,988 for the reference construction (Table 4-3). The same set of loads has been applied in all subsequent analyses and optimization tasks.

Table 4-3 presents output values of the reference construction analysis with the applied loads. The maximum group index, *Girder use,* was almost equal to 1, meaning nearly full use of the structure. However, the other values indicate a potential for optimization with a more uniform stress distribution. The *MaterialCost* of the structure was 87.24; this value had been used as a reference to evaluate the system efficiency in subsequent optimization cases.

Table 4-3. Output parameters of reference construction calculations

Output parameter	*Value*
MaterialCost	87.24
Arch use	0.863
Girder use	0.988
Crossbar use	0.740

External Crossbar use	0.741
Brace use	0.160
Hanger use	0.225

4.3.2 Optimization assumptions

Four optimization tasks have been conducted. They differed in parameters selected to be adjusted by the genetic algorithm: cross-section dimensions, static scheme parameters, both cross-section dimensions and static scheme parameters, both cross-section dimensions and static scheme parameters with a material declaration. Fig. 4-9 presents the selection of variables for the subsequent optimization tasks. Cross-sections-only optimization did not include hangers system type differentiation, so one optimization cycle was performed. For the rest cases, a separate cycle for each hanger system was performed; thus, each hangers' layout optimization performance was analyzed.

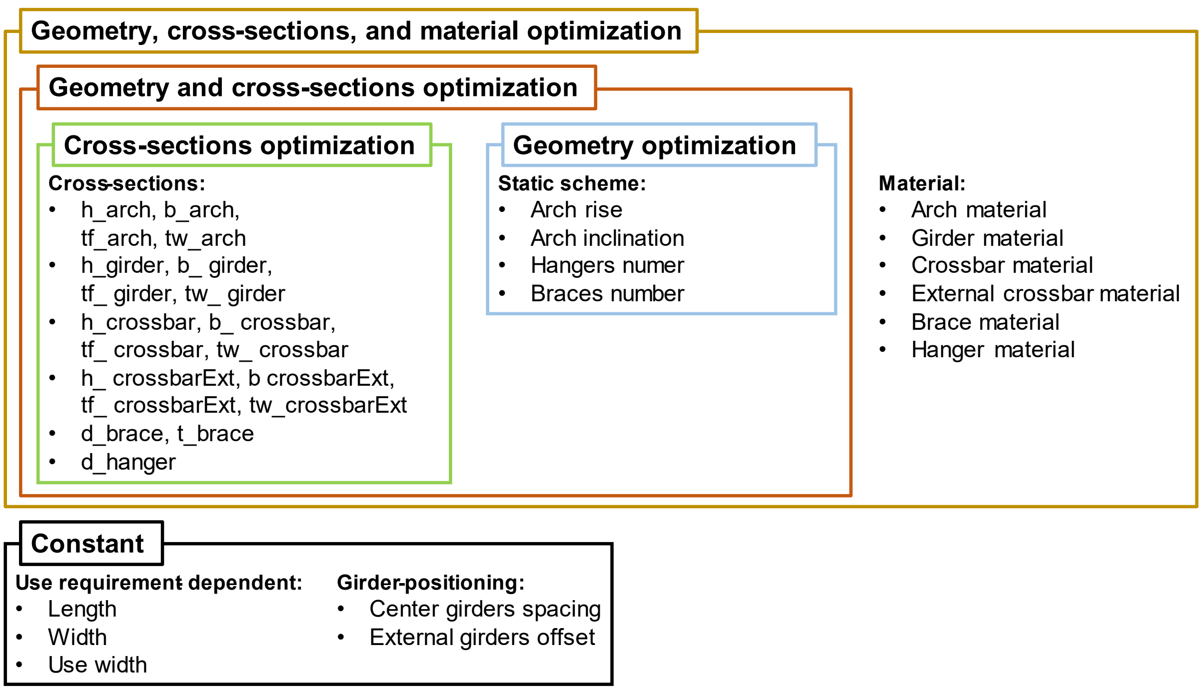


Fig. 4-9. Optimization cases and their adjustable parameters

Along with parameters non-adjustable for the specific case, two groups were constant for each optimization task: with values depending on the bridge use requirements and girder-positioning ones. Their values matched the reference construction (Table 4-2).

Optimization requires setting the range and step values for adjustable parameters.

Table 4-4 lists ranges for cross-sections' dimensions; additional adjustable parameters were *Arch inclination* (-15 to 15, step 1) and *Brace number* (4 to 6, step 2).

Table 4-4. Ranges and steps of cross-sections' parameters

Parameter	*Minimum, mm*	*Maximum, mm*	*Step, mm*
h_arch	800	2000	100
b_arch	500	1500	100
tf_arch	20	50	10
tw_arch	10	50	10
h_girder	800	2000	100
b_girder	200	1000	100
tf_girder	10	50	10
tw_girder	12	20	4
h_crossbar	800	2000	100
b_crossbar	300	1500	100
tf_crossbar	10	50	10
tw_crossbar	12	20	4
h_crossbarExt	800	2000	100
b_crossbarExt	800	2000	100
tf_crossbarExt	10	50	10
tw_crossbarExt	10	50	10
d_brace	200	1000	100
t_brace	12	32	4
d_hanger	30	100	10

To specify ranges and steps for *Arch rise* and *Hangers number*, one hundred examples for each layout were analyzed. The examples differed only in these two examined parameters; the *Arch inclination* and *Braces number* were set according to the reference construction (Table 4-2), and the cross-section dimensions to the construction optimized with the cross-section-only case (Fig. 4-12, described in 4.3.3).

The analysis confirmed the possibility of modeling a structure that consumes less material but still fulfills the assumed conditions (Fig. 4-10). Based on the *Use* and *MaterialCost* output parameters, the examples formed three groups: non-fulfilling the static strength conditions (*Use* > 1); fulfilling the conditions, but non-optimized (*Use* ≤ 1, MaterialCost ≥ 87.24); fulfilling the conditions and optimized (*Use* ≤ 1, MaterialCost < 87.24). The value of 87.24 refers to the *MaterialCost* of the reference construction (Table 4-3). Based on the results, the ranges of *Arch rise* and *Hangers number* for each hangers' system (Table 4-5) were specified. They were set to include the spectrum of optimized instances. The determined *Arch rise* step=0.1m allowed the optimization algorithm to adjust the value precisely. The *Hangers number* steps resulted from the hangers layout: 1 for vertical and radial, 2 for oblique and network.

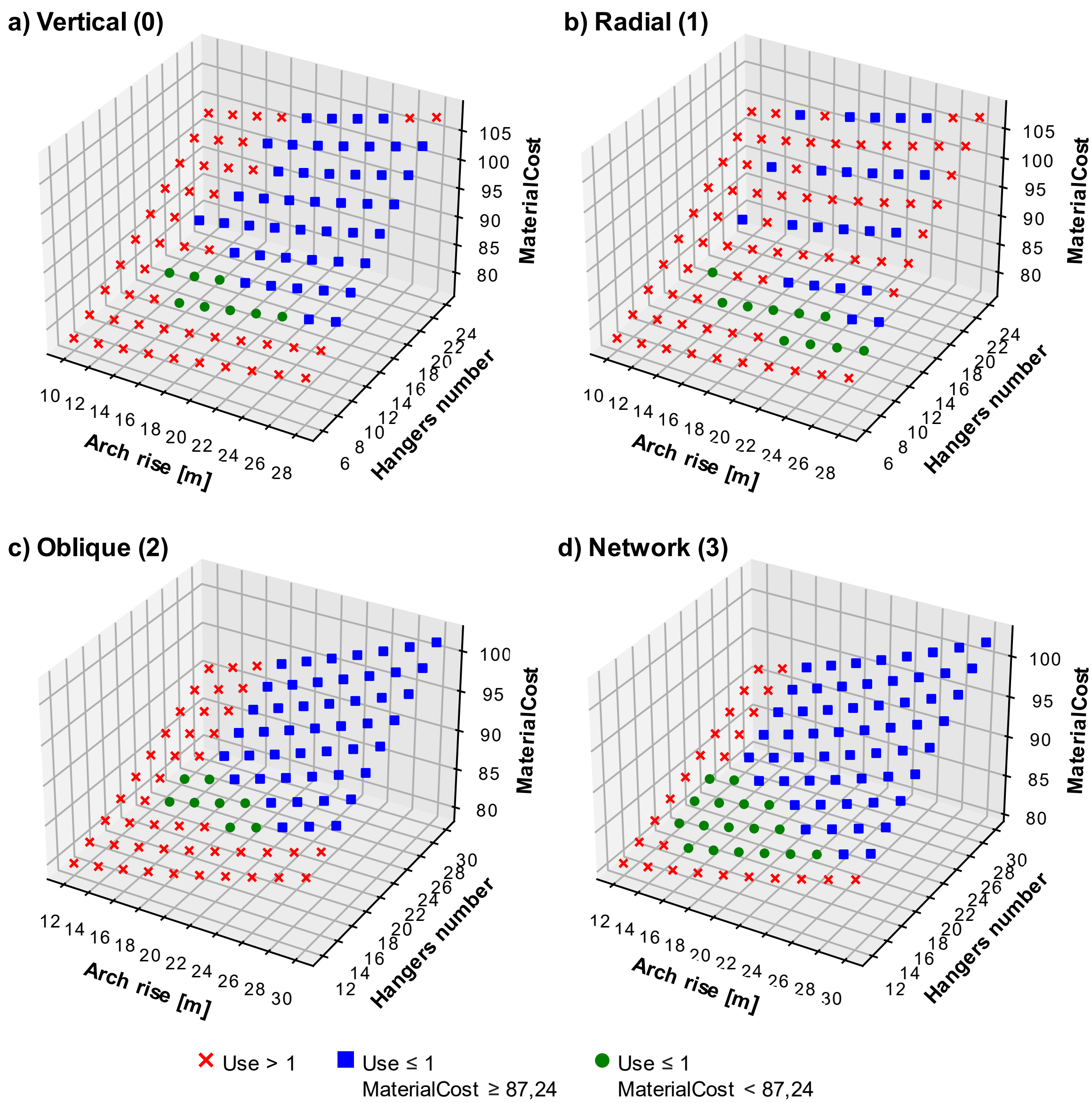


Fig. 4-10. Results of the ranges setting analysis

Table 4-5. Ranges and steps of *Arch rise* and *Hangers number* parameters

Hanger system	*Arch rise, m*	*Hangers number*
Vertical (0)	14.0 to 26.0; step 0.1	6 to 12; step 1
Radial (1)	14.0 to 26.0; step 0.1	6 to 12; step 1
Oblique (2)	14.0 to 26.0; step 0.1	14 to 20; step 2
Network (3)	14.0 to 26.0; step 0.1	12 to 18; step 2

In the three optimization cases (cross-section dimensions, static scheme parameters, both cross-section dimensions and static scheme parameters), the materials' properties were constant and were applied according to the reference construction: S355 (f_y=355 MPa, ρ=79 kN/m^3) for all the elements except hangers (f_y=1570 MPa, ρ=79 kN/m³). In the fourth case, the algorithm selected each non-hangers bar group material: S235 (f_y=235 MPa, ρ=79 kN/m³), S275 (f_y=275 MPa, ρ=79 kN/m³), or S355. The following material cost factors $k_{material}$ were applied: 0.7 for S235, 0.8 for S275, 1 for S355, and 3 for the hangers' steel. The values are relative, but they may also match the actual material market prices; thus, the calculated *MaterialCost* would estimate the actual cost.

The genetic algorithm requires adjustment of its governing parameters: number of generations and population size in each generation. The number of generations differed depending on the complexity of each optimization task: 70 for cross-section dimensions, 30 for static scheme parameters, 100 for both cross-section dimensions and static scheme parameters, 150 for both cross-section dimensions and static scheme parameters with material determination. The population size was 20 for all the cases.

4.3.3 Cross-sections optimization

During the cross-sections optimization, the system adjusted the cross-sections' dimensions of all the bar groups **(**Fig. 4-9), which is a typical optimization task. The construction static scheme parameters and materials properties remained constant through analysis; their values matched the reference construction.

The optimization algorithm operated on 19 parameters.

Table 4-4 presents their ranges and step values. The optimization was conducted through 70 generations of a population of 20.

Fig. 4-11 presents the outputs of the best examples (with the lowest *MaterialCost)* of subsequent generations.

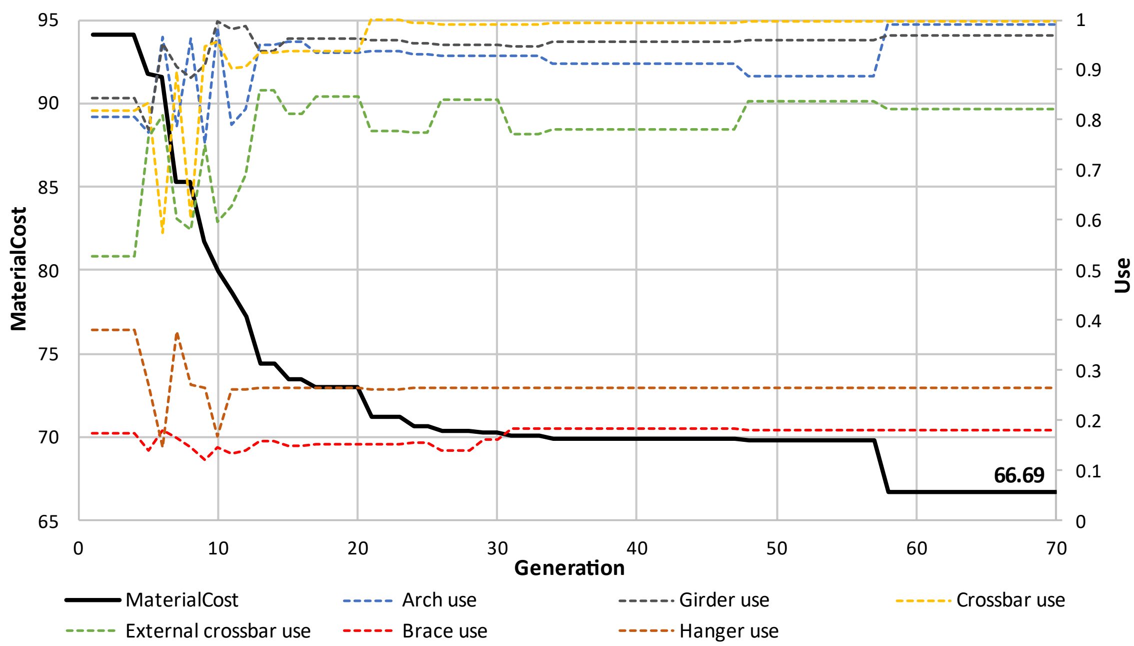


Fig. 4-11. Cross-sections optimization results

The initial part of the process was characterized by a rapid decrease of *MaterialCost* and high variability of *use* indices. The wide set of adjustable parameters allowed for many combinations; therefore, almost every generation introduced better solutions. The further part was steadier, but it also included separate generations of significant improvement.

The algorithm optimized the structure by increasing and balancing the use of bar groups. In the final solution, the *use* indices of the three main groups (arches, girders, crossbars) were close to 1, which indicated almost full use. However, the introduced constraints prevented the optimization of braces and hangers. Each generation provided at least one example that fulfilled strength conditions – none of the *use* indices exceeded 1.

Fig. 4-12 shows the comparison of the reference and optimized constructions' cross-sections. It indicates that the operation of the system is more sophisticated than the basic uniform reduction of cross-sections' areas; girders and crossbars were strengthened, which resulted in significant savings for arches.

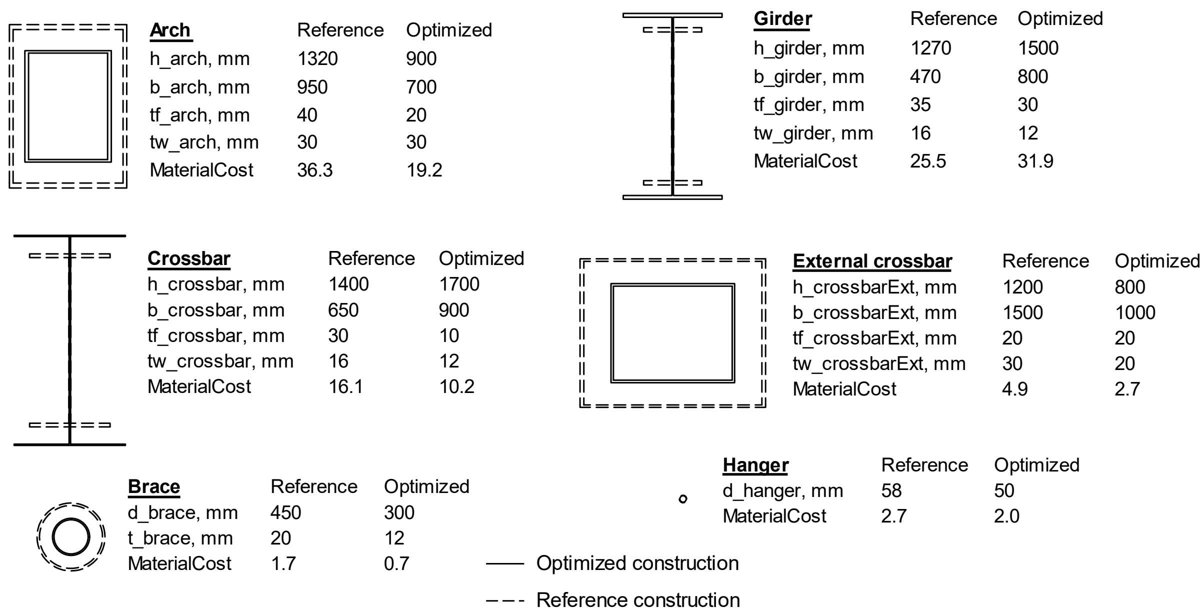


Fig. 4-12. Comparison of the reference and optimized constructions' cross-sections

4.3.4 Geometry optimization

During the geometry optimization, the system adjusted the static scheme parameters: *Arch rise*, *Arch inclination*, *Hangers number*, and *Braces number*. The cross-sections' dimensions and materials' parameters remained constant through analysis; their values matched the reference construction. Such a type of optimization task is uncommon, but it provided additional comparative data.

Each hanger system was analyzed separately. The ranges and steps were constant for *Arch inclination* (-15 to 15, step 1) and *Brace number* (4 to 6, step 2), while for *Arch rise* and *Hangers number,* the values depended on hanger layout (Table 4-5). The optimization was conducted through 30 generations of a population of 20.

Fig. 4-13 presents the outputs of the best examples (with the lowest *MaterialCost)* of subsequent generations, separately for each hangers system.

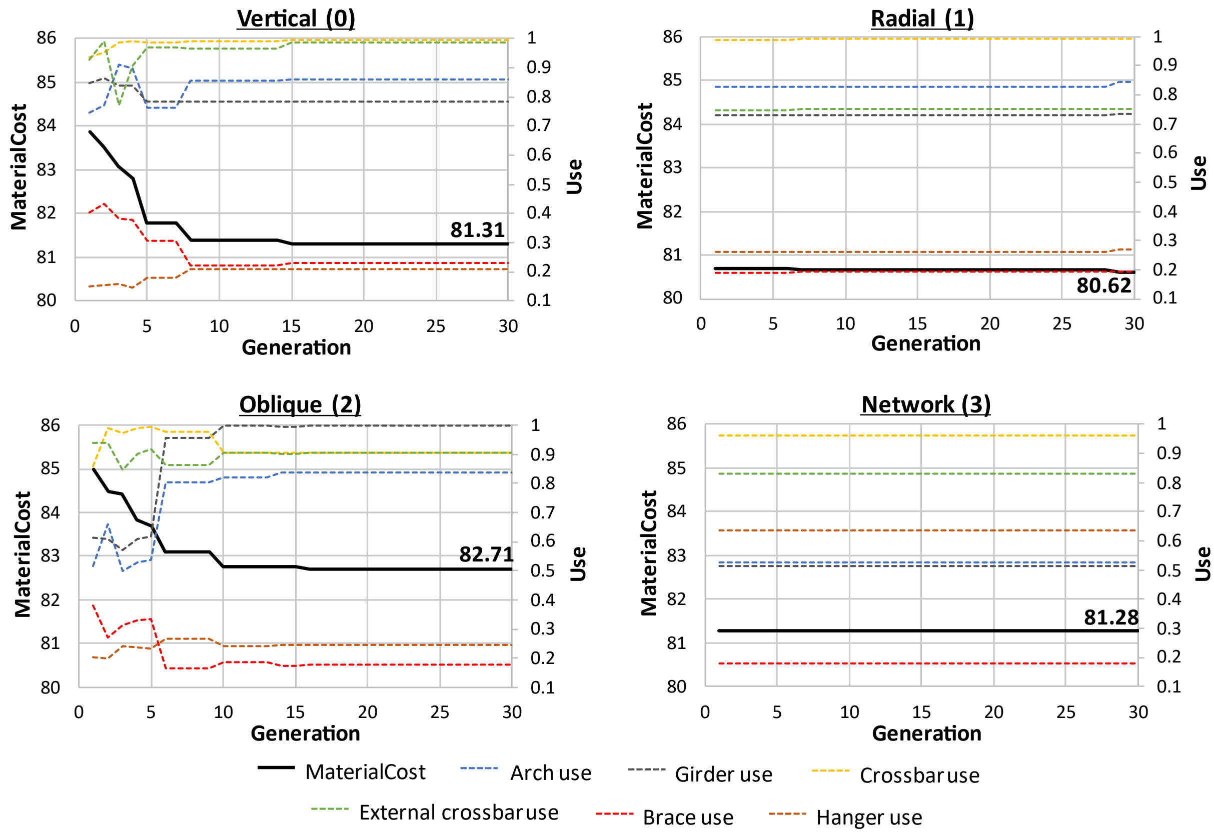


Fig. 4-13. Geometry optimization results

For the vertical and oblique systems, the *MaterialCost* decreased significantly in the initial part of the process, and the *use* indices varied. The other two cases' courses were steady: the first-generation solution was near-final for the radial system and the final for the network system. The use of hangers and braces was inefficient, similar to the cross-section optimization. The small set of parameters also prevented effective optimization of the other bar groups, especially for the network system. The values of the final solutions' parameters are listed in Table C-1.

4.3.5 Geometry and cross-sections optimization

During the geometry and cross-sections' optimization, the system adjusted the static scheme parameters and cross-sections' dimensions. The bar groups' materials remained constant and were set accordingly to the reference construction. This case merged two previous cases (4.3.3, 4.3.4), which gave the algorithm a broad scope of operation but also increased the complexity of the task.

Each hanger system was analyzed separately. The ranges and steps were constant for cross-sections' dimensions (

Table 4-4), *Arch inclination* (-15 to 15, step 1), and *Brace number* (4 to 6, step 2), while for *Arch rise* and *Hangers number* the values depended on hangers layout (Table 4-5). The optimization was conducted through 100 generations of population 20.

Fig. 4-14 presents the outputs of the best examples (with the lowest *MaterialCost)* of subsequent generations, separately for each hangers system.

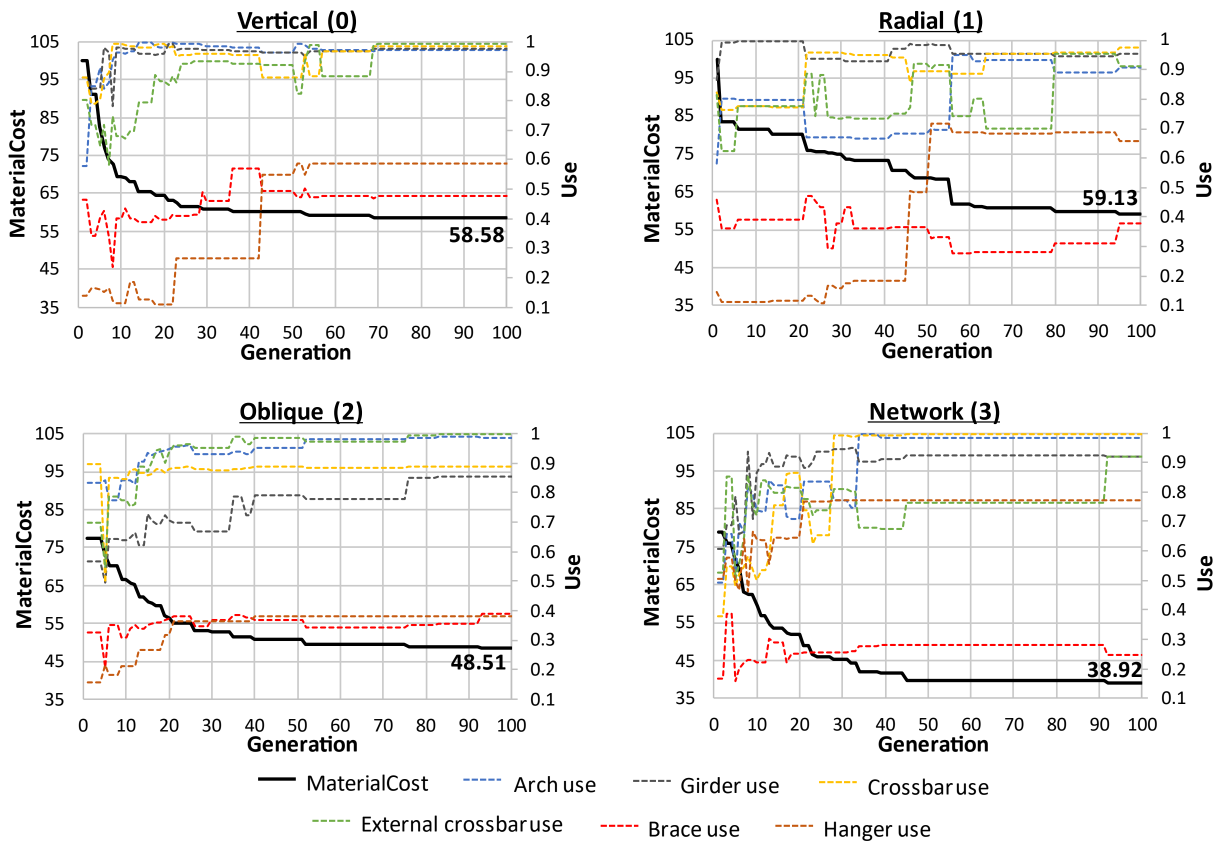


Fig. 4-14. Geometry and cross-sections optimization results

The algorithm utilized a wide range of adjustable parameters to provide more effective solutions than the previous, constrained analyses. The increase in task complexity did not decrease the effectiveness of the system. However, the higher number of iterations was reasonable: the high-variability part of the analysis extended to around 50 generations.

The use of hangers and braces increased significantly compared to the previous analyses. The uniform distribution for almost all bar groups is observed especially for network layout; it resulted in the most effective optimization. The values of the final solutions' parameters are listed in Table C-2.

4.3.6 Geometry, cross-sections, and material optimization

The last optimization case merged geometry and cross-sections' optimization with the materials' adjustment. The algorithm determined each non-hangers bar group material: S235, S275, or S355. This analysis was the most complex but also gave the system the highest flexibility.

Each hanger system was analyzed separately. The ranges and steps were constant for cross-sections' dimensions (

Table 4-4), *Arch inclination* (-15 to 15, step 1), and *Brace number* (4 to 6, step 2), while for *Arch rise* and *Hangers number* the values depended on hangers layout (Table 4-5). The optimization was conducted through 150 generations of population 20.

Fig. 4-15 presents the outputs of the best examples (with the lowest *MaterialCost)* of subsequent generations, separately for each hangers system.

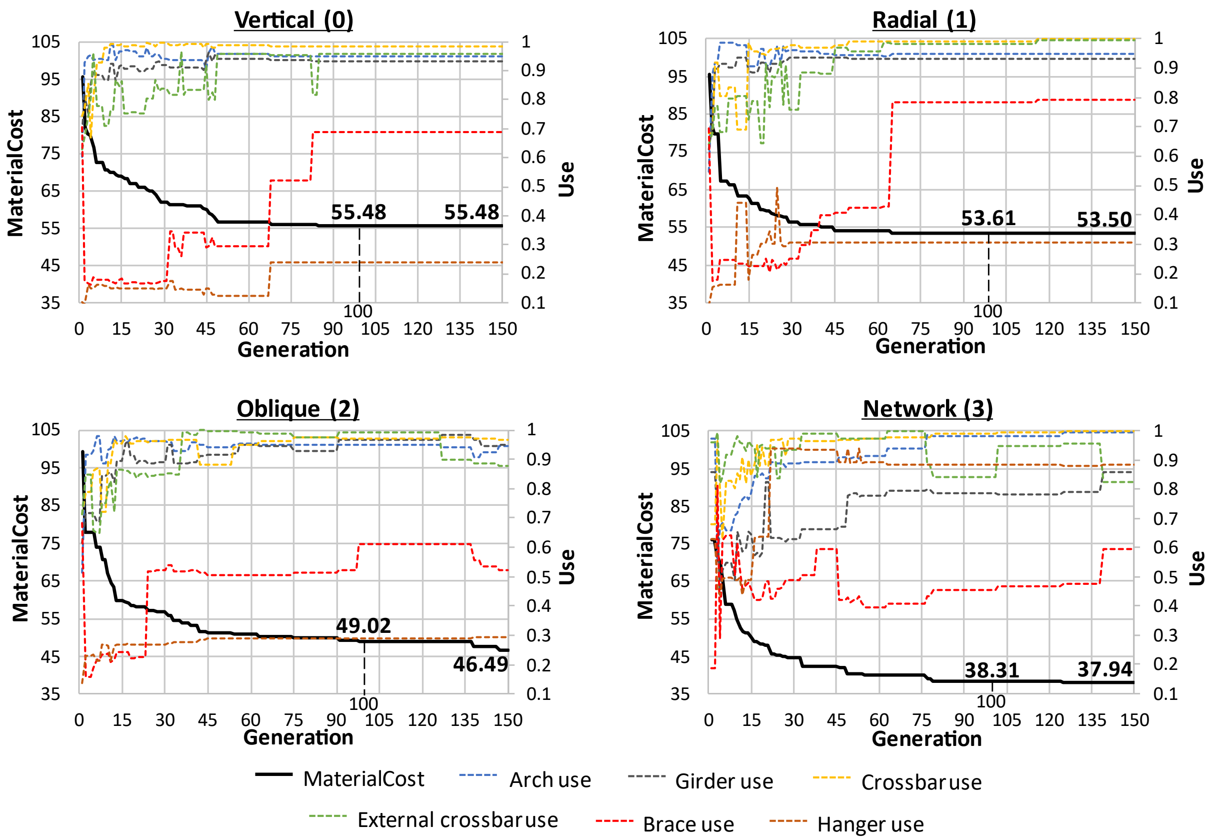


Fig. 4-15. Geometry, cross-sections, and material optimization results

The most complex case gave the algorithm the widest set of adjustable parameters, hence also the highest flexibility in the construction forming. The final *MaterialCost* parameters were lower for all the hanger systems compared to the previous case without material adjusting. However, it should be noted that both analyses have been conducted with a different number of generations. If comparing both analyses outputs after 100 iterations, the system effectivity is better for vertical, radial, and network layouts but worse for the oblique. It is a basis for the assumption that higher flexibility generally improves the system's performance, but the increase in complexity should be followed by the increase in the number of generations. The values of the final solutions' parameters are listed in Table C-3.

4.4 Conclusions

The solution introduced in this chapter uses visual programming to link BIM modeling, FEM analysis, and optimization algorithm. The solution is a step towards utilizing digital twins' techniques in the design phase. The system is an effective geometrical optimization approach. The algorithm iteratively analyzes and adjusts variants, leading to optimal structural choices. Therefore, the system is beneficial for designers in their day-to-day tasks. Moreover, the integration of BIM and FEM models is crucial for creating a fully-functional digital twin of a bridge. Here, the BIM and FEM models have been integrated by visual programming, offering parametricism and flexibility to different tasks.

The reference construction optimization confirmed the system efficacy; the structure has been optimized in each of the four cases. The growing task complexity did not diminish the system's performance – on the contrary, the flexibility provided with a broader set of adjustable parameters increased the effectiveness. The most challenging case of the geometry, cross-sections, and material adjustment provided the solutions of the best-balanced *use* indices, which resulted in the lowest *MaterialCost.*

The algorithm based its adjustments on the optimization objective function that rewarded lowering *MaterialCost* and refused non-fulfilling the static conditions. The function did not explicitly introduce the FEM analysis rules; the algorithm had to learn them – or rather, learn how to manage them - with subsequent iterations. Given the system's performance, it has the potential to address other complex indirect rules and constraints: sustainability and deterioration issues, erection planning, time-scheduling, and design aspects. Including them and utilizing visual programming-BIM modeling cooperation could lead to an automated end-to-end designing and modeling framework.

The script generates models of arch bridges, but its framework can be adapted to other structures. The described analyses lasted from 2 to 12 hours (single analysis of a hanger system case), operating on a standard personal computer – this time can be considered acceptable in the design practice. The used visual programming language, as an open-source project, can be freely utilized and tailored to specific engineering needs. These factors enhance the practical applicability of the solution. With the increasing popularity of visual programming for civil engineers and the development of optimization algorithms, alike automated solutions may become a regular enrichment of the design processes. In this framework, the system fulfills the iterative, time-consuming chores, allowing the designer to focus on creative tasks.

5 Generating synthetic data for point clouds' machine learning automation

5.1 Introduction

Point clouds are becoming basic sources of geometrical data for physical infrastructure. However, data about the construction's shape is insufficient for digitizing facilities in the form of data-rich digital twins and performing advanced analysis – for that, semantic data is also required. Since manual extraction of semantic data from point clouds and its interpretation is time-consuming, error-prone, and inefficient [315–319], much research effort is put into automation. The automation of point clouds is often based on machine learning (ML).

A properly operating ML system is based on two fundamentals: an algorithm suited for the task and a dataset for training. With the increasing computational power of algorithms, often the data is a bottleneck [244]. A proper dataset should be diverse, reliable, and contain a sufficient number of examples. Also, if the selected algorithm is a supervised one, the data examples must be labeled. If the examples' number is insufficient for the task complexity, ML solutions suffer from the data hunger effect [215] – insufficient training data inhibit achieving optimal algorithms' performance, hindering automation.

The acquisition of point clouds is time-consuming and expensive. Scanning some areas states practical challenges (e.g., closing a road to scan the lower part of a bridge). Moreover, manual labeling of points is tedious. Due to that, large datasets of point clouds to train algorithms are rare. The data hunger problem (the problem of insufficient data) is frequent in point cloud-dedicated ML solutions. One of the methods to overcome the data hunger problem is producing synthetic data. Synthetic data have some advantages compared to real data. Primarily, it is easier to acquire since there is no need for numerous scanning. Synthetic data is also scalable (the amount of generated data is limited only by computation power) and gives control over examples' features and distribution. Moreover, synthetic data ensures reliable annotations. However, to be used as an enhancement or replacement of real data in the process of training the machine learning algorithm, synthetic data properties should mimic the properties of real data.

Synthetic point clouds can be generated by geometrical models' sampling or by scanners' simulators. These techniques allow for preparing synthetic machine learning datasets acting as enhancements or replacements of real data. Several attempts have been made to create simulating tools. However, there is still a lack of publicly available solutions that give sufficient control over simulation parameters and acquired point clouds' properties, emulate the operation of scanners, and do not require dedicated preparatory work (e.g., creating models only for the scanning simulation).

The author decided to address the need to generate synthetic point clouds. For this purpose, the author developed a simulator of laser scanning called DynamoPCSim. The simulator generates synthetic point clouds from geometrical and BIM models. The simulator is based on ray tracing, a technique matching the operation of scanning lasers. Due to that, the produced point clouds' characteristics are closer to the real ones. The simulator is deployed as an add-in (a package) to Dynamo, an open-source visual programming environment. Since Dynamo enables linkage with Autodesk Revit modeling software, the simulator can utilize BIM models directly without requiring dedicated models prepared only for the purpose of scanning simulation. Due to the modular structure of the simulator and the flexibility of visual programming, it can be adjusted to different scanning properties and point clouds' parameters to be acquired. The validity of created examples is assessed in an experiment using PointNet – a state-of-the-art network for point clouds' semantic segmentation.

The created by the author system, DynamoPCSim, addresses the first but indispensable step in developing machine learning solutions automating the utilization of point clouds – acquiring the data to train the algorithms. The trained systems can extract point clouds' information, segment them (the segmentation example will be described), and finally, also generate and update the point-cloud-based geometrical models of digital twins. The automatically updated models will match the real conditions of the physical objects, realizing the idea of a model coexisting with an object through the whole lifecycle, also in the operational phase. As for bridges, the information about geometrical changes will allow to detect and compare deformations, monitor the structure, and alarm about hazards.

5.1.1 Need for point clouds' data

With the advances in computational power and data management, machine learning has become used cross-sectorally, also in AEC [250]. A machine learning system, to operate optimally, must be based on an algorithm proper for the task. The algorithm should be selected regarding its features and the task's objectives. As for point clouds, especially deep learning has recently gained attention [213]. Most algorithms in use, including neural networks, are supervised, meaning that labeled data is needed to train them. Proper training data is the foundation of ML systems [244].

A dataset for supervised machine learning is a set of instances (examples) with labels. Data examples and their labels can take different forms, depending on the task: for elements recognition on images, the examples are images, and labels are names of the elements on images; for estimating the time of tasks, examples are historic tasks' properties (e.g., task type, number of people involved), and labels are tasks' time. Data labeling (data annotation) is a crucial step in dataset preparation since unreliable labels can harm the performance of a system. Labeling can be performed with manual human annotation, semi-automated, or automated approaches. For point clouds, usually, every point has its label assigning it to a class (e.g., "column", "beam") or an instance (e.g., "column 1", "column 2", "beam A", "beam B"). Labeling point clouds manually is a typical approach; given the tediousness of this procedure [315–317,319,320], labeling big sets of point clouds is rare.

In order to generalize well (i.e., perform with acceptable performance on new, unseen data), the algorithm should be trained on an extensive set of examples. The dataset should be big and diverse (i.e., contain sufficient representatives of different classes and cases; class imbalance is a typical problem in preparing a dataset). Collecting point clouds presents different challenges. The acquisition is time-consuming and expensive in terms of labor and tools. Some scenes are dangerous (e.g., earthquake- [321] or wildfiredamaged buildings) or inaccessible [322]; others present practical issues (e.g., closing roads to scan their infrastructure). Adverse weather conditions like rain, dust, and wind should also be considered [206]. These challenges are acceptable for acquiring singular point clouds, but they hinder preparing massive datasets.

When the examples' number is not sufficient for the complexity of a task, ML solutions suffer from the data hunger effect [215] – insufficient training data inhibit achieving optimal algorithms' performance. Due to the scarcity of datasets, the data hunger problem is frequent in point cloud-dedicated ML solutions. Efforts to solve data hunger problems include methodologies, data annotation techniques, and producing synthetic data [215]. Methodologies try to overcome the lack of labeled data by using weakly-, semi-, and self-supervised methods and transfer learning. Data annotation techniques are attempts to increase the amount of labeled data by fully- and semi-automatic annotation; these techniques increase the effectiveness of labeling, but collecting data is still required. Producing synthetic data is convenient, especially when collecting real data is impractical. Synthetic data is scalable (the amount of generated data is limited only by computation power), gives control over examples' features and distribution, and ensures reliable annotations.

Utilizing publicly available datasets is another strategy to overcome the lack of data. As for point clouds, extensive public datasets are rare, especially those containing only real examples (e.g., S3DIS [323], ScanNet [324], SemanticKITTI [325], 10 RC Highway Bridges [238]); many datasets are synthetic (e.g., SynthCity [326], KITTI-CARLA [327]) or hybrid (containing real and synthetic examples; e.g., Paris-CARLA-3D [328]). Dataset sharing is undoubtedly a beneficial trend. Nevertheless, the datasets may pose numerous challenges for machine learning. The lack of labeling occurs, especially for real datasets. The lack of points' parameters (e.g., color, intensity) prevents inputting more advanced model features. Datasets are often acquired with one scanner, so collected data is undiversified in terms of scanner inaccuracies. Also, datasets are often oriented on a task or objects group and cannot be reused; in a standard machine learning approach, a model is trained with a dataset matching the task (e.g., for semantic segmentation of bridge point clouds, the dataset should contain point clouds with bridge elements).

5.1.2 Generating synthetic point clouds

Since scarcity of data prevents the machine learning introduction, producing synthetic training examples is a popular strategy. This strategy is especially beneficial when collecting and labeling real data is problematic. The time-consuming, expensive, and environment-dependent nature of collecting point clouds, as well as the ineffectiveness of labeling result in the scarcity of real point cloud datasets. Therefore, generating synthetic point clouds is a gate to empower the machine learning automation of point clouds' tasks. This section describes approaches and strategies for generating synthetic point clouds.

Producing synthetic data has numerous advantages. The most obvious one is facilitating the collecting process: simulations erase practical collecting issues and are faster and cheaper. As for point clouds, the possibility of generating examples of inaccessible places is also beneficial. The collected data amount is limited only by available computation power and storage. The dataset properties, e.g., distribution of examples, can be designed, which can address the class imbalance. Depending on the complexity of a simulation tool, various properties of examples can be collected. Also, the data can be automatically labeled. However, aside from the advantages, generating synthetic data methods must fill a crucial requirement: to be used as an enhancement or replacement of real training examples, synthetic data properties should be sufficiently similar to real data so that any classifier trained on the synthetic data generalizes to real data with minimal loss in performance. For that, the generating strategy and simulation tool's features are significant.

Generating synthetic point clouds can be approached differently. The first technique, augmentation, involves modifying a real point cloud with different strategies, e.g., introducing small perturbances into points' properties, deleting or adding points, or translating and rotating point groups. Augmentation has constraints: adding properties to points is hindered, and the examples are diversified only in terms of local properties – simulating different scanners' locations (e.g., for incomplete scans) is limited. Other, more advanced techniques utilizing real point clouds are: collecting samples of individual objects to arrange them into new scenes [329] or augmenting real point clouds with synthetic elements [317]. These techniques enhance a dataset but require preexisting labeled real examples.

Utilizing digital models is another strategy, allowing the generation of synthetic point clouds without the need for preexisting real examples. In addition to geometrical data, model elements may contain semantic data, which can be attached to points or used for automated labeling. Two main strategies utilizing models are sampling and simulations.

Sampling involves gathering points of model elements' geometries. The points should be collected on the exterior faces of geometries. Usually, the points are collected uniformly, then transformations can be added to simulate real-world imperfections. Several sampling attempts have been performed. [213] presents a process of transforming Revit models to sampled point clouds using AutoCAD, Sketchup, and FME Workbench; labeling is done manually, and the produced point cloud is volumetric (not only the exterior faces of geometries have been sampled). In [316], Blender is used to create CSV files with coordinates and labels of sampled points of historical buildings' models in OBJ format. In [218], Grasshopper visual programming has been used for defining parametric shapes of historical dome systems, which has been converted to mesh and subsampled in CloudCompare. [319] presents subsampling IFC and complementary OBJ models to generate labeled point clouds with coordinates and color data. Sampling does not emulate the operation of scanners, resulting in distinctions between generated point clouds and real ones.

Scanning simulations is a more advanced technique. It consists of creating synthetic scanners following real scanners in their operation way. It is usually approached with ray casting [330], a computer graphics rendering technique; with that, the intersection of a simulated ray and model geometries is gathered, similar to real scanning. The scanner can have adjustable properties, which allows the emulation of different real scanners. Also, the synthetic scanner can be static or mobile. The distribution of points depends on the properties and line-of-sight of the synthetic scanner, making the resulting point cloud closer to real. The simulators may also introduce occlusions, density variations, and noise.

Several synthetic scanners' simulation attempts have been performed. In [331], BlenSor [332], an add-in to Blender [333], is used for generating synthetic point clouds from IFC models; labeling has been done with the k-nearest neighbor algorithm using other labeled point clouds. In [334], BlenSor is utilized for FBX models; created point clouds have been labeled manually. Helios [335], another scanning simulator, has been used with OBJ models and accompanying XML files [336]; Revit models have been converted to IFC and then to OBJ with Blender. Also, other simulation tools are used: the robotic simulator Gazebo [337] for synthetic point clouds to study natural terrain traversability [338], the driving simulator CARLA [339] for a mobile scanning simulator [340], or Unity 3D engine [341] for a modular LiDAR simulator regarding metrological parameters and scanning error models with semi-automatic labeling [342] and for simulating the Pico Flexx camera to generate point clouds of sewer networks [343]. Revit and Tekla models have been used for generating synthetic post-earthquake data [321]. Utilizing Grant Theft Auto V (GTA-V) game scenes is another interesting approach [216].

Attempts to generate synthetic point clouds with simulators differ in utilized tools, types and formats of models, synthetic scanners' parameters, noise and errors strategies, collected point clouds' properties, and labeling strategy (manual, semi-, or fully-automated). Though the examples are reported to be useful for dedicated tasks, some issues hinder their wider, versatile use.

The unrealistic properties of generated point clouds is the most significant issue. As mentioned, the ray casting simulators' operation leads to a more realistic points' distribution compared to uniform surface sampling. Also, replicating simulation errors influence the distribution. The errors lead to imperfections in points' properties. Two leading strategies to simulate errors are: introducing random noise (usually Gaussian-random translation) or introducing simulated scanners parameters (e.g., horizontal, vertical, range errors). Introducing random noise is easier to applicate but does not give enough flexibility to emulate real tools. Also, not only coordinates may be a subject of scanning errors – nevertheless, it is rarely addressed.

Although ray casting, a basis of simulators' operation, is sufficient for most cases, the complex approaches can benefit from ray tracing. The difference between these two is that ray casting does not permit rays' reflections – the first intersection of a ray and scanned geometry declares the point location. However, for some materials, this is not the case in real scanning – if a material is reflective, the ray bounces until it reaches a non-reflective material; only then the point's position is declared. This phenomenon influences point clouds of areas with mirrors, windows, or water. Ray tracing, a more complex approach, is rarely applied in scanning simulators [342], with simplified ray casting as a default.

The lack of additional points' properties (e.g., color, intensity, surface normal) is another issue. It can result directly from the lack of simulators' features or the lack of semantic data in models on which the simulators operate. Intensity is a special case since its calculation is not straightforward [344,345] and depends on material properties and scanners' details, which are not always revealed [207]. Nevertheless, additional properties can be useful for machine learning tasks [346].

Labeling strategy is crucial for the practical usefulness of a scanning simulator. Nowadays' solutions recognize this importance, so manual labeling is rare. However, some solutions introduce semi-automated strategies that still require significant labor. The fully-automated labeling leverage semantic data in models, annotating the points regarding the name or class of model elements. The practical importance of automated labeling is manifested in the reported time for manual or semi-automated labeling: avg. 1.52 hours for a reinforced bridge (avg. 17 216 081 points per point cloud) [347], 12-16 hours for a gasoline refinery [334].

Most simulating solutions are developed from the perspective of the target problem [342]. The solutions lack functions because they were not required for collecting data for a given research. This approach is understandable but hinders reusability. Also, some solutions require complex preparations, like converting models with numerous software or creating models only for the scanning simulation. These factors force researchers to begin automation projects by creating their own data-collecting solutions. A starting point of a versatile, flexible tool would be beneficial to point researchers' focus to tasks beyond generating data.

5.1.3 Requirements of a versatile simulation tool

The need for automating point clouds' tasks leads to machine learning, which requires data to operate. The need for point clouds' data, together with problems of collecting real examples, lead to synthetic point clouds' generating solutions. Among them, synthetic scanners emulate the operation of real tools, giving a possibility to generate synthetic data of properties following the real. A versatile, applicable tool would be promising to enhance the point clouds' automation research. Examples of automation tasks and issues of current solutions form requirements for such a tool:

- Emulating real scanners operation
- Applicability (versatility, automated and straightforward process, customization)
- Fully-automated labeling
- Emulating scanning errors
- Introducing noise
- Collecting additional point clouds' properties

5.2 Visual programming synthetic point cloud simulator

The requirements for a versatile scanning simulator were the basis for creating the tool: DynamoPCSim. DynamoPCSim is a terrestrial laser scanning (TLS) simulator based on visual programming and operating on BIM and parametric geometrical models. It implements ray tracing and fully-automated labeling. This section presents its way of operation and how it addresses the requirements.

5.2.1 Characteristics and operation

5.2.1.1 Emulating real scanners operation: ray tracing

LiDAR terrestrial laser scanners emit rays pulses, which reflect from surrounding elements, determining locations and properties of point cloud's points. This process can be simulated by ray tracing.

Ray tracing is a 3D computer graphics rendering technique. Ray casting, a simplified version of ray tracing, is the default for scanning simulators. However, implementing ray tracing allows to emulate the phenomena of rays' reflections. In real scanning, if a surface hit by a ray is of reflective material, the ray reflects until it reaches a non-reflective surface. It influences point clouds of areas with mirrors, windows, or water. The points resulting from reflections are considered noise and are typically discarded during preprocessing. However, the simulator should emulate real scanning with all its imperfections rather than produce perfect outcomes (especially since machine learning solutions to detect reflected noise points is a potential area of research – for that, examples including reflected noise points are indispensable).

DynamoPCSim implements ray tracing (using geometry3Sharp library [348]) with the possibility of limiting the maximum number of reflections (Fig. 5-1). No reflections result in ray casting.

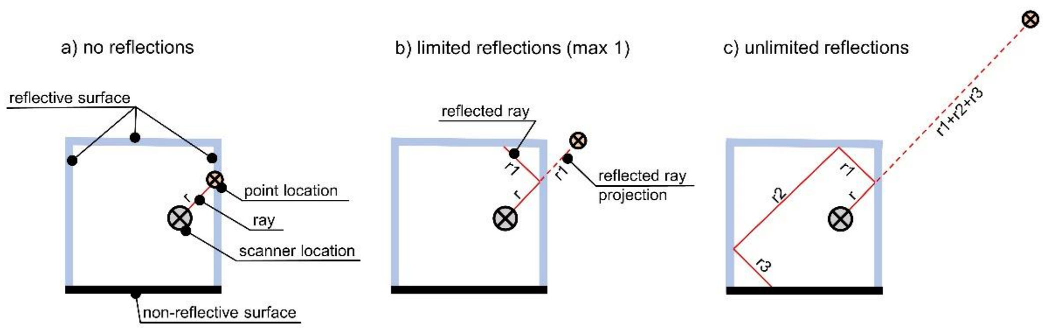


Fig. 5-1. Ray tracing implemented with the possibility of limiting the maximum number of reflections: a) no reflections, b) limited reflections, c) unlimited reflections

Other issues of real scanners' operation are scanner location, horizontal and vertical steps, field of view, and maximum range. DynamoPCSim implements them as adjustable parameters.

5.2.1.2 Applicability: visual programming and use of non-specific models

DynamoPCSim is implemented as a set of Dynamo visual programming language methods. Visual programming is an alternative to traditional text-based programming, where the logic of algorithms is declared by linking blocks representing methods instead of text commands [183]. Blocks take inputs, process them, and produce outputs, which can be inputs for the following blocks. Ease of use increases visual programming popularity, especially among non-programmers. Visual programming is becoming utilized widely also in AEC, especially given the linkage of visual programming software and BIM modeling environments.

Dynamo [349] is an open-source visual programming environment. Its methods are focused mainly on parametric geometrical models' generation, but its utilities can be extended by external packages – DynamoPCSim is such a package of custom nodes (*zero-touch* nodes) implemented in C# programming language. Dynamo can be used as a standalone application or be linked with commercial software, including Autodesk Revit [350]. This collaboration enables creating, editing, or utilizing Revit BIM models by Dynamo algorithms.

DynamoPCSim, as an extension to Dynamo, inherits characteristics of visual programming. The modularity of visual programming methods ensures the versatility of applications. Parametricism enables adjustments to the task regarding model geometries and scanning properties. The algorithms can operate on parametrized models, first adjusting its geometry and then processing it. For the scanning simulations scenario, one algorithm can modify a model, perform scanning and save the outcome as an XYZ file. Model modifications depend on its parametrization: an algorithm can adjust dimensions, location, or the number of model elements, producing various scenes for scanning simulations. It automates the process of collecting data.

Dynamo can operate as an open-source, standalone application and, as such, process geometrical models. However, due to the possible linkage with Autodesk Revit, it can also operate on BIM models. It straightforwards the process of scanning simulations – there is no need to prepare specific models only for simulations; the simulations can be performed on a BIM model created during the construction project processes.

5.2.1.3 Fully-automated labeling: use of model semantic data

DynamoPCSim implements fully-automated labeling – points labeling is performed directly during scanning. Labeling can be adjusted: points can be labeled with the name or class of an element, which geometry has been hit by a ray, or with other user-specified labels. If additional labels are required (e.g., one label for element class and another for surface material), they can be implemented as additional parameters attached to points. The labels can be sourced from BIM models' semantics.

5.2.1.4 Emulating scanning errors

Scanning errors affect the properties of point clouds' points. Scanning errors can be divided into random, gross, and systematic [351]. Random errors cannot be predicted, and their behavior is described with stochastic models. Gross errors (e.g., multi-path reflections' results) also do not have a universal model, but this type is often detected and eliminated during preprocessing. Systematic errors result from environmental and instrumental sources. Environmental errors come from, e.g., temperature and atmospheric pressure and are significant particularly for long-range scanning, like airborne laser scanning. The instrumental errors come from features and imperfections of scanning tools, e.g., axis offsets, nonorthogonality, and wobble. Instrumental errors are well-researched [207], and their models are declared. This type of errors is reflected in scanners' specifications (e.g., the uncertainty of range, vertical, and horizontal steps). As reported [344,352], the producers' specifications are reliable and can be used to declare instrumental errors.

DynamoPCSim implements errors on different layers. Instrumental errors are implemented with horizontal, vertical, and range errors of a synthetic scanner. These errors are applied to points' coordinates as normal-distribution random values of base and additional (depending on the distance from the scanner to the point) standard deviation declared by a user. Besides scanners' errors, additional random translations of points or alterations in other parameters can be introduced. These errors can be applied to all points or selected groups – for instance, alterations of greater values may be applied to points of more varying elements, e.g., vegetation.

5.2.1.5 Introducing noise

Real scanning scenes usually include or are affected by elements not occurring in models (e.g., vegetation, movable non-persistent objects). Therefore, the ability to introduce noise points is an important feature of a scanning simulator to make the synthetic point clouds closer to real. Adding noise to machine learning training examples increases the algorithms' performance [329].

Noise can be introduced in DynamoPCSim with different strategies. The most straightforward is placing noise points randomly with user-selected area and density. A more complex approach is to model or generate noise elements and include them in a scanned scene. If the elements have parametrized dimensions and locations, an algorithm can dynamically change their appearance. In this scenario, the noise elements may act as clutter objects. The noise points can also be copied with translations and rotated.

5.2.1.6 Collecting additional point cloud properties

Point clouds' points can have different properties. The indispensable are coordinates reflected by X, Y, Z location. Optional additional properties include, e.g., color, surface normal, intensity, return number, number of returns, waveform data, or classification. Color and intensity are reported as useful for interpretation and machine learning tasks [206,344].

DynamoPCSim attaches additional properties (properties beyond coordinates and labels) to points during scanning. Types of properties can be divided into semantics-derived and calculated. The semantics-derived ones utilize semantic data of model elements. Properties like color (in R, G, B form) or material (both name and its properties) are attached to points of the element's surfaces. The user can specify which properties should be derived from semantics.

Calculated properties may result from semantical, geometrical, and scanning factors. Values of calculated properties are not derived directly but calculated with a model. Examples of calculated properties are surfaces normal or intensity.

Intensity indicates the power of a ray reflected from the scanned object [344]. Its value depends on scanning range, target material and geometry, and atmospheric and instrumental factors. As reported [207,345], particularly the atmospheric and instrumental transmission factors are nontrivial to quantify, which results from the lack of standardized data in scanners' specifications. The difficulties reflect in various attempts to intensity calculations [353–355], including machine learning predictions [345].

As the intensity is reported as useful for machine learning training (e.g., detecting cavities and cracks, humidity, and biodeterioration [344]), implementing its calculation in a simulation scanner would be beneficial. However, without the exact parameters of LiDAR sensors, intensity cannot be adequately calculated during the simulation [345]. Therefore, DynamoPCSim introduces a simplified way to calculate intensity regarding only the scanning range (5-1).

$$I = 1/d \qquad (5\text{-}1)$$

where: I – intensity; d – distance from the scanner to point.

However firmly simplified, such calculated values may be useful for machine learning tasks of scenes of uniform material. Features normalization can decrease potential inefficiencies of unrealistic ranges of the simplified intensity values.

5.2.2 Features and workflow

DynamoPCSim is a visual programming tool, so it derives a modular structure. Methods of the simulator, represented as visual programming blocks, are linked in an algorithm to form its logic. The methods are divided into three main groups: *Mesh*, *Scanner*, and *PointCloud*. An algorithm can also utilize other visual programming methods to, e.g., operate on a model, which may precede the scanning process. Fig. 5-2 presents a general flowchart of the solution.

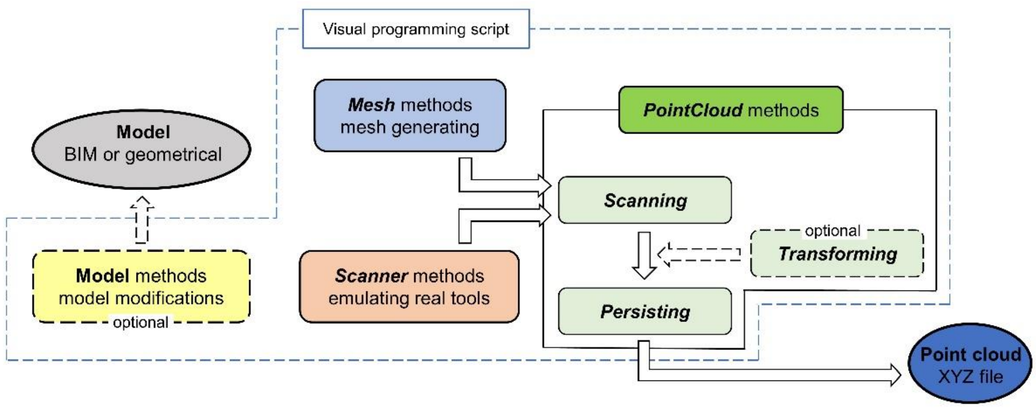


Fig. 5-2. DynamoPCSim flowchart

5.2.2.1 Model (a pre-stage)

Models alteration is an optional pre-stage. Although the scanning can be performed on models of unchangeable elements, alteration of models' parameters (both geometrical and semantical) can substantially enhance dataset diversity. The alteration can be done manually or algorithmically, which automates the process. Dynamo algorithms can adjust models in various ways and regarding numerous parameters – the approach is dictated by the task needs and user invention. The most straightforward are varying model elements' parameters and location, as well as generating geometries.

5.2.2.2 Mesh

Mesh generation is the first step in the simulation process. Mesh is a triangulated representation of geometries. In this solution, the mesh is enhanced with semantic data. Model elements are transformed into a semantically rich mesh, which will be the subject of ray tracing during scanning.

Mesh is composed of *MeshElements*, which are based on model elements. A *MeshElement* can be based on a geometrical object or a BIM model element. A *MeshElement* based on a BIM model element can derive its semantic data (e.g., material name and properties, like reflectivity, essential for scanning with rays' reflections). *MeshElements* can also be enhanced with other user-specified data. These properties are called *MeshAdditionalProperties*. A *MeshElement* can have a label, typically a name of a model element or its class. *MeshElements* labels and properties will be a base for properties of generated point clouds.

Fig. 5-3 presents an example of creating a *Mesh* based on *MeshElements* from BIM model elements representing a bridge abutment. The labels are the BIM element names. The *MeshElements* derive the BIM elements' surfaces colors properties. The *Mesh* is visualized with the *Mesh.Visualize* method.

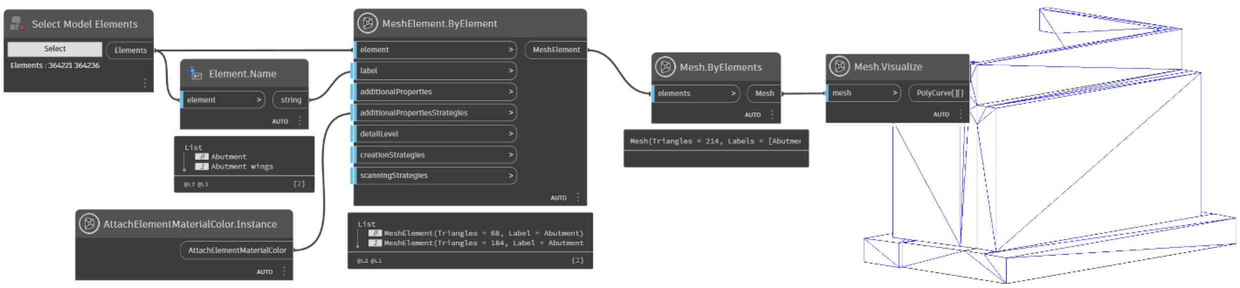


Fig. 5-3. Mesh creation example

From a geometrical perspective, a mesh is a set of triangles. The complexity of a *Mesh*, reflected in its triangles number, affects computation efficiency (both *Mesh* creation and later scanning). Therefore, some model elements may be the subject of optimized mesh creation. The mesh creation can be adjusted with *MeshElementCreationStrategies*. *ByDetailedCreation* is the default, non-optimized option in which the whole element's geometry is triangulated. *SelectFacesWithBiggestAreaByNumber*, *SelectFacesWithBiggestAreaByParameter* (with the *parameter* declaring a portion of selected faces), and *SkipCurvedFaces* allow excluding part of geometries from processing, reducing *Mesh* complexity and increasing computation efficiency. The strategies can be applied individually or stacked. Fig. 5-4 presents examples of *MeshElementCreationStrategies* implementation for a bridge barrier.

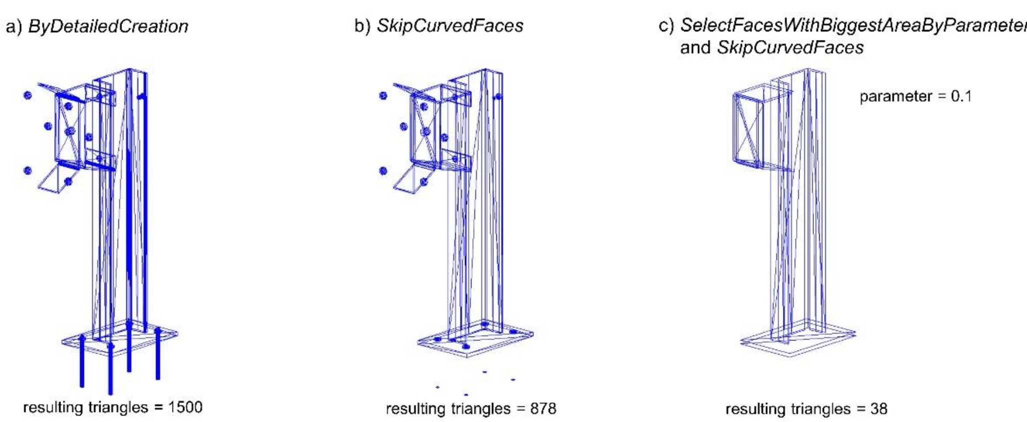


Fig. 5-4. Examples of MeshElementCreationStrategies implementations: a) ByDetailedCreation, b) SelectFacesWithBiggestAreaByParameter, c) SelectFacesWithBiggestAreaByParameter and SkipCurvedFaces

MeshElements can have different *MeshScanningStrategies* attached: *AddConstantTranslation*, *AddRandomNormalTranslation*, and *AddRandomTranslation*. They introduce alterations in points' locations. The scanning strategies are utilized during scanning and can also be applied in the next stages.

However, the possibility to attach them for individual *MeshElements* enables customization – other *MeshElements* can have different scanning strategies.

Generated *Mesh* can be saved with *Mesh.SaveToJSON* method. The saved JSON file can be used to retrieve the *Mesh* with *Mesh.FromJSON*. It is useful particularly for the reusability of complex *Meshes*, which creation may take a longer time.

Table D-1 lists the main *Mesh* classes and methods.

5.2.2.3 Scanner

The *Scanner* group of methods enables customizing the synthetic scanner's properties following the ones in real scanners' specifications. *HorizontalAngleResolution* and *VerticalAngleResolution* declare angular steps of emitting rays, affecting the scanning resolution. *TopBlankArea* and *BottomBlankArea* limit the S*canner's* vertical field of view, while *MaximumRange* limits its range.

A *Scanner* can also have errors attached: *HorizontalScannerError*, *VerticalScannerError*, *RangeScannerError*. Properties and errors enable replicating real scanners and affect the properties of generated point clouds.

Table D-2 lists the main *Scanner* classes and methods.

5.2.2.4 PointCloud

The *PointCloud* methods are the final phase of the simulation. The *PointCloud* is first generated by simulated scanning; then, it can be transformed and finally saved as a dataset example. Table D-3 lists the main *PointClouds* classes and methods.

A *PointCloud* can be generated by two methods: *PointCloud.ByScanning* or *PointCloud.ByScanningWithReflections*. *PointCloud.ByScanning* does not regard reflections, resulting in ray casting operation way; *PointCloud.ByScanningWithReflections* does, resulting in ray tracing. Both methods have shared inputs: *Mesh* to be scanned, *Scanner*, *scannerPosition* declaring the *Scanner* location in a model, and *additionalPropertiesTypesToAttach*. *AdditionalPropertiesTypesToAttach* declares which properties will be attached to generated point cloud's points (besides coordinates and labels, which are always included). A user can specify custom additional properties or choose predefined ones: *color*, *intensity*, or *normal*. *PointCloud.ByScanningWithReflections* has two additional inputs regarding reflections. *ReflectivityStrategy* is to declare which triangles of *Mesh* should be considered as reflective. A user specifies a *Mesh* property (usually resulting from the material, e.g., reflectivity or opacity) and the property value threshold – if the property value exceeds the threshold, the *Mesh* triangle is treated as reflective. *MaxRaysReflections* limits the possible reflections' number (Fig. 5-1).

Fig. 5-5 presents point clouds generated without and with reflections. Reflections introduce noise, typically discarded during preprocessing. However, generating training examples with labeled (with the *isProductOfReflection*) reflected points is desirable, as automated detection of this noise type is a potential area for machine learning research, currently limited by the lack of data.

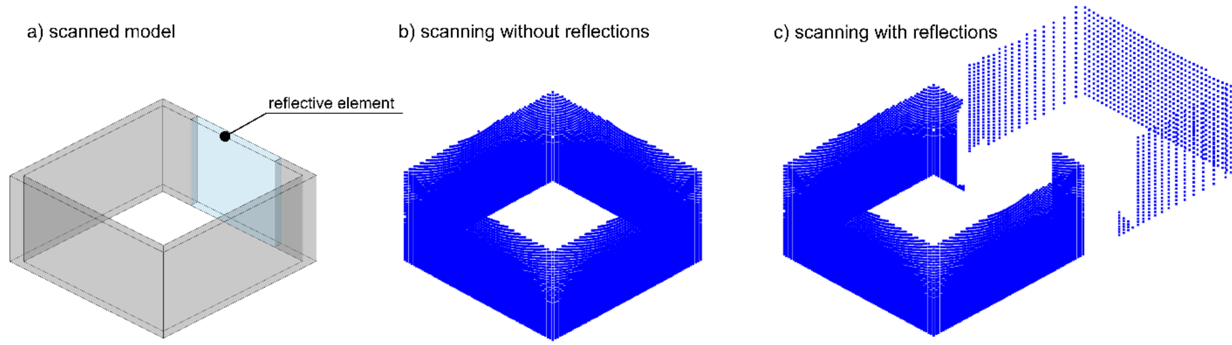


Fig. 5-5. Synthetic point clouds examples: a) scanned model; b) scanning without reflections c) scanning with reflections

A model can be scanned from multiple scanners' locations during an algorithm run. The point clouds from single locations can be merged with *PointCloud.Join*. This method replicates the registration (registration merges and adjusts scenes to global coordinates, but adjusting is not required for the simulator since the singular scenes are already in the model's global coordinates). *PointCloud.Join* has *similarityRange* input, which declares how closely located points should be merged. Joining enables generating full-coverage or partial scenes (useful, e.g., for automated completion training).

Fig. 5-6 presents exemplary single-location point clouds and their merge; point clouds are visualized with *PointCloud.Visualize* method. Due to limitations in the BIM environment, *PointCloud.Visualize* visualizes only points' locations, not regarding other properties, e.g., color; the other properties can be visualized in point-clouds dedicated software.

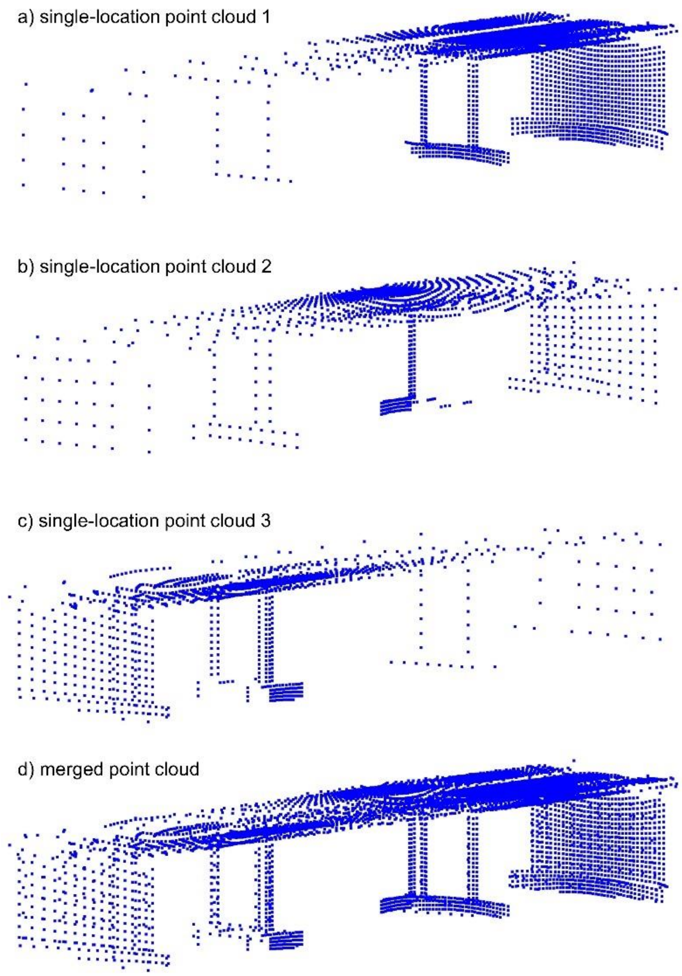


Fig. 5-6. Joining synthetic point clouds example: a), b), c) single-locations point clouds, d) merged point cloud

Generated point clouds can be filtered using *FilteredByAllLabels* and *FilteredByLabel* methods. The first produces a list of point clouds containing elements of only one of the labels – for instance, if an input point clouds have points of three labels (e.g., *beam, column, terrain*), the resulting list will contain three point clouds: one with *beam* points, one with *column* points, and one with *terrain* points. The second one produces a point cloud containing points of a label specified by the user.

Generated point clouds can be transformed with *PointCloud.Transform* method. The method takes *transformationStrategies* as input. *TransformationStrategies* allow various alterations in points features, including translating, rotating, copying, and deleting points, adding noise, and altering properties (Table D-3). The strategies can be applied individually or stacked; they can also be applied to all or selected points based on labels or properties' values. Transformations are an efficient way to enhance the dataset – since scanning is the most computationally expensive process, transforming a point cloud with various transformations limits the data generation time. Fig. 5-7 presents transformations examples.

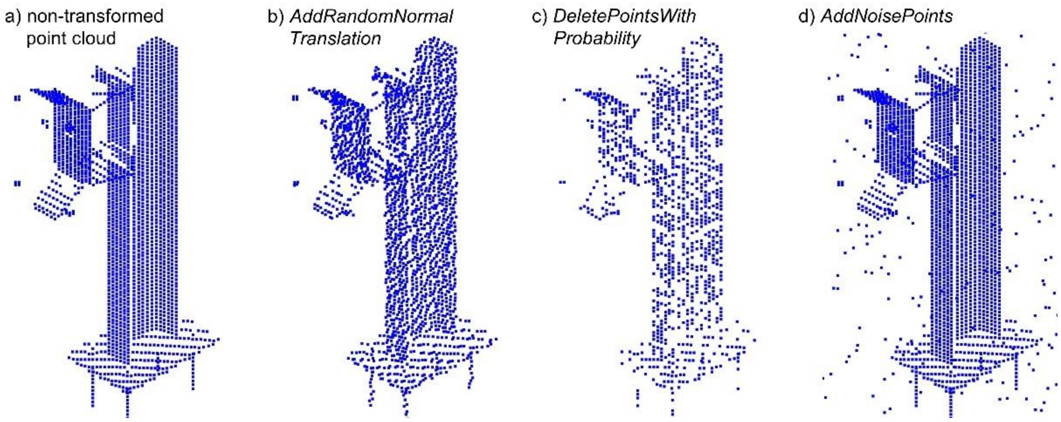


Fig. 5-7. Transforming synthetic point clouds examples: a) non-transformed point cloud, b) AddRandomNormalTranslation, c) DeletePointsWithProbability, d) AddNoisePoints

Finally, the synthetic point cloud can be saved with *PointCloud.SaveToXYZ* as XYZ file. A user can specify the point cloud's properties and custom comments to save. The comments are useful for data versioning and storing the example's information, e.g., time of creation, index, scanning properties, or the number of points per label. Listing 5-1 presents saved point cloud records' fragment. XYZ point clouds can also be retrieved by an algorithm with *PointCloud.FromXYZ* to enable its algorithmic processing.

Listing 5-1. Saved point cloud in the XYZ format

```
# PointCloud:
#     Points: 43369
#     Labels: {Terrain, Car, ColumnCap, ColumnPier, Vegetation}
#     Points per label:
#        Terrain: 35189
#        Car: 212
#        ColumnCap: 3173
#        ColumnPier: 2549
#        Vegetation: 2246
# Scanner position: Point(X = -7.829, Y = 3.707, Z = 2.000)

# Properties: {X; Y; Z; R; G; B; Label}
   -7.18721041   3.67348326   0.30447498   176   168   179   Terrain
   -7.09395208   3.73058129   0.32160446   182   190   178   Terrain
   -6.91809341   3.70114433   0.28966486   181   180   180   Terrain
[…]
   4.98684439   -2.07249774   0.13930763   114   119   117   ColumnPier
   4.98798234   -2.06895748   0.64154200   105   116   111   ColumnPier
[…]
```

5.3 Validation for machine learning

The main expected purpose of the tool is to generate training data for machine learning. To validate the usefulness of generated synthetic point clouds for machine learning tasks, I performed a class-based semantic segmentation experiment using PointNet [222], a deep learning architecture of proven performance [204]. The task was to segment a point cloud of a bridge column and its surrounding into four classes: *ColumnPier*, *ColumnCap*, *Car*, and *Terrain* (including vegetation). PointNet architecture has been trained on synthetic point clouds and then validated on both synthetic and real point clouds. The validation approach was to assess the similarity of the model performance on synthetic and real validation sets.

5.3.1 Data validation approach

I have selected the validation approach to assess if the model trained on the synthetic data can generalize well on real-world data – if so, the synthetic data can replace or enhance real data for machine learning. Therefore, I was concerned more about the relation of performance (accuracy, recall) on synthetic and real data rather than the absolute performance of the model (to some extent – still expecting the model to have acceptable performance and outscore simple heuristics, e.g., predicting always the most frequent class). For that reason, I also had not implemented training data processing, e.g., augmentation or class balancing. The determined validation approach validated the data rather than the model.

5.3.2 Real data

The point cloud selected for validation presents a bridge on the A1 highway above Moczury Lake in Mikołów, Poland (Fig. 5-8). It was acquired using the Z+F Imager 5010 scanner (Table 5-1) by the author assisted by 3D Format company specialists. The acquired point cloud was manually cut into the area containing the bridge’s column and its surroundings and then segmented using CloudCompare [356] software, which took 50 minutes. The point cloud consists of 18 923 287 points: 762 574 *ColumnPier*, 773 493 *ColumnCap*, 267 965 *Car*, and 17 119 255 *Terrain*.

Table 5-1. Z+F Imager 5010 scanner specification [357]

Parameter	*Value*
maximum range	187.3 m
range uncertainty (constant part)	1 mm
range uncertainty (variable part)	10 ppm
max. vertical field of view	320 °
max. horizontal field of view	360 °
min. horizontal step size	0.0002 °
min. vertical step size	0.0004 °

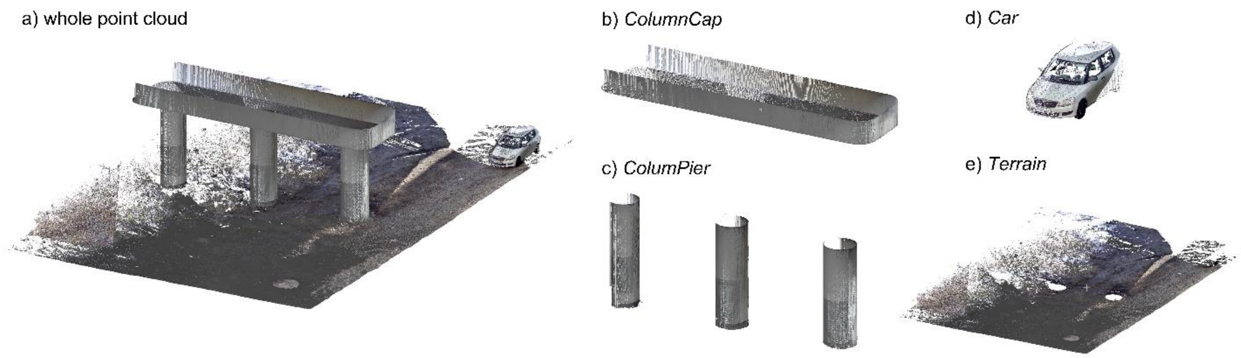


Fig. 5-8. Point cloud used for the validation experiment: a) whole point cloud, b) ColumnCap, c) ColumnPier, d) Car, e) Terrain segments.

5.3.3 Synthetic data

The synthetic data has been generated with the DynamoPCSim tool utilizing a BIM model of the bridge column and its surroundings, matching the real scene (Fig. 5-9). The model consists of 3 column piers, column cap, car, and terrain. The column piers, cap, and terrain model elements' locations were static, while the car location varied during the data generation. Before each scan, the algorithm declared the car location with a random pick in the specified area. Two scanners' locations were also declared randomly. The algorithm also altered the position and density of vegetation included in the *Terrain* class. The vegetation had not been modeled but implemented with *AddNoisePoints* transformation. Also, *AddRandomNormalTranslation* and *AddRandomNormalPointsColorAlteration* have been implemented to ensure the diversity of synthetic point clouds' properties. The synthetic scanner properties have been set to emulate the Z+F Imager 5010 scanner (Table 5-1).

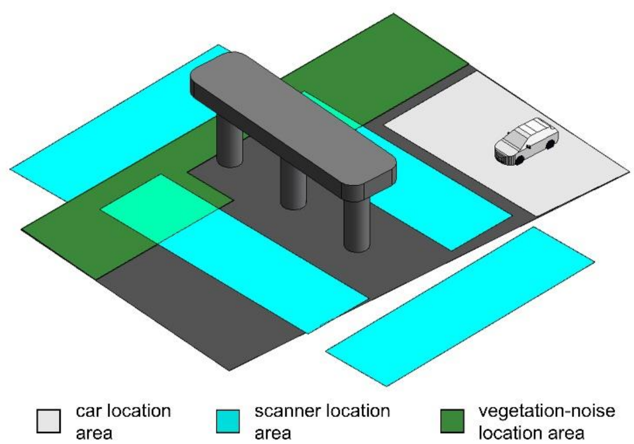


Fig. 5-9. Model for generating synthetic point clouds' dataset

One thousand synthetic point clouds have been generated (Fig. 5-10). Each point cloud merges scans from two locations, which ensures diversification of points distribution while allowing for an efficient generation. The acquisition time for a point cloud varied from 15s to 130s with an average of 49s, resulting in acquiring the entire dataset in 13h 37min on a standard laptop (AMD Ryzen 7 4800H 4.2 GHz, 16 GB RAM, 512 GB SSD). The vertical and horizontal angular steps have been set to 1° for efficiency. The generated point clouds' points number ranged from 2270 to 61591, which is much lower than for a typical point cloud. Nevertheless, given that for the machine learning task the input point clouds are sampled, this density is sufficient for learning.

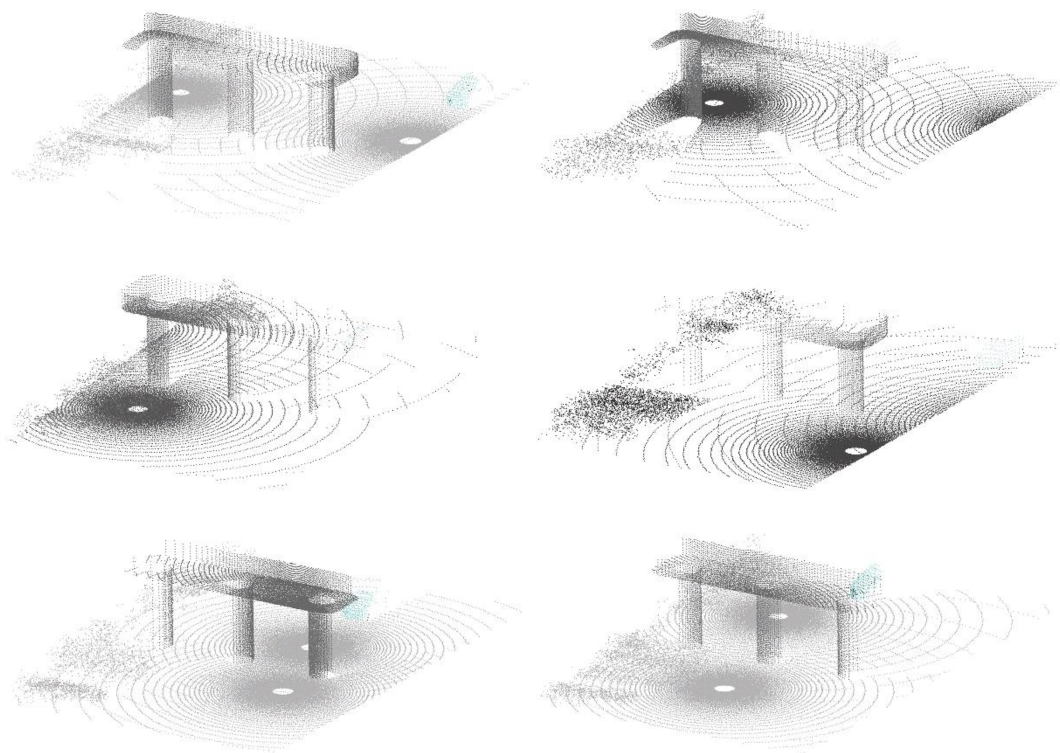


Fig. 5-10. Examples of generated synthetic point clouds for the validation experiment

5.3.4 Machine learning algorithm

The generated point clouds have been used for training the PointNet network [222]. The network input features were coordinates of 2048 points, and the batch size was 32. The network has been trained in 100 epochs with an initial learning rate of 10^{-3} (decaying by half every 20 epochs, as suggested by authors). The trained model has been validated on batches of real and synthetic data (Table 5-2).

Table 5-2. Performance of the trained PointNet model on a) real and b) synthetic validation batches and c) their comparison

a) real batch performance	*Precision*	*Recall*	*f-1 score*
ColumnPier	96.34%	91.11%	93.65%
ColumnCap	95.88%	97.13%	96.50%
Car	99.72%	76.18%	86.37%
Terrain	99.42%	99.97%	99.69%

Average	97.84%	91.10%	94.06%
b) synthetic batch performance	*Precision*	*Recall*	*f-1 score*
ColumnPier	94.60%	93.82%	94.21%
ColumnCap	98.13%	98.64%	98.38%
Car	99.39%	81.66%	89.66%
Terrain	99.62%	99.76%	99.69%
Average	97.94%	93.47%	95.49%
c) comparison (|real – synthetic|)	*Precision*	*Recall*	*f-1 score*
ColumnPier	1.74%	2.71%	0.56%
ColumnCap	2.25%	1.51%	1.88%
Car	0.33%	5.48%	3.28%
Terrain	0.21%	0.21%	0.00%
Average	0.10%	2.37%	1.43%

The model achieved an average f-1 score of 94.06% on real and 95.49 % on synthetic dataset, a relatively high overall performance for a semantic segmentation task. The performance is affected by imbalanced data (most instances are *Terrain*) and particularly one-scene specialization; similar performance is much harder achievable for a “general” segmentation task when a model operates on new scenes and object instances. However, as the purpose of this validation was not to evaluate the model but the data, I will discuss not the absolute performance but the relation between performance on synthetic and real data.

The synthetic-data-trained PointNet achieved comparable performance on both synthetic and real validation datasets (Table 5-2c). The biggest difference is in *Car* class recall (5.48%). The *Car* class is particularly demanding, given its scarcity and distribution difference in real and synthetic datasets – real scanners’ locations provided good exposure for a car, which was not always the case for randomly located synthetic scanners. Performance metrics ratios for other classes are comparable.

In addition to performance values, the model also *behaved* similarly on real and synthetic data. Fig. 5-11 presents confusion matrices for inferences on a whole real and synthetic point cloud (an example of a synthetic validation dataset). The patterns of incorrect predictions tend to be similar, e.g., part of *Car* instances predicted as *Terrain,* part of lower level *ColumnPier* instances predicted as *Terrain,* and higher level ones – as *ColumnCap*. The similarity of performance and *behavior* on synthetic and real data of a synthetic-data-trained model gives a premise for the successful usage of DynamoPCSim synthetic point clouds for machine learning.

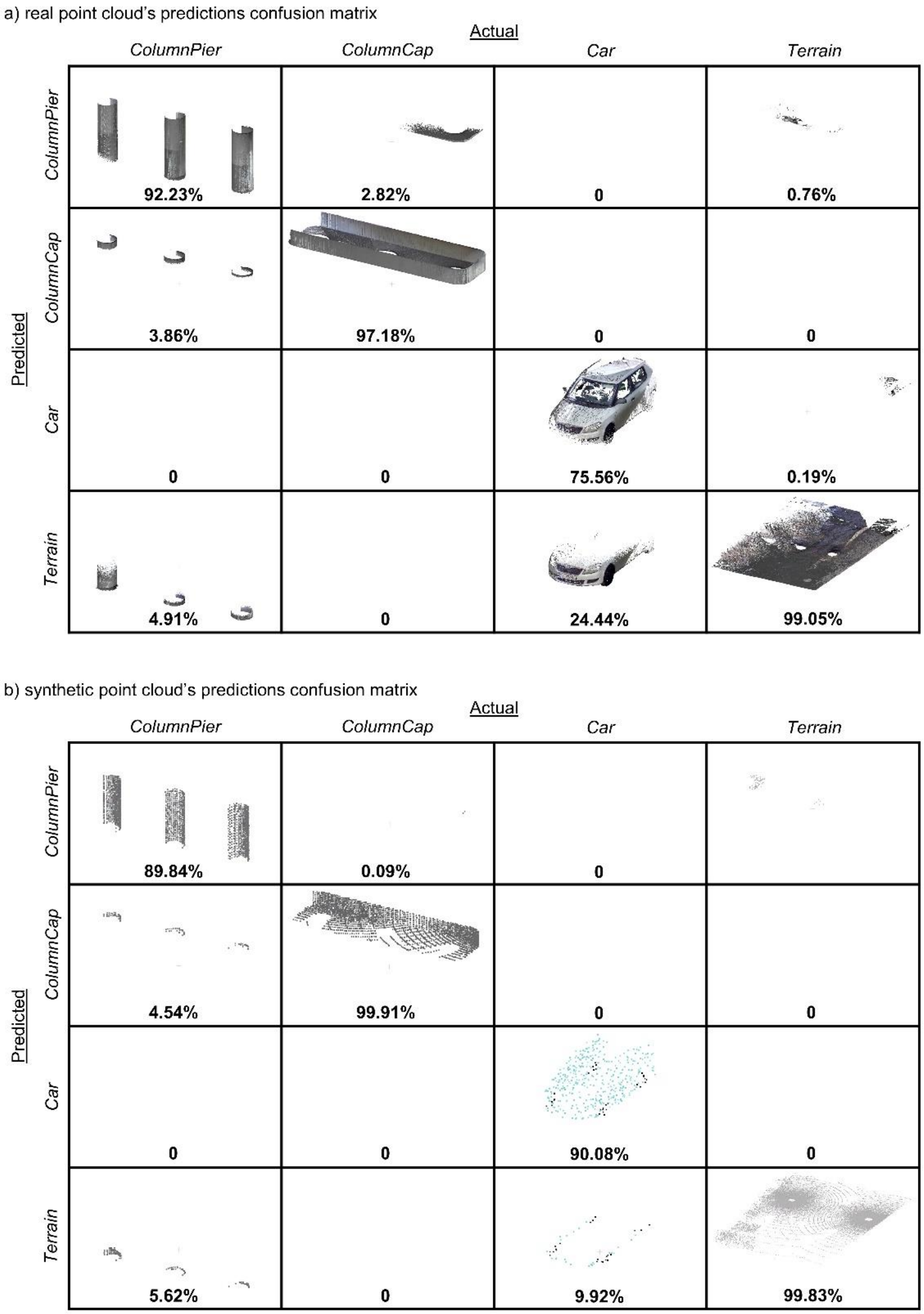


Fig. 5-11. Confusion matrices (with graphical outcomes) of predictions on a) the real point cloud and b) a synthetic point cloud

5.4 Conclusions

The area of point clouds is fastly developing with new hardware tools, processing methods, and use cases. Point clouds fulfill the need for reliable, actual geometrical representations, constituting a basis for civil engineering digital twins. Machine learning is becoming an automation solution, also in AEC, which, like every other industry, needs to increase its effectiveness to keep up with the nowadays' needs. The plethora of point clouds' use cases shows potential for automation.

Point clouds, as structured data containers, are a proper subject of machine learning algorithms. The algorithms can enhance point clouds' processing, interpretation, and utilization. However, for proper training resulting in effective operation, machine learning algorithms need data. Difficulties in acquiring big, diversified datasets of real point clouds incline into synthetic data.

DynamoPCSim, a solution created by the author, enables the generation of synthetic point clouds. It simulates real scanners' operation with ray tracing and operates on geometric and BIM models, producing labeled point clouds. It can be customized to a task, utilizing its parametricity and modular structure. A performed experiment of semantic segmentation using a neural network confirmed the usability of DynamoPCSim-produced data for machine learning.

The here-proposed system focuses on generating the proper data for training the machine learning algorithms. Although indispensable, this is only a first step into machine learning systems automating the utilization of point clouds. The described solution does not implement the whole workflow to extract point clouds' information, segment them, and finally generate and update point-cloud-based geometrical models of digital twins. In this idea, the automatically updated models will match the real conditions of the physical objects, realizing the vision of a model coexisting with an object through the whole lifecycle, also in the operational phase. It will open new analysis possibilities.

Implementing a complex system for updating and analyzing geometrical models of digital twins can be considered a future work. The systems of "Scan-to-BIM" or "as-built generation" are an area of extensive research. Usually, they take point clouds as inputs, and their task is to generate a BIM model. However, due to the diversity of elements and systems in structures, as well as the diversity of their applications, no method has yet been able to generalize on creating semantically-rich BIMs of different types of objects [227]. Additionally, to build a BIM model, not only geometry is needed, but also semantics. It cannot be derived from point clouds. With the difficulties in creating a system generalizing well on different types of objects, the author conceptualizes a facility-dedicated approach (described in the future work section, 6.2.3).

6 Conclusions

6.1 Summary

The ongoing digitization presents both challenges and opportunities. Industries are transforming at an unprecedented pace. They also have become interconnected, which requires a new paradigm of cooperation. Digital twinning, announced as a crucial component of Industry 4.0, becomes an enabler of holistic digital transformation. As a relatively new concept, digital twins constantly evolve with new principles and use cases in various industries. This process faces challenges of standardization, education, and implementation costs. However, advances in computing, modeling, or sensing transformed digital twining from an academic-only technological utopia to a tool in industrial practice.

The modern world states challenges for civil engineering. Sustainable development goals, economic aims, and a growing population motivate the introduction of modern technologies. The need for transformation is expressed by the problems of aging infrastructure, the margin of civil engineering productivity, and practitioners' expectations. Bridges, as complex, unique structures of logistic and strategic importance, must adapt to fulfill the demands and benefit from technological advances.

The concept of digital twins, the virtual counterparts of objects through the entire lifecycle, is gaining momentum with increasing academic and industrial interest, also in civil engineering. The applications are not yet comprehensive and focus rather on particular components. These partial applications are beneficial, pushing the utilization of modern techniques. Nonetheless, more activities are still needed to widen the idea of extensive digitization. Bridges require suited digital twinning frameworks and novel use cases providing coherent benefits. Applicability is crucial in crafting solutions to be used not only in academia but also in industrial practice; to develop, digital twinning must be adopted by practitioners. Therefore, bridge digital twinning framework should promote the evolutionary approach. Civil engineering digital twinning should not be a forced revolution but a natural adoption of beneficial techniques. The framework should, therefore, rather guide into practical benefits than restrain with idealized rules.

The proposed in this dissertation concept of a digital twin of a bridge is one of the first approaches to establish the foundations for these data-rich models of bridges. The dissertation identifies techniques for creating and utilizing digital twins of bridges and proposes their characteristics. The proposed concept addresses the general multi-industrial digital twinning fundamentals (identified with the literature study), the specificity of bridges, and the demands of designers, engineers, and managers of civil engineering facilities. The proposed digital twin is a virtual counterpart of the bridge in its entire lifecycle characterized by actuality, intelligence and autonomy, interaction, interoperability, modularity, expansibility and scalability, accessibility and security, and uniqueness. Actuality, intelligence and autonomy, as well as interaction, are the foundation of the digital twinning concept. Interaction (the ability to cooperate with other objects in the digital dimension) is enabled by interoperability (the compatibility with general principles and standards of data exchange). Modularity enables practical implementations; not all the functionalities of digital twins must be implemented from the very beginning. The expansibility of functionalities and scalability are the results of previous characteristics. Accessibility for users and other instances, but with the security of data, as well as uniqueness (e.g., of identifiers), are the indispensable factors for effective implementations.

From the technological perspective, it is an evolution of current practices, which have not yet been properly integrated. The proposed digital twin utilizes BIM (Building Information Modeling), SHM (Structural Health Monitoring), and AI (Artificial Intelligence). It states IFC (Industry Foundation Classes) as the base for the central model. It is also enhanced by other techniques (e.g., visual programming, point clouds). This approach enables the evolutionary process of adopting digital twins in civil engineering. Civil engineering digital twinning does not have to be perfect from the start. It should mature naturally, alluring practitioners with its benefits.

To make this evolutionary development of digital twins possible, digital twinning must provide practical use cases giving clear benefits to engineers and managers of facilities. The benefits are, among others, the automation of the design, assessment of the structural condition, and effective data management. These benefits are realized by new digital twins' use cases. Two of such approaches have been proposed in the dissertation.

The first proposed technique is the automation of design using visual programming and genetic algorithm [183]. The solution presents how visual programming can integrate BIM and FEM Models. Such integration is crucial for creating fully-functional digital twins of bridges. Moreover, the implemented optimization algorithm allowed for analyzing much more variants than in the traditional design. It leads to optimal structural choices and enhances the effectiveness of the process.

The second proposed solution regards creating and updating geometries of digital twins also in the building and operational phases. It is a technique to generate synthetic point clouds as datasets to train machine learning algorithms. The trained systems can extract point clouds' information, segment them, and finally, also generate and update the point-cloud-based geometrical models of digital twins. The automatically updated models will match the real conditions of the physical objects, realizing the idea of a model coexisting with an object through the whole lifecycle, also in the operational phase. As for bridges, the information about geometrical changes will allow to detect and compare deformations, monitor the structure, and alarm about hazards. Such a complex system is a consideration for future work (section 6.2.3).

6.2 Limitations and future work

6.2.1 Bridge digital twinning framework

The here-proposed techniques, notions, and characteristics base foundations for the digital twinning of bridges. They guide adoption but do not answer all the questions of practical implementations. It leads to the identification of future work paths.

The first is specificity. Future research should aim to concretize the recommendations in digital twinning characteristics. What are the expected synchronization rates for various sources of data? Which intelligent algorithms are optimal for which tasks? Which systems' infrastructure solutions (e.g., databases, cloud providers) are effective? How to ensure cybersecurity? Digital twinning encompasses wide multi-domain areas – every characteristic and module requires focused research. Eventually, the maturity of digital twinning will lead to standardization, which is another future work path requiring comprehensive global multi-industrial cooperation.

The next future work path is more AEC-specific. Industry Foundation Classes is proposed as the central model schema of the civil engineering digital twin. IFC documentation is still forming with new standardization concepts and new interoperability opportunities. Nonetheless, in practice, IFC capabilities are still used in limited form. Digital twin comprises various data types, so guidelines for coding the data beyond geometry and objects' semantics (with IFC data schema itself and as linked sources) should be recognized.

Digital twinning is evolving also due to dynamic technical advances. Civil engineering will potentially benefit from emerging (e.g., virtual, mixed, and augmented reality) and new solutions. Future research will need to address the adoption of the techniques.

6.2.2 Optimization using visual programming and genetic algorithm

The visual programming-genetic algorithm tandem is evaluated as effective for geometric optimization regarding basic variables. Nonetheless, the variables should address additional factors, especially with the demands of economic and social costs in the entire lifecycle.

Though the solution includes buckling as an example of implementing design effect rules, it neglects several other factors: the composite character of the structure, dynamic response, and second-order effects. For norms-compatible analyses, also the load cases must be enhanced. The algorithm-applied loads caused almost full use of the reference structure, but it was designed for additional factors (temperature, wind, snow, special loads), unlike the system-optimized solutions. Therefore, their final *MaterialCost* values cannot be directly compared to assess the actual savings but rather to evaluate the system performance. These limitations, together with mentioned upgrades, are potential future-steps aims. Also, the performance of the other optimization algorithms (particle swarm optimization, harmony search optimization, simulated annealing) might be investigated.

6.2.3 Generating synthetic data for point clouds' machine learning automation

DynamoPCSim, a synthetic point clouds' simulator, still has a wide field for development. The future work paths come from expanding features, increasing compatibility and computing efficiency, and adapting to new practices:

- IFC compliance

  DynamoPCSim operates on geometrical models of Dynamo (an independent, open-source application) or Revit BIM models (due to possible linkage of Dynamo and the software). Adding the IFC compliance would straightforward the usage of various-sources models for simulation (currently, IFC models can be imported to Revit and used as such, but the import may not address all the model details). Utilizing IFC models is a positive trend enhancing the OpenBIM idea.

- More efficient computing

  DynamoPCSim depends on the Dynamo computation force, affected by the computation power of the computer it runs on. It enables generating machine learning datasets on a typical personal-use computer, but the complexity of a scanned model and higher scanning resolution harm the efficiency. Potential independence from software and local machine computing power can be acquired by implementing the simulator as a web application. As such, it would utilize cloud computing and assure wide software-independent access.
- Point clouds' points additional properties gathering

  DynamoPCSim allows attaching user-specified semantic data to point clouds' points (e.g., color, material properties), as well as calculating properties like normals and intensity. However, due to insufficient data in real scanners standardization and complex calculations model, the intensity calculation is currently implemented in a simplified form. Although this form may be useful, more accurate calculations would undoubtedly increase the intensity usefulness as machine learning algorithms feature. Also, other currently not implemented properties may be valuable.
- Additional transformation strategies

  DynamoPCSim enables point clouds' transformations like introducing noise, adding, translating, rotating, or deleting points, and altering their properties. However, potential future use for various tasks may reveal the need for additional transformations.
- Adapting to new devices and collecting practices

  Point clouds are a dynamic field, with new tools being constantly introduced. DynamoPCSims simulates popular terrestrial laser scanners, but new devices like mobile handheld and backpack scanners have already started to be used in practice. To adapt to new practices, the simulator will require the ability to simulate modern tools.

The proposed system focuses on generating the proper data for training the machine learning algorithms. Although indispensable, this is only a first step into machine learning systems automating the utilization of point clouds. The solution does not implement the whole workflow to extract point clouds' information, segment them, and finally generate and update point-cloud-based geometrical models of digital twins. In this idea, the automatically updated models will match the real conditions of the physical objects, realizing the vision of a model coexisting with an object through the whole lifecycle, also in the operational phase. It will open new analysis possibilities.

Implementing a complex system for updating and analyzing geometrical models of digital twins can be considered a future work. The systems of "Scan-to-BIM" or "as-built generation" are an area of extensive research. Usually, they take point clouds as inputs, and their task is to generate a BIM model. However, due to the diversity of elements and systems in structures, as well as the diversity of their applications, no method has yet been able to generalize on creating semantically-rich BIMs of different types of objects [227]. Additionally, to build a BIM model, not only geometry is needed, but also semantics. It cannot be derived from point clouds.

With the difficulties in creating a system generalizing well on different types of objects, the author conceptualizes a facility-dedicated approach. The approach assumes not generating the models from scratch but rather updating the models created during design. The parametrized BIM model created during design can be used for generating synthetic point clouds (using DynamoPCSim) for the designed facility. The data will be used to train machine learning algorithms for extracting geometrical parameters from point clouds. Because the algorithm will be dedicated to one facility, its task will be easier compared to algorithms working on point clouds of different objects. The set of parameters will be known; the algorithm task will be limited to determine their values. Also, the one-facility point clouds are much less varying (e.g., the elements' positions are relatively fixed). The algorithms trained on synthetic point clouds will take real point clouds as input and update the parameters of the BIM model. The scanning of a real facility can be performed on a regular schedule, leading to a BIM-model-based history of geometrical changes. The regularly updated BIM model will contain semantical data and vectorized geometries. The vectorized geometries are much easier to compare than raw point clouds. Monitoring, identifying, and comparing the geometrical changes will allow for the detection of structural malfunctions. As for bridges, monitoring the geometrical changes is crucial, especially for objects on the ground deformation areas (e.g., mining or post-seismic activities, tunneling). The model updating can therefore lead to expert systems alarming about hazards.

6.3 Concluding remarks

The ongoing digital transformation is undoubtedly a challenge – but also an opportunity. Given the increasingly complex interconnections of scientific and industrial activities of various sectors, digital twinning improves the effectiveness of civil engineering and enables interdisciplinary cooperation. It also delivers new, practical tools, assisting designers, engineers, and managers in day-to-day tasks. The here-proposed principles and solutions for digital twins of bridges are a brick towards the infrastructure of the future: reaching the modern world's goals and delivering everyday value.

The conducted analysis of the state-of-the-art in the field of digital twins of bridges, conceptual work, and in-situ experiments allowed for achieving the aims stated at the beginning of the dissertation. The foundations for the creation of digital twins of bridges have been declared by proposing a framework that takes into account the specificity of bridges and their lifecycle. It allows for the virtualization of not only physical objects but also their accompanying processes. The author is aware that the presented material does not fulfill all the assumptions of the digital twins' concept. However, the presented research and the resulting acquired competencies are the catalysts for further activities that will aim at increasingly complete implementations of digital twins of bridges in the future.

Appendix A. Digital twinning overview bridge-oriented articles

Table A-1 Digital twinning bridge-oriented publications

Scopus database; access: 15.01.2023;
code: TITLE ((“digital twin” OR “digital twins” OR “digital twinning”) AND (bridge OR bridges))

Authors	*Title*	*Year*	*Document Type*	*Citations count*
Shim C.-S., Dang N.-S., Lon S., Jeon C.-H.	Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model	2019	Article	76
Lu R., Brilakis I.	Digital twinning of existing reinforced concrete bridges from labelled point clusters	2019	Article	81
Ye C., Butler L., Calka B., et.al.	A digital twin of bridges for structural health monitoring	2019	Conference Paper	34
Shim C.S., Kang H.R., Dang N.S.	Digital twin models for maintenance of cable-supported bridges	2019	Conference Paper	13
Dang N.S., Shim C.S.	Bridge assessment for PSC girder bridge using digital twins model	2020	Book Chapter	11
Cervenka J., Jendele L., Zalsky J., et al.	Digital twin approach for durability and reliability assessment of bridges	2020	Conference Paper	0
Sofia H., Anas E., Faiz O.	Mobile mapping, machine learning and digital twin for road infrastructure monitoring and maintenance: Case study of mohammed VI bridge in Morocco	2020	Conference Paper	19
Jiang F., Ding Y., Song Y., Geng F., Wang Z.	An Architecture of Lifecycle Fatigue Management of Steel Bridges Driven by Digital Twin	2021	Article	3
Kaewunruen S., Sresakoolchai J., Ma W., Phil-Ebosie O.	Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions	2021	Article	29
Lin K., Xu Y.-L., Lu X., Guan Z., Li J.	Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes	2021	Article	30
Jiang F., Ding Y., Song Y., Geng F., Wang Z.	Digital Twin-driven framework for fatigue life prediction of steel bridges using a probabilistic multiscale model: Application to segmental orthotropic steel deck specimen	2021	Article	18
Kang J.-S., Chung K., Hong E.J.	Multimedia knowledge-based bridge health monitoring using digital twin	2021	Article	19

Authors	*Title*	*Year*	*Document Type*	*Citations count*
Mohammadi M., Rashidi M., Mousavi V., Karami A., Yu Y., Samali B.	Quality evaluation of digital twins generated based on uav photogrammetry and tls: Bridge case study	2021	Article	22
Schacht G., Wenner M., Marx S.	The digital bridge twin [Der digitale brückenzwilling]	2021	Article	0
Moradi S., Eftekhar Azam S., Mofid M.	A Physics Informed Neural Network Integrated Digital Twin for Monitoring of the Bridges	2021	Conference Paper	0
Ai L., Bayat M., Comert G., Ziehl P.	An Autonomous Bridge Load Rating Framework Using Digital Twin	2021	Conference Paper	0
Mohammadi M., Rashidi M., Mousavi V., et al.	Case study on accuracy comparison of digital twins developed for a heritage bridge via UAV photogrammetry and terrestrial laser scanning	2021	Conference Paper	6
Mafipour M.S., Vilgertshofer S., Borrmann A.	Deriving Digital Twin Models of Existing Bridges from Point Cloud Data Using Parametric Models and Metaheuristic Algorithms	2021	Conference Paper	1
Gunner S., Voyagaki E., Gavriel G., et al.	Digital Twins for civil engineering: the Clifton Suspension Bridge (UK)	2021	Conference Paper	1
de Freitas Bello V.S., Popescu C., et al.	Framework for facility management of bridge structures using digital twins	2021	Conference Paper	0
Adibfar A., Costin A.M.	Integrated Management of Bridge Infrastructure through Bridge Digital Twins: A Preliminary Case Study	2021	Conference Paper	0
Kazemian M., Nikdel S., Mohammadesmaeili M., et al.	Kalix Bridge Digital Twin—Structural Loads from Future Extreme Climate Events	2021	Conference Paper	0
May I., Rösner C.	AI-based generation of 3-D digital twins of existing bridges [Skalierbare Technologie zur Erstellung von 3-D Digital Twins für Bestandsbrücken]	2022	Article	0
Lorenzen S.R., Berthold H., Rupp M., et al.	Artificial intelligence, digital twins and the future of bridge management [Künstliche Intelligenz, Digitale Zwillinge und die Zukunft des Brückenmanagements]	2022	Article	0
Ghahari F., Malekghaini N.,	Bridge Digital Twinning Using an Output-Only Bayesian Model Updating Method and Recorded Seismic Measurements	2022	Article	3

Authors	*Title*	*Year*	*Document Type*	*Citations count*
Ebrahimian H., Taciroglu E.				
Adibfar A., Costin A.M.	Creation of a Mock-up Bridge Digital Twin by Fusing Intelligent Transportation Systems (ITS) Data into Bridge Information Model (BrIM)	2022	Article	1
Zhao H., Tan C., Obrien E.J., Zhang B., Uddin N., Guo H.	Developing Digital Twins to Characterize Bridge Behavior Using Measurements Taken under Random Traffic	2022	Article	4
Dan D., Ying Y., Ge L.	Digital Twin System of Bridges Group Based on Machine Vision Fusion Monitoring of Bridge Traffic Load	2022	Article	6
Braml T., Wimmer J., Varabei Y., et al.	Digital twin: Asset administration shell BBox as data storage over the life cycle of a bridge [Digitaler Zwilling: Verwaltungsschale BBox als Datenablage über den Lebenszyklus einer Brücke]	2022	Article	3
Lin K., Xu Y.-L., Lu X., Guan Z., Li J.	Digital twin-based life-cycle seismic performance assessment of a long-span cable-stayed bridge	2022	Article	0
Yu S., Li D., Ou J.	Digital twin-based structure health hybrid monitoring and fatigue evaluation of orthotropic steel deck in cable-stayed bridge	2022	Article	0
Jiang F., Ding Y., Song Y., Geng F., Wang Z.	Digital Twin-driven framework for fatigue lifecycle management of steel bridges	2022	Article	1
Yoon S., Lee S., Kye S., Kim I.-H., Jung H.-J., Spencer B.F., Jr	Seismic fragility analysis of deteriorated bridge structures employing a UAV inspection-based updated digital twin	2022	Article	0
Rojas-Mercedes N., Erazo K., Di Sarno L.	Seismic fragility curves for a concrete bridge using structural health monitoring and digital twins	2022	Article	2
Rageh A., Azam S.E., Alomari Q., Linzell D., Wood R.	Model Updating and Parameter Identification for Developing Digital Twins for Riveted Steel Railway Bridges	2022	Book Chapter	2
Zhou C., Xiao D., Hu J., Yang Y., et al.	An Example of Digital Twins for Bridge Monitoring and Maintenance: Preliminary Results	2022	Conference Paper	0
Taraben J., Helmrich M., Morgenthal G.	Bridge Condition Assessment Based on Image Data and Digital Twins	2022	Conference Paper	0

Authors	*Title*	*Year*	*Document Type*	*Citations count*
Giorgadze I.M., Vahdatikhaki F., Voordijk J.H.	Conceptual Modeling of Lifecycle Digital Twin Architecture for Bridges: A Data Structure Approach	2022	Conference Paper	0
Mafipour M.S., Vilgertshofer S., Borrmann A.	Creating digital twins of existing bridges through AI-based methods	2022	Conference Paper	0
Ying G., Zhang C., Hu J., Chen W., et al.	Design of a bridge digital twin system for Intelligent operation and maintenance based on machine vision	2022	Conference Paper	0
Hidayat F., Supangkat S.H., Hanafi K.	Digital Twin of Road and Bridge Construction Monitoring and Maintenance	2022	Conference Paper	0
Saback de Freitas Bello V., Popescu C., et al.	Framework for Bridge Management Systems (BMS) Using Digital Twins	2022	Conference Paper	1
Wenzel B., Möller E., Schmid B., Weber C., Morgenthal G.	The New Little Belt Bridge - the role of the physical model and it's digital twin	2022	Conference Paper	0
Ramonell C., Chacón R.	Towards Automated Pipelines for Processing Load Test Data on a HS Railway Bridge in Spain using a Digital Twin	2022	Conference Paper	0
Futai M.M., Bittencourt T.N., Santos R.R., et al.	Utilization of Digital Twins for Bridge Inspection, Monitoring and Maintenance	2022	Conference Paper	2
Hosamo H.H., Hosamo M.H.	Digital Twin Technology for Bridge Maintenance using 3D Laser Scanning: A Review	2022	Review	2

Appendix B. Modeling IFC

Listing B-1. Modeling IFC example

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION ((''), '2;1');
FILE_NAME ('Bridge model example', '2023-01-14T17:42:13', 'Kamil Korus',
'Silesian University of Technology', 'Processor version 5.1.0.0', 'Xbim.Common', '');
FILE_SCHEMA (('IFC4X3_ADD1'));
ENDSEC;

DATA;
#1=IFCPROJECT('0vGCH5MKb7IxXhcP$Yz3$a',$,'Bridge project',
'This project is a simple example of modeling a bridge',$,$,$,(#14,#17),#2);

#2=IFCUNITASSIGNMENT((#3,#4,#5,#6,#7,#8,#9,#10,#11));
#3=IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#4=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#5=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#6=IFCSIUNIT(*,.SOLIDANGLEUNIT.,$,.STERADIAN.);
#7=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#8=IFCSIUNIT(*,.MASSUNIT.,$,.GRAM.);
#9=IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
#10=IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
#11=IFCSIUNIT(*,.LUMINOUSINTENSITYUNIT.,$,.LUMEN.);

#12=IFCCARTESIANPOINT((0.,0.,0.));
#13=IFCAXIS2PLACEMENT3D(#12,$,$);
#14=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.E-05,#13,$);
#15=IFCCARTESIANPOINT((0.,0.));
#16=IFCAXIS2PLACEMENT2D(#15,$);
#17=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Plan',2,1.E-05,#16,$);

#18=IFCLOCALPLACEMENT($,#13);
#19=IFCDIRECTION((1.,0.,0.));
#20=IFCDIRECTION((0.,1.,0.));
#21=IFCDIRECTION((0.,0.,1.));

#22=IFCFACILITY('3eRo4UYs96VvgzHYlN5rwP',$,'Facility',$,$,$,$,$,$);
#23=IFCRELAGGREGATES('1zjaZZpoz2EwQTKxCE3dQj',$,$,$,#1,(#22));

#24=IFCBRIDGE('2pyXrKuJX2hROBEU2KQ1rq',$,'Bridge',$,$,$,$,$,$,.GIRDER.);
#25=IFCRELCONTAINEDINSPATIALSTRUCTURE('2PeVAh2uH2phENraW6lvHj',$,$,$,(#24),#22);

#26=IFCBRIDGEPART('2N8pyojwb4Bgw07KZBEF3O',$,'Superstructure',$,$,#36,#55,$,$,.USERDEFINED.,
.SUPERSTRUCTURE.);
```

```
#27=IFCRELCONTAINEDINSPATIALSTRUCTURE('1LqnXVBdfEMwYGj9_UUshh',$,$,$,(#26,#30),#24);

#28=IFCMATERIAL('Concrete',$,$);
#29=IFCRELASSOCIATESMATERIAL('295_zxIQT9QBPS_FCpe59Y',$,$,$,(#26),#28);

#30=IFCALIGNMENT('2eDu6iXCn6qfh2qIv8t0P2',$,$,$,$,#18,#35,$);
#31=IFCPOLYLINE((#32,#33));
#32=IFCCARTESIANPOINT((0.,0.,0.));
#33=IFCCARTESIANPOINT((20000.,0.,400.));
#34=IFCSHAPEREPRESENTATION(#14,$,'Curve3D',(#31));
#35=IFCPRODUCTDEFINITIONSHAPE('Alignment shape definition',$,(#34));

#36=IFCLOCALPLACEMENT($,#37);
#37=IFCAXIS2PLACEMENT3D(#38,$,$);
#38=IFCCARTESIANPOINT((0.,0.,-80.));

#39=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#40);
#40=IFCPOLYLINE((#41,#42,#43,#44,#45,#46,#47,#48,#49,#50,#51,#41));
#41=IFCCARTESIANPOINT((0.,0.));
#42=IFCCARTESIANPOINT((-3800.,-76.));
#43=IFCCARTESIANPOINT((-6750.,-17.));
#44=IFCCARTESIANPOINT((-6750.,-237.));
#45=IFCCARTESIANPOINT((-4300.,-356.));
#46=IFCCARTESIANPOINT((-3900.,-1100.));
#47=IFCCARTESIANPOINT((3900.,-1100.));
#48=IFCCARTESIANPOINT((4300.,-356.));
#49=IFCCARTESIANPOINT((6750.,-237.));
#50=IFCCARTESIANPOINT((6750.,-17.));
#51=IFCCARTESIANPOINT((3800.,-76.));

#53=IFCFIXEDREFERENCESWEPTAREASOLID(#39,$,#31,$,$,$);
#54=IFCSHAPEREPRESENTATION(#14,'Body','SolidModel',(#53));
#55=IFCPRODUCTDEFINITIONSHAPE('Superstructure shape definition',$,(#54));

#56=IFCPERFORMANCEHISTORY('2AKe$ZJ1bCxunXTKWapVSF',$,
'Superstructure performance history',$,$,$,'OPERATION',$);
#57=IFCRELAGGREGATES('0dSmTizeDBMREGS$lvAfsh',$,$,$,#26,(#56));
#58=IFCLIBRARYINFORMATION('Superstructure performance database',$,$,$,
'https://www.digitaltwin-database.com/0vGCH5MKb7IxXhcP$Yz3$a/2N8pyojwb4Bgw07KZBEF3O',$);
#59=IFCRELASSOCIATESLIBRARY('01mMvMoyvFJwH8aYXLKCjV',$,$,$,(#56),#58);


ENDSEC;
END-ISO-10303-21;
```

Appendix C. Results of the optimization using visual programming and genetic algorithm

Table C-1. Geometry optimization final parameters and outputs

Parameter	*Vertical (0)*	*Radial (1)*	*Oblique (2)*	*Network (3)*
Arch rise, m	14.3	16.5	16.0	16.6
Arch inclination, °	-11	-15	-13	-14
Hangers number	10	9	18	14
Braces number	4	4	4	4
Arch use	0.862	0.843	0.838	0.525
Girder use	0.785	0.735	0.997	0.515
Crossbar use	0.996	0.994	0.907	0.962
External crossbar use	0.986	0.753	0.907	0.831
Brace use	0.228	0.192	0.177	0.180
Hanger use	0.207	0.271	0.245	0.636
MaterialCost	81.31	80.62	82.71	81.28

Table C-2. Geometry and cross-sections optimization final parameters and outputs

Parameter	*Vertical (0)*	*Radial (1)*	*Oblique (2)*	*Network (3)*
Arch rise, m	18.4	17.4	18.4	18.9
Arch inclination, °	-7	-2	-5	-3
Hangers number	9	7	16	12
Braces number	4	4	4	4
h_arch, mm	1300	1000	900	800
b_arch, mm	1300	900	700	500
tf_arch, mm	30	30	30	20
tw_arch, mm	20	20	20	10
h_girder, mm	1100	1600	900	1500
b_girder, mm	400	500	200	600
tf_girder, mm	20	30	20	10
tw_girder, mm	12	12	12	12
h_crossbar, mm	1600	1700	1200	1200
b_crossbar, mm	400	300	1100	900
tf_crossbar, mm	40	50	30	30

tw_crossbar, mm	12	16	12	12
h_crossbarExt, mm	1100	900	1000	1500
b_crossbarExt, mm	1100	800	1200	1300
tf_crossbarExt, mm	20	20	40	10
tw_crossbarExt, mm	20	20	20	10
d_brace, mm	300	400	300	300
t_brace, mm	12	20	12	12
d_hanger, mm	30	30	50	60
Arch use	0.970	0.909	0.987	0.982
Girder use	0.975	0.954	0.853	0.920
Crossbar use	0.985	0.977	0.887	0.996
External crossbar use	0.993	0.913	0.997	0.919
Brace use	0.479	0.378	0.391	0.246
Hanger use	0.588	0.659	0.381	0.772
MaterialCost	58.58	59.13	48.51	38.92

Table C-3. Geometry, cross-sections, and materials optimization final parameters and outputs

Parameter	*Vertical (0)*	*Radial (1)*	*Oblique (2)*	*Network (3)*
Arch rise, m	20.9	23.2	18.6	16.9
Arch inclination, °	-2	1	0	-8
Hangers number	10	7	14	12
Braces number	4	4	4	4
h_arch, mm	1600	1400	800	1100
b_arch, mm	1000	900	1100	700
tf_arch, mm	40	40	30	30
tw_arch, mm	10	10	10	10
h_girder, mm	1500	1300	1000	1500
b_girder, mm	500	500	700	400
tf_girder, mm	30	40	20	20
tw_girder, mm	12	12	12	12
h_crossbar, mm	1700	1500	1400	1500
b_crossbar, mm	600	700	800	1100
tf_crossbar, mm	30	30	40	20
tw_crossbar, mm	16	12	12	12

h_crossbarExt, mm	1300	1100	1800	1300
b_crossbarExt, mm	1600	1300	1200	1100
tf_crossbarExt, mm	20	20	20	30
tw_crossbarExt, mm	20	10	20	20
d_brace, mm	200	200	300	200
t_brace, mm	12	12	12	12
d_hanger, mm	50	40	60	50
Arch material	S275	S275	S275	S275
Girder material	S275	S275	S235	S235
Crossbar material	S275	S355	S275	S355
External crossbar material	S235	S235	S275	S275
Brace material	S235	S235	S235	S235
Arch use	0.949	0.947	0.945	0.995
Girder use	0.932	0.930	0.951	0.858
Crossbar use	0.982	0.998	0.966	0.999
External crossbar use	0.958	0.997	0.876	0.827
Brace use	0.690	0.791	0.521	0.596
Hanger use	0.237	0.308	0.294	0.883
MaterialCost	55.48	53.50	46.49	37.94

Appendix D. DynamoPCSim methods

Table D-1, Table D-2, and Table D-3 list the main classes of DynamoPCSim and their main methods. *Inputs* and *Outputs* columns present both methods' input parameters and outputs of their operations, as well as their data types; these can be simple types (e.g., *integer*, *double*, or *string*) or complex classes (e.g., *Geometry*, *MeshElement*). Square brackets (*[]*) at the end of a type name indicate a list of that type instances; two square brackets (*[][]*) indicate a nested list (list of lists). Parentheses (*()*) indicate the default value of an input, if exists. Some classes inherit from base classes (e.g., *AddNoisePoints* inherits from *PointCloudTransformationStrategy*); methods of these *child* classes are listed in the *parenting* classes sections.

Table D-1. DynamoPCSim *Mesh* main classes and methods

class: **Mesh**			
Method	*Inputs*	*Outputs*	*Description*
ByElements	elements : *MeshElement[]*	Mesh : *Mesh*	Creates a Mesh from MeshElements.
ByTriangles	triangles: *MeshTriangle[]*	Mesh : *Mesh*	Creates a Mesh from MeshTriangles.
FromJSON	filePath : *string*	Mesh : *Mesh*	Retrieves a Mesh from a JSON file.
FilteredByAllLabels	mesh : *Mesh*	Meshes: *Mesh[]*	Splits the Mesh into multiple single-label Meshes (containing MeshTriangles of only one label).
FilteredByLabel	mesh : *Mesh* label : *string*	Mesh : *Mesh*	Creates a single-label Mesh with MeshTriangles of only declared label.
GetRayIntersection	mesh : *Mesh* ray : *Ray*	PointCloudPoint : *PointCloudPoint*	Get the point resulting from the intersection of a Ray and Mesh. The same operation for multiple rays is performed during scanning.
SaveToJSON	mesh : *Mesh* filePath : *string*	-	Saves the Mesh into JSON file.
Visualize	mesh : *Mesh*	PolyCurves: *PolyCurve[][]*	Visualizes the Mesh as polycurves representing MeshTriangles. For Each Mesh's MeshElement, a separate list of PolyCurves is created.
Labels	mesh : *Mesh*	Labels : *string[]*	Gives a list of all distinct labels of the Mesh.
Triangles	mesh : *Mesh*	MeshTriangles : *MeshTriangle[]*	Gives a list of all MeshTriangles of the Mesh.
class: **MeshElement**			

Method	*Inputs*	*Outputs*	*Description*
ByElement	element : *ModelElement* label : *string* ("unknown") additionalProperties : *MeshAdditionalProperty[ ]* (empty list) additionalPropertiesStrategies : *MeshElementAdditionalPropertyStrategy[]* (empty list) detailLevel : *integer* (3) creationStrategies : *MeshElementCreationStrategies []* (ByDetailedCreation) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	Creates a MeshElement based on the model element. The method enables attaching the label and properties to the element by declaring them directly or with properties collecting strategies (using model semantic data), as well as declaring creation (for computation efficiency) and scanning (used in later scanning) strategies.
ByElements	elements : *ModelElement[]* label : *string* ("unknown") additionalProperties : *MeshAdditionalProperty[ ]* (empty list) additionalPropertiesStrategies : *MeshElementAdditionalPropertyStrategy[]* (empty list) detailLevel : *integer* (3) creationStrategies : *MeshElementCreationStrategies []* (ByDetailedCreation) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	A variation of *MeshElement.ByElement* method for multiple elements. One MeshElement is created from multiple ModelElements.

BySolid	solid : *Solid* label : *string* (“unknown”) additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	Creates a MeshElement based on the solid. The method enables attaching the label and properties to the element, as well as scanning strategies (used in later scanning).
BySolids	solids : *Solid[]* label : *string* (“unknown”) additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	A variation of *MeshElement.BySolid* method for multiple solids. One MeshElement is created from multiple solids.
BySolids	surfaces : *Surface[]* label : *string* (“unknown”) additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	Creates a MeshElement based on surfaces. The method enables attaching a label and properties to the element, as well as scanning strategies (used in later scanning).
ByTriangles	triangles : *MeshTriangle[]* label : *string* (“unknown”) additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshElement : *MeshElement*	Creates a MeshElement based on MeshTriangles. The method gives advanced control over the mesh geometry since triangulation is not done automatically, but the mesh consists of user-created triangles.
Visualize	meshElement : *MeshElement*	PolyCurves: *PolyCurve[]*	Visualizes the MeshElement as polycurves representing MeshTriangles.
Label	meshElement : *MeshElement*	Label : *string*	Gives a label of the MeshElement.

Triangles	meshElement : *MeshElement*	MeshTriangles : *MeshTriangle[]*	Gives a list of all MeshTriangles of the MeshElement.
class: **MeshTriangle**			
Method	*Inputs*	*Outputs*	*Description*
ByPoints	p1 : *Point* p2 : *Point* p3 : *Point* label : *string* ("unknown") additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshTriangle : *MeshTriangle*	Creates a MeshTriangle based on points. The method enables attaching the label and properties to the triangle, as well as scanning strategies (used in later scanning).
ByPolyCurve	polyCurve: *PolyCurve* label : *string* ("unknown") additionalProperties : *MeshAdditionalProperty[ ]* (empty list) scanningStrategies: *MeshScanningStrategy[]* (empty list)	MeshTriangle : *MeshTriangle*	Creates a MeshTriangle based on a triangular polycurve. The method enables attaching the label and properties to the triangle, as well as scanning strategies (used in later scanning).
AdditionalProperties	meshTriangle : *MeshTriangle*	additionalProperties : *MeshAdditionalProperty[ ]*	Gives a list of all additional properties attached to the MeshTriangle.
Geometry	meshTriangle : *MeshTriangle*	Geometry : *PolyCurve*	Gives a polycurve representing the geometry of the MeshTriangle.
Vertices	meshTriangle : *MeshTriangle*	Vertices : *MeshPoint[]*	Gives a list of 3 MeshPoints of the MeshTriangle.
Label	meshTriangle : *MeshTriangle*	Label : *string*	Gives a label of the MeshTriangle.
class: **MeshPoint**			
Method	*Inputs*	*Outputs*	*Description*
Geometry	meshPoint : *MeshPoint*	Geometry : *Point*	Gives a point representing the MeshPoint location.
X	meshPoint : *MeshPoint*	XCoordinate : *double*	Gives the X coordinate of the MeshPoint.

Y	meshPoint : *MeshPoint*	YCoordinate : *double*	Gives the Y coordinate of the MeshPoint.
Z	meshPoint : *MeshPoint*	ZCoordinate : *double*	Gives the Z coordinate of the MeshPoint.
class: **MeshAdditionalProperty**			
Method	*Inputs*	*Outputs*	*Description*
ByLabelAndValue	label : *string* value : *object* (any type)	MeshAdditionalProperty : *MeshAdditionalProperty*	Creates a MeshAdditionalProperty by declaring its label and value (of any type). The property can be attached to a Mesh, MeshElement, or MeshTriangle.
Label	meshAdditionalProperty : *MeshAdditionalProperty*	Label : *string*	Gives a label of the MeshAdditionalProperty.
Value	meshAdditionalProperty : *MeshAdditionalProperty*	Value : *object* (any type)	Gives a value of the MeshAdditionalProperty.
MeshColor.ByRGB	r : *integer* (0 – 255) g : *integer* (0 – 255) b : *integer* (0 – 255)	ColorProperties : *MeshAdditionalProperty[ ]*	Creates a list of MeshAdditionalProperties representing the color. These properties can also be created with universal *MeshAdditionalProperty.ByLabelAndValue* method, but since the color is often used, this method is a shortcut for creating it.
class: **MeshElementAdditionalPropertyStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
AttachElementMaterialColor.Instance	-	Instance : *AttachElementMaterialColor*	Gets an instance of the MeshElementAdditionalPropertyStrategy that enables automated attaching color of model elements (using model semantics) as Mesh properties.
AttachElementMaterialTransparencyAsReflectivity.Instance	-	Instance : *AttachElementMaterialTransparencyAsReflectivity*	Gets an instance of the MeshElementAdditionalPropertyStrategy that enables automated attaching material reflectivity of model elements (using model semantics) as Mesh

			properties. This property is useful for scanning with reflections.
class: **MeshReflectivityDeclarationStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
ByMeshAdditionalPropertyThreshold.Instance	-	Instance : *ByMeshAdditionalPropertyThreshold*	Gets an instance of the MeshReflectivityDeclarationStrategy that enables distinction of the reflective triangles of a Mesh based on its properties values (MeshTriangles with properties values above the threshold are treated as reflective). The declaration strategy is useful for scanning with reflections.
class: **MeshElementCreationStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
ByDetailedCreation.Instance	detailLevel : *integer* (0-3)	Instance : *ByDetailedCreation*	Gets an instance of the MeshElementCreationStrategy that leads to creating a MeshElement geometry with all details.
SelectFacesWithBiggestAreaByNumber.Instance	number: *integer* detailLevel : *integer* (0-3)	Instance : *SelectFacesWithBiggestAreaByNumber*	Gets an instance of the MeshElementCreationStrategy that leads to creating a MeshElement geometry with only a selected number of the biggest faces.
SelectFacesWithBiggestAreaByParameter.Instance	parameter: *double* (0-1) detailLevel : *integer* (0-3)	Instance : *SelectFacesWithBiggestAreaByNumber*	Gets an instance of the MeshElementCreationStrategy that leads to creating a MeshElement geometry with only a selected portion of the biggest faces. The parameter specifies the portion, e.g., parameter = 0.1 indicates selecting 10 percent of the biggest faces.
class: **MeshScanningStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
AddConstant Translation.Instance	x : *double* y : *double* z : *double*	Instance : *AddConstantTranslation*	Gets an instance of the MeshScanningStrategy that leads to applying constant translation to point

			cloud points of a MeshTriangle, to which the instance has been attached.
AddRandomNormal Translation.Instance	mean : *double* (0) stdDev : *double* direction : *Vector*	Instance : *AddRandomNormalTranslation*	Gets an instance of the MeshScanningStrategy that leads to applying random translation (in a normal distribution) to point cloud points of a MeshTriangle, to which the instance has been attached.
AddRandomTranslation.Instance	minRange : *double* maxRange : *double* direction : *Vector*	Instance : *AddRandomTranslation*	Gets an instance of the MeshScanningStrategy that leads to applying random translation to point cloud points of a MeshTriangle, to which the instance has been attached.

Table D-2. DynamoPCSim *Scanner* main classes and methods

class: **Scanner**			
Method	*Inputs*	*Outputs*	*Description*
BySetupAnglesMaximumRangeAndErrors	horizontalAngleResolution : *double* verticalAngleResolution : *double* topBlankAngleArea : *double* (0) bottomBlankAngleArea : *double* (0) maximumRange : *double* errors : *ScannerError[]* (empty list)	Scanner : *Scanner*	Creates a Scanner with customized properties, including ScannerErrors. It enables emulating real tools' specifications.
GenerateRaysSphere	scanner : *Scanner* position : *Point*	Rays : *Ray[]*	Creates a list of Rays representing a "rays sphere" at a position regarding the Scanner's properties. This list is also used during scanning.
HorizontalAngleResolution	scanner : *Scanner*	HorizontalAngleResolution : *double*	Gives the horizontal angle resolution of the Scanner.
VerticalAngleResolution	scanner : *Scanner*	VerticalAngleResolution : *double*	Gives the vertical angle resolution of the Scanner.

TopBlankAngleArea	scanner : *Scanner*	TopBlankAngleArea : *double*	Gives the top blank angle area of the Scanner.
BottomBlankAngleArea	scanner : *Scanner*	BottomBlankAngleArea : *double*	Gives the bottom blank angle area of the Scanner.
MaximumRange	scanner : *Scanner*	MaximumRange : *double*	Gives the maximum range of the Scanner.
Errors	scanner : *Scanner*	ScannerErrors : *ScannerError[]*	Gives a list of ScannerErrors attached to the Scanner.
class: **ScannerError** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
HorizontalScannerError.Instance	baseStdDev : *double* addStdDevPerDistanceUnit : *double* (0) mean : *double* (0)	Instance : *HorizontalScannerError*	Gets an instance of the ScannerError that leads to applying random horizontal translations (in a normal distribution) to point cloud points.
VerticalScannerError.Instance	baseStdDev : *double* addStdDevPerDistanceUnit : *double* (0) mean : *double* (0)	Instance : *VerticalScannerError*	Gets an instance of the ScannerError that leads to applying random vertical translations (in a normal distribution) to point cloud points.
RangeScannerError.Instance	baseStdDev : *double* addStdDevPerDistanceUnit : *double* (0) mean : *double* (0)	Instance : *RangeScannerError*	Gets an instance of the ScannerError that leads to applying random range translations (in a normal distribution) to point cloud points.
class: **Ray**			
Method	*Inputs*	*Outputs*	*Description*
ByOriginAndDirection	origin : *Point* direction : *Vector*	Ray : *Ray*	Creates a Ray originating in the specified point and with the specified direction.
Visualize	ray: *Ray* lineLength : *double*	Line : *line*	Visualizes the Ray as a line of the specified length.
Origin	ray: *Ray*	Origin : *Point*	Gives a point representing the origin of the Ray.
Direction	ray: *Ray*	Direction : *Vector*	Gives a vector representing the direction of the Ray.

Table D-3. DynamoPCSim *PointCloud* main classes and methods

class: **PointCloud**			
Method	*Inputs*	*Outputs*	*Description*
ByScanning	mesh : *Mesh* scanner : *Scanner* scannerPosition : *Point* additionalPropertiesTypesToAttach : *PointCloudAdditionalPropertyType[]* (empty list)	PointCloud : *PointCloud*	Creates a PointCloud simulating scanning. The Mesh is scanned by the Scanner in the specified position. The method enables declaring properties' types to be attached to resulting points.
ByScanningWithReflections	mesh : *Mesh* scanner : *Scanner* scannerPosition : *Point* reflectivityStrategy : *MeshReflectivityDeclarationStrategy* maxRaysReflections : *int* additionalPropertiesTypesToAttach : *PointCloudAdditionalPropertyType[]* (empty list)	PointCloud : *PointCloud*	Creates a PointCloud simulating scanning with possible reflections. The Mesh is scanned by the Scanner in the specified position. The method enables declaring properties' types to be attached to resulting points. MeshReflectivityDeclarationStrategy enables declaring a strategy to select reflective parts of the scanned Mesh. MaxRaysReflections enables limiting the maximum reflections' number.
FromXYZ	filePath : *string*	PointCloud : *PointCloud*	Retrieves a PointCloud from an XYZ file. The PointCloud can then be processed by the algorithm.
Join	pointClouds : *PointCloud[]* similarityRange : *double*	PointCloud : *PointCloud*	Merges a list of PointClouds to one PointCloud, simulating the registration process. SimilarityRange specifies the distance in which points of the input PointClouds are merged into one.
Sample	pointCloud : *PointCloud* samplingStrategy : *PointCloudSamplingStrategy*	PointCloud : *PointCloud*	Creates a PointCloud by sampling the input PointCloud.
Transform	pointCloud : *PointCloud* transformationStrategies : *PointCloudTransformationStrategy[]*	PointCloud : *PointCloud*	Creates a transformed PointCloud

FilteredByAllLabels	pointCloud : *PointCloud*	PointClouds : *PointCloud[]*	Splits the input PointClouds into multiple single-label PointClouds (containing points of only one label).
FilteredByLabel	pointCloud : *PointCloud* label : *string*	PointCloud : *PointCloud*	Creates a single-label PointCloud with points of only declared label.
SaveToXYZ	pointCloud : *PointCloud* filePath : *string* propertiesToSave : *PointCloudAdditionalPropertyType[]* (empty list) customComments : *string* (empty list)	-	Saves the PointCloud into XYZ file with declared properties to save and optional custom comments.
Visualize	pointCloud : *PointCloud*	Points: *Point[][]*	Visualizes the PointCloud points. Points are separated into lists regarding their labels.
Labels	pointCloud : *PointCloud*	Labels : *string[]*	Gives a list of all distinct labels of the PointCloud.
Points	pointCloud : *PointCloud*	MeshTriangles : MeshTriangle[]	Gives a list of all PointCloudPoints of the PointCloud.
class: **PointCloudPoint**			
Method	*Inputs*	*Outputs*	*Description*
Label	pointCloudPoint : *PointCloudPoint*	Label : *string*	Gives a label of the PointCloudPoint.
AdditionalProperties	pointCloudPoint : *PointCloudPoint*	AdditionalProperties : *PointCloudAdditionalProperty*	Gives a list of all additional properties attached to the PointCloudPoint.
Geometry	pointCloudPoint : *PointCloudPoint*	Geometry : *Point*	Gives a point representing the PointCloudPoint location.
X	pointCloudPoint : *PointCloudPoint*	XCoordinate : *double*	Gives the X coordinate of the PointCloudPoint.
Y	pointCloudPoint : *PointCloudPoint*	YCoordinate : *double*	Gives the Y coordinate of the PointCloudPoint.
Z	pointCloudPoint : *PointCloudPoint*	ZCoordinate : *double*	Gives the Z coordinate of the PointCloudPoint.
class: **PointCloudAdditionalPropertyType**			
Method	*Inputs*	*Outputs*	*Description*

ByLabel	label : *string*	PointCloudAdditionalPropertyType : *PointCloudAdditionalPropertyType*	Creates an instance of PointCloudAdditionalPropertyType by the label. It is a general method for creating types.
PointCloudPointColor.Instance	-	PointCloudPointColor : *PointCloudAdditionalPropertyType[]*	Gets a list of PointCloudAdditionalPropertyTypes instances representing color.
PointCloudPointColorRed.Instance	-	PointCloudPointColorRed : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the red component of a color.
PointCloudPointColorGreen.Instance	-	PointCloudPointColorGreen : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the green component of a color.
PointCloudPointColorBlue.Instance	-	PointCloudPointColorBlue : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the blue component of a color.
PointCloudPointNormal.Instance	-	PointCloudPointNormal : *PointCloudAdditionalPropertyType[]*	Gets a list of PointCloudAdditionalPropertyTypes instances representing surface normal.
PointCloudPointColorNormalX.Instance	-	PointCloudPointNormalX : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the X component of a surface normal.
PointCloudPointColorNormalY.Instance	-	PointCloudPointNormalY : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the Y component of a surface normal
PointCloudPointColorNormalZ.Instance	-	PointCloudPointNormalZ : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the Z component of a surface normal
PointCloudPointIntensity.Instance	-	PointCloudPointIntensity : *PointCloudAdditionalPropertyType*	Gets an instance of PointCloudAdditionalPropertyType representing the intensity.

class: **PointCloudSamplingStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
RandomSampling.Instance	pointsNumber : *integer*	Instance : *RandomSampling*	Gets an instance of the PointCloudSamplingStrategy that enables sampling the selected number of random points.
RandomSamplingWithPreservedLabelsDistribution.Instance	pointsNumber : *integer*	Instance : *RandomSamplingWithPreservedLabelsDistribution*	Gets an instance of the PointCloudSamplingStrategy that enables sampling the selected number of random points with a preserved distribution of labels.

class: **PointCloudTransformationStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
AddConstantTranslation.Instance	x : *double* y : *double* z : *double* selectionStrategies : *SelectAll*	Instance : *AddConstantTranslation*	Gets an instance of the PointCloudTransformationStrategy that leads to applying constant translation to selected point cloud points.
AddRandomNormalTranslation.Instance	mean : *double* (0) stdDev : *double* direction : *Vector* selectionStrategies : *SelectAll*	Instance : *AddRandomNormalTranslation*	Gets an instance of the PointCloudTransformationStrategy that leads to applying random translation (in a normal distribution) to selected point cloud points.
AddRandomTranslation.Instance	minRange : *double* maxRange : *double* direction : *Vector* selectionStrategies : *SelectAll*	Instance : *AddRandomTranslation*	Gets an instance of the PointCloudTransformationStrategy that leads to applying random translation to selected point cloud points.
AddRandomNormalAdditionalPropertyAlteration.Instance	mean : *double* (0) stdDev : *double* shouldBeInteger : *bool* propertyLabel : *string* selectionStrategies : *SelectAll*	Instance : *AddRandomNormalAdditionalPropertyAlteration*	Gets an instance of the PointCloudTransformationStrategy that leads to applying alteration (in a normal distribution) to an additional property of the label to selected point cloud points.
RotatePointsAroundTheirCentroidInHori	degree : *double* selectionStrategies : *SelectAll*	Instance : *RotatePointsAroundThei*	Gets an instance of the PointCloudTransformationStrategy

zontalPlane.Instance		*rCentroidInHorizontalPlane*	that leads to rotating selected point cloud points around their centroid.
CopyPointsWithTranslations.Instance	translationVectors : *Vector[]* selectionStrategies : *SelectAll*	Instance : *CopyPointsWithTranslations*	Gets an instance of the PointCloudTransformationStrategy that leads to copying selected point cloud points with specified translation vectors.
DeletePoints.Instance	selectionStrategies : *SelectAll*	Instance : *DeletePoints*	Gets an instance of the PointCloudTransformationStrategy that leads to deleting selected point cloud points.
DeletePointsWithProbability.Instance	deletionProbability : *double* (0-1) selectionStrategies : *SelectAll*	Instance : *DeletePointsWithProbability*	Gets an instance of the PointCloudTransformationStrategy that leads to deleting selected point cloud points with the specified probability.
AddNoisePoints.Instance	pointsNumber : *integer* minBoundingBoxPoint : *Point* maxBoundingBoxPoint : *Point* label : *string* noisePointsStrategies : *NoisePointsStrategy[]*	Instance : *AddNoisePoints*	Gets an instance of the PointCloudTransformationStrategy that leads to adding noise points of the specified label. Min/MaxBoundingBoxPoints specifies an area of points' insertion. NoisePointsStrategies specifies, e.g., attaching properties to the noise points.
class: **NoisePointsStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*
AddAdditionalPropertyWithValue.Instance	propertyLabel : *string* value : *object* (any type)	Instance : *AddAdditionalPropertyWithValue*	Gets an instance of the NoisePointsStrategy that leads to attaching a property of a specified label and value to noise points.
AddAdditionalPropertyInRange.Instance	propertyLabel : *string* min : *double* min : *double* isInteger : *bool* (false)	Instance : *AddAdditionalPropertyInRange*	Gets an instance of the NoisePointsStrategy that leads to attaching a property of a specified label and random value in the range to noise points.
class: **PointCloudPointsSelectionStrategy** (and inheriting classes)			
Method	*Inputs*	*Outputs*	*Description*

SelectAll.Instance	-	Instance : *SelectAll*	Gets an instance of the PointCloudPointsSelectionStrategy that leads to selecting all points of a point cloud.
SelectByLabels.Instance	labels : *string[]*	Instance : *SelectByLabels*	Gets an instance of the PointCloudPointsSelectionStrategy that leads to selecting point cloud's points based on their labels.
SelectByAdditionalPropertyValues.Instance	propertyLabel : *string* permittedValues : *object[]* (list of any types)	Instance : *SelectByAdditionalPropertyValues*	Gets an instance of the PointCloudPointsSelectionStrategy that leads to selecting point cloud's points with a property of the specified label and of the specified values.
SelectByAdditionalPropertyValueRange.Instance	propertyLabel : *string* min : *double* max : *double*	Instance : *SelectByAdditionalPropertyValueRange*	Gets an instance of the PointCloudPointsSelectionStrategy that leads to selecting point cloud's points that have a property with the specified label and of values in the specified range.

References

[1] W. Brooks, K. Donovan, Eliminating Uncertainty in Market Access: The Impact of New Bridges in Rural Nicaragua, Econometrica. 88 (2020) 1965–1997. https://doi.org/10.3982/ecta15828.

[2] A.R.& T.B. Association, Bridge Report 2022, 2022. https://artbabridgereport.org/reports/2022-ARTBA-Bridge-Report.pdf.

[3] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, L. Wang, Modeling, Simulation, Information Technology & Processing Roadmap - Technology Area 11, Natl. Aeronaut. Sp. Adm. (2010).

[4] Research and Markets, Digital Twin Market Research Report: By Type, Technology, Enterprise, Application, Industry - Global Industry Analysis and Growth Forecast to 2030, 2021.

[5] S. Boschert, R. Rosen, Digital Twin - The Simulation Aspect, in: Mechatron. Futur. Challenges Solut. Mechatron. Syst. Their Des., Springer International Publishing, 2016: pp. 59–74. https://doi.org/10.1007/978-3-319-32156-1.

[6] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, Transdiscipl. Perspect. Complex Syst. New Find. Approaches. (2016) 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

[7] M.W. Grieves, Product lifecycle management: the new paradigm for enterprises, Int. J. Prod. Dev. 2 (2005) 71–84. https://doi.org/10.1504/ijpd.2005.006669.

[8] M. Grieves, Product Lifecycle Management: Driving the Next Generation of Lean Thinking, 2006. https://doi.org/10.1111/j.1540-5885.2007.00250_2.x.

[9] B. Piascik, J. Vickers, D. Lowry, S. Scotti, J. Stewart, A. Calomino, Materials, Structures, Mechanical Systems, and Manufacturing Roadmap - Technology Area 12, Natl. Aeronaut. Sp. Adm. (2010).

[10] P.A. Kobryn, E.J. Tuegel, Condidtion-based Maintenance Plus Structural Integrity (CBM+SI) & the Airframe Digital Twin, 2011.

[11] E.J. Tuegel, A.R. Ingraffea, T.G. Eason, S.M. Spottswood, Reengineering aircraft structural life prediction using a digital twin, Int. J. Aerosp. Eng. 2011 (2011). https://doi.org/10.1155/2011/154798.

[12] M. Grieves, Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management, Space Coast Press, 2011.

[13] E.H. Glaessgen, D.S. Stargel, The digital twin paradigm for future NASA and U.S. Air force vehicles, Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2012). https://doi.org/10.2514/6.2012-1818.

[14] E.J. Tuegel, The airframe digital twin: Some challenges to realization, Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2012) 1–8. https://doi.org/10.2514/6.2012-1812.

[15] J. Lee, E. Lapira, B. Bagheri, H. an Kao, Recent advances and trends in predictive manufacturing systems in big data environment, Manuf. Lett. 1 (2013) 38–41. https://doi.org/10.1016/j.mfglet.2013.09.005.

[16] A. Cerrone, J. Hochhalter, G. Heber, A. Ingraffea, On the effects of modeling as-manufactured geometry: Toward digital twin, Int. J. Aerosp. Eng. 2014 (2014). https://doi.org/10.1155/2014/439278.

[17] M. Grieves, Digital Twin : Manufacturing Excellence through Virtual Factory Replication, 2015.

[18] I. Gartner, Gartners Top 10 Technology Trends 2017, (2016). https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017.

[19] B.T. Gockel, A.W. Tudor, M.D. Brandyberry, R.C. Penmetsa, E.J. Tuegel, Challenges with structural life forecasting using realistic mission profiles, Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2012) 1–11. https://doi.org/10.2514/6.2012-1813.

[20] K. Reifsnider, P. Majumdar, Multiphysics stimulated simulation digital twin methods for fleet management, Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2013) 1–11.

[21] J.D. Hochhalter, W.P. Leser, J.A. Newman, E.H. Glaessgen, S.R. Cornell, S.A. Willard, G. Heber, V.K. Gupta, V. Yamakov, S.R. Cornell, S.A. Willard, G. Heber, Coupling Damage-Sensing Particles to the Digitial Twin

Concept, NASA Tech. Reports Serv. (2014) 10. https://ntrs.nasa.gov/search.jsp?R=20140006408.

[22] E.M. Kraft, The US air force digital thread/digital Twin – life cycle integration and use of computational and experimental knowledge, 54th AIAA Aerosp. Sci. Meet. 0 (2016) 1–22. https://doi.org/10.2514/6.2016-0897.

[23] M. Schluse, J. Rossmann, S. Development, From Simulation to Experimentable Digital Twins, IEEE Int. Symp. Syst. Eng. (2016) 1–6.

[24] T. Gabor, L. Belzner, M. Kiermeier, M.T. Beck, A. Neitz, A simulation-based architecture for smart cyber-physical systems, Proc. - 2016 IEEE Int. Conf. Auton. Comput. ICAC 2016. (2016) 374–379. https://doi.org/10.1109/ICAC.2016.29.

[25] A. Canedo, Industrial IoT lifecycle via digital twins, in: Proc. Elev. IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign Syst. Synth. - CODES '16., 2016. https://doi.org/doi:10.1145/2968456.2974007.

[26] G.N. Schroeder, C. Steinmetz, C.E. Pereira, D.B. Espindola, Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange, IFAC-PapersOnLine. 49 (2016) 12–17. https://doi.org/10.1016/j.ifacol.2016.11.115.

[27] E. Negri, L. Fumagalli, M. Macchi, A Review of the Roles of Digital Twin in CPS-based Production Systems, Procedia Manuf. 11 (2017) 939–948. https://doi.org/10.1016/j.promfg.2017.07.198.

[28] A. Parrott, L. Warshaw, Industry 4.0 and the digital twin, Deloitte Univ. Press. (2017) 1–17. https://dupress.deloitte.com/dup-us-en/focus/industry-4-0/digital-twin-technology-smart-factory.html.

[29] R. Stark, T. Damerau, Digital Twin, in: CIRP Encycl. Prod. Eng., Springer Berlin Heidelberg, Berlin, Heidelberg, 2019: pp. 1–8. https://doi.org/10.1007/978-3-642-35950-7_16870-1.

[30] A. Croatti, M. Gabellini, S. Montagna, A. Ricci, On the Integration of Agents and Digital Twins in Healthcare, J. Med. Syst. 44 (2020) 1–8. https://doi.org/10.1007/s10916-020-01623-5.

[31] M. Liu, S. Fang, H. Dong, C. Xu, Review of digital twin about concepts, technologies, and industrial applications, J. Manuf. Syst. 58 (2021) 346–361. https://doi.org/10.1016/j.jmsy.2020.06.017.

[32] D.M. Botín-Sanabria, S. Mihaita, R.E. Peimbert-García, M.A. Ramírez-Moreno, R.A. Ramírez-Mendoza, J. de J. Lozoya-Santos, Digital Twin Technology Challenges and Applications: A Comprehensive Review, Remote Sens. 14 (2022). https://doi.org/10.3390/RS14061335.

[33] F. Jiang, L. Ma, T. Broyd, K. Chen, Digital twin and its implementations in the civil engineering sector, Autom. Constr. 130 (2021) 103838. https://doi.org/10.1016/j.autcon.2021.103838.

[34] F. Tao, H. Zhang, A. Liu, A.Y.C. Nee, Digital Twin in Industry: State-of-the-Art, IEEE Trans. Ind. Informatics. 15 (2019) 2405–2415. https://doi.org/10.1109/TII.2018.2873186.

[35] L. Li, B. Lei, C. Mao, Digital twin in smart manufacturing, J. Ind. Inf. Integr. 26 (2022) 100289. https://doi.org/10.1016/j.jii.2021.100289.

[36] Q. Liu, H. Zhang, J. Leng, X. Chen, Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system, Int. J. Prod. Res. 57 (2019) 3903–3919. https://doi.org/10.1080/00207543.2018.1471243.

[37] F.E.I. Tao, M. Zhang, Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing, IEEE Access. 5 (2017) 20418–20427. https://doi.org/10.1109/ACCESS.2017.2756069.

[38] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S.C.Y. Lu, A.Y.C. Nee, Digital twin-driven product design framework, Int. J. Prod. Res. 57 (2019) 3935–3953. https://doi.org/10.1080/00207543.2018.1443229.

[39] H. Zhang, Q. Liu, X. Chen, D. Zhang, J. Leng, X. Chen, D. Zhang, J. Leng, A Digital Twin-Based Approach for Designing and Multi-Objective Optimization of Hollow Glass Production Line, IEEE Access. 5 (2017) 26901–26911. https://doi.org/10.1109/ACCESS.2017.2766453.

[40] U. Dahmen, J. Rossmann, Experimentable Digital Twins for a Modeling and Simulation-based Engineering Approach, 4th IEEE Int. Symp. Syst. Eng. ISSE 2018 - Proc. (2018). https://doi.org/10.1109/SysEng.2018.8544383.

[41] F. Tao, M. Zhang, Y. Liu, A.Y.C. Nee, Digital twin driven prognostics and health management for complex equipment, CIRP Ann. 67 (2018) 169–172. https://doi.org/10.1016/j.cirp.2018.04.055.

[42] T. DebRoy, W. Zhang, J. Turner, S.S. Babu, Building digital twins of 3D printing machines, Scr. Mater. 135 (2017) 119–124. https://doi.org/10.1016/j.scriptamat.2016.12.005.

[43] M. Schluse, J. Rossmann, From Simulation to Experimentable Digital Twins, IEEE Int. Symp. Syst. Eng. (2016) 1–6.

[44] M.G. Kapteyn, J.V.R. Pretorius, K.E. Willcox, A probabilistic graphical model foundation for enabling predictive digital twins at scale, Nat. Comput. Sci. 1 (2021) 337–347. https://doi.org/10.1038/s43588-021-00069-0.

[45] K.M. Alam, A. El Saddik, C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems, IEEE Access. 5 (2017) 2050–2062. https://doi.org/10.1109/ACCESS.2017.2657006.

[46] K. Alshammari, T. Beach, Y. Rezgui, Cybersecurity for digital twins in the built environment: Current research and future directions, J. Inf. Technol. Constr. 26 (2021) 159–173. https://doi.org/10.36680/j.itcon.2021.010.

[47] I.S. Khan, M.O. Ahmad, J. Majava, Industry 4.0 and sustainable development: A systematic mapping of triple bottom line, Circular Economy and Sustainable Business Models perspectives, J. Clean. Prod. 297 (2021) 126655. https://doi.org/10.1016/j.jclepro.2021.126655.

[48] Q. Qi, F. Tao, Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison, IEEE Access. 6 (2018) 3585–3593. https://doi.org/10.1109/ACCESS.2018.2793265.

[49] A. Parott, L. Warshaw, Deloitte Industry 4.0 and the digital twin, Deloitte. (2017). https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html.

[50] V. Alcácer, V. Cruz-Machado, Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems, Eng. Sci. Technol. an Int. J. 22 (2019) 899–919. https://doi.org/10.1016/j.jestch.2019.01.006.

[51] B. Teisserenc, S. Sepasgozar, Adoption of Blockchain Technology through Digital Twins in the Construction Industry 4.0: A PESTELS Approach, Buildings. 11 (2021) 670. https://doi.org/10.3390/buildings11120670.

[52] D. Lee, S.H. Lee, N. Masoud, M.S. Krishnan, V.C. Li, Integrated digital twin and blockchain framework to support accountable information sharing in construction projects, Autom. Constr. 127 (2021) 103688. https://doi.org/10.1016/j.autcon.2021.103688.

[53] D. Kiritsis, Computer-Aided Design Closed-loop PLM for intelligent products in the era of the Internet of things, Comput. Des. 43 (2011) 479–501. https://doi.org/10.1016/j.cad.2010.03.002.

[54] M. Abramovici, J.C. Göbel, H.B. Dang, Semantic data management for the development and continuous reconfiguration of smart products and systems, CIRP Ann. - Manuf. Technol. 65 (2016) 185–188. https://doi.org/10.1016/j.cirp.2016.04.051.

[55] K.A. Hribernik, L. Rabe, K.D. Thoben, J. Schumacher, The product avatar as a product-instance-centric information management concept, Int. J. Prod. Lifecycle Manag. 1 (2006) 367–379. https://doi.org/10.1504/IJPLM.2006.011055.

[56] J. Ríos, J.C. Hernández, M. Oliva, F. Mas, J.C. Hernandez-matias, Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft, Adv. Transdiscipl. Eng. 2 (2015) 657–666. https://doi.org/10.3233/978-1-61499-544-9-657.

[57] M. Sacco, P. Pedrazzoli, W. Terkaj, VFF: Virtual Factory Framework, 2010 IEEE Int. Technol. Manag. Conf. ICE 2010. (2010). https://doi.org/10.1109/ICE.2010.7477041.

[58] Y. Lu, X. Xu, Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services, Robot. Comput. Integr. Manuf. 57 (2019) 92–102. https://doi.org/10.1016/j.rcim.2018.11.006.

[59] M. Schluse, M. Priggemeyer, L. Atorf, J. Rossmann, Experimentable Digital Twins-Streamlining Simulation-Based Systems Engineering for Industry 4.0, IEEE Trans. Ind. Informatics. 14 (2018) 1722–1731. https://doi.org/10.1109/TII.2018.2804917.

[60] K. Bruynseels, F.S. de Sio, J. van den Hoven, F.S. De Sio, J. Van Den Hoven, Digital Twins in health care: Ethical implications of an emerging engineering paradigm, Front. Genet. 9 (2018) 1–11. https://doi.org/10.3389/fgene.2018.00031.

[61] K. Gkouskou, I. Vlastos, P. Karkalousos, D. Chaniotis, D. Sanoudou, A.G. Eliopoulos, The “virtual Digital Twins” Concept in Precision Nutrition, Adv. Nutr. 11 (2020) 1405–1413. https://doi.org/10.1093/advances/nmaa089.

[62] S. Aheleroff, X. Xu, R.Y. Zhong, Y. Lu, Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model, Adv. Eng. Informatics. 47 (2021) 101225. https://doi.org/10.1016/j.aei.2020.101225.

[63] K. Subramanian, Digital Twin for Drug Discovery and Development—The Virtual Liver, J. Indian Inst. Sci. 100 (2020) 653–662. https://doi.org/10.1007/s41745-020-00185-2.

[64] M.A. Saghiri, J. Vakhnovetsky, A.M. Saghiri, The future of digital twins in precision dentistry, J. Oral Biol. Craniofacial Res. 13 (2023) 19. https://doi.org/10.1016/j.jobcr.2022.10.003.

[65] A. Gaggioli, Digital Twins: An Emerging Paradigm in Cyberpsychology Research?, Cyberpsychology, Behav. Soc. Netw. 21 (2018) 468–469. https://doi.org/10.1089/cyber.2018.29118.csi.

[66] Z. Cheng, V.E. Kuzmichev, Digital twin and men's underwear design, IOP Conf. Ser. Mater. Sci. Eng. 459 (2018). https://doi.org/10.1088/1757-899X/459/1/012075.

[67] S. Neethirajan, B. Kemp, Digital twins in livestock farming, Animals. 11 (2021). https://doi.org/10.3390/ani11041008.

[68] A. El Saddik, Digital Twins: The Convergence of Multimedia Technologies, IEEE Multimed. 25 (2018) 87–92. https://doi.org/10.1109/MMUL.2018.023121167.

[69] P. Bauer, B. Stevens, W. Hazeleger, A digital twin of Earth for the green transition, Nat. Clim. Chang. 2021 112. 11 (2021) 80–83. https://doi.org/10.1038/s41558-021-00986-y.

[70] S. Dhar, P. Tarafdar, I. Bose, Understanding the evolution of an emerging technological paradigm and its impact: The case of Digital Twin, Technol. Forecast. Soc. Change. 185 (2022). https://doi.org/10.1016/j.techfore.2022.122098.

[71] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital Twin in manufacturing: A categorical literature review and classification, IFAC-PapersOnLine. 51 (2018) 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474.

[72] K.Y.H. Lim, P. Zheng, C.H. Chen, A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives, J. Intell. Manuf. 31 (2020) 1313–1337. https://doi.org/10.1007/s10845-019-01512-w.

[73] C. Semeraro, M. Lezoche, H. Panetto, M. Dassisti, Digital twin paradigm: A systematic literature review, Comput. Ind. 130 (2021) 103469. https://doi.org/10.1016/j.compind.2021.103469.

[74] E. VanDerHorn, S. Mahadevan, Digital Twin: Generalization, characterization and implementation, Decis. Support Syst. 145 (2021) 113524. https://doi.org/10.1016/j.dss.2021.113524.

[75] M. Singh, E. Fuenmayor, E. Hinchy, Y. Qiao, N. Murray, D. Devine, Digital Twin: Origin to Future, Appl. Syst. Innov. 4 (2021) 36. https://doi.org/10.3390/asi4020036.

[76] F. Tao, Q. Qi, Make more digital twins, Nature. 573 (2019) 490–491. https://doi.org/10.1038/d41586-019-02849-1.

[77] Siemens, Digital twin - Driving business value throughout the building life cycle, 2018. https://assets.new.siemens.com/siemens/assets/api/uuid:610b5974-241d-4321-8ae6-55c6167446bf/bim-digitwin-ru.pdf.

[78] Siemens, Simulation & - Digital Twin in 2030, 2020. https://new.siemens.com/global/en/company/stories/research-technologies/digitaltwin/outlook2030.html.

[79] Gartner, Prepare for the Impact of Digital Twins, 2017. https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins/.

[80] C. Boje, A. Guerriero, S. Kubicki, Y. Rezgui, Towards a semantic Construction Digital Twin: Directions for future research, Autom. Constr. 114 (2020) 103179. https://doi.org/10.1016/j.autcon.2020.103179.

[81] R. Sacks, I. Brilakis, E. Pikas, H.S. Xie, M. Girolami, Construction with digital twin information systems, Data-Centric Eng. 1 (2020). https://doi.org/10.1017/dce.2020.16.

[82] S.J. et al. Bolton A, Enzer M, The Gemini Principles, Cent. Digit. Built Britain Univ. Cambridge. (2018) 15. https://www.cdbb.cam.ac.uk/system/files/documents/TheGeminiPrinciples.pdf.

[83] S.H. Khajavi, N.H. Motlagh, A. Jaribion, L.C. Werner, J. Holmstrom, Digital Twin: Vision, benefits, boundaries, and creation for buildings, IEEE Access. 7 (2019) 147406–147419.

https://doi.org/10.1109/ACCESS.2019.2946515.
[84] Q. Lu, X. Xie, A.K. Parlikad, J.M. Schooling, Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance, Autom. Constr. 118 (2020). https://doi.org/10.1016/J.AUTCON.2020.103277.
[85] Q. Lu, L. Chen, S. Li, M. Pitt, Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings, Autom. Constr. 115 (2020). https://doi.org/10.1016/J.AUTCON.2020.103183.
[86] T. Greif, N. Stein, C.M. Flath, Peeking into the void: Digital twins for construction site logistics, Comput. Ind. 121 (2020). https://doi.org/10.1016/J.COMPIND.2020.103264.
[87] A. Karakra, F. Fontanili, E. Lamine, J. Lamothe, HospiT'Win: A predictive simulation-based digital twin for patients pathways in hospital, 2019 IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2019 - Proc. (2019) 1–4. https://doi.org/10.1109/BHI.2019.8834534.
[88] D.G.J. Opoku, S. Perera, R. Osei-Kyei, M. Rashidi, Digital twin application in the construction industry: A literature review, J. Build. Eng. 40 (2021) 102726. https://doi.org/10.1016/j.jobe.2021.102726.
[89] M. Deng, C.C. Menassa, V.R. Kamat, From BIM to digital twins: A systematic review of the evolution of intelligent building representations in the AEC-FM industry, J. Inf. Technol. Constr. 26 (2021) 58–83. https://doi.org/10.36680/J.ITCON.2021.005.
[90] L. Hou, S. Wu, G.K. Zhang, Y. Tan, X. Wang, Literature review of digital twins applications in constructionworkforce safety, Appl. Sci. 11 (2021) 1–21. https://doi.org/10.3390/APP11010339.
[91] S. Kaewunruen, P. Rungskunroch, J. Welsh, A digital-twin evaluation of Net Zero Energy Building for existing buildings, Sustain. 11 (2019). https://doi.org/10.3390/SU11010159.
[92] D.N. Ford, C.M. Wolf, Smart Cities with Digital Twin Systems for Disaster Management, J. Manag. Eng. 36 (2020) 1–10. https://doi.org/10.1061/(asce)me.1943-5479.0000779.
[93] F. Xue, W. Lu, Z. Chen, C.J. Webster, From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles, ISPRS J. Photogramm. Remote Sens. 167 (2020) 418–431. https://doi.org/10.1016/J.ISPRSJPRS.2020.07.020.
[94] N. Mohammadi, J.E. Taylor, Smart city digital twins, 2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc. 2018-January (2018) 1–5. https://doi.org/10.1109/SSCI.2017.8285439.
[95] Q. Lu, A.K. Parlikad, P. Woodall, G. Don Ranasinghe, X. Xie, Z. Liang, E. Konstantinou, J. Heaton, J. Schooling, Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus, J. Manag. Eng. 36 (2020) 05020004. https://doi.org/10.1061/(asce)me.1943-5479.0000763.
[96] G. Schrotter, C. Hürzeler, The Digital Twin of the City of Zurich for Urban Planning, PFG - J. Photogramm. Remote Sens. Geoinf. Sci. 88 (2020) 99–112. https://doi.org/10.1007/s41064-020-00092-2.
[97] G. White, A. Zink, L. Codecá, S. Clarke, A digital twin smart city for citizen feedback, Cities. 110 (2021). https://doi.org/10.1016/j.cities.2020.103064.
[98] T. Ruohomaki, E. Airaksinen, P. Huuska, O. Kesaniemi, M. Martikka, J. Suomisto, Smart City Platform Enabling Digital Twin, 9th Int. Conf. Intell. Syst. 2018 Theory, Res. Innov. Appl. IS 2018 - Proc. (2018) 155–161. https://doi.org/10.1109/IS.2018.8710517.
[99] M. Austin, P. Delgoshaei, M. Coelho, M. Heidarinejad, Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach, J. Manag. Eng. 36 (2020). https://doi.org/10.1061/(ASCE)ME.1943-5479.0000774.
[100] J. Turner, R. Ferris, S. Eckhart, Infrastructure Digital Twin Maturity: A Model for Measuring Progress, 2021.
[101] Deloitte, New Technologies Case Study: Data Sharing in Infrastructure, 2017.
[102] C.S. Shim, H.R. Kang, N.S. Dang, Digital twin models for maintenance of cable-supported bridges, Int. Conf. Smart Infrastruct. Constr. 2019, ICSIC 2019 Driv. Data-Informed Decis. 2019 (2019) 737–742. https://doi.org/10.1680/icsic.64669.737.
[103] S. Gunner, E. Voyagaki, G. Gavriel, N. Carhart, J. MacDonald, T. Tryfonas, C. Taylor, M. Pregnolato, Digital Twins for civil engineering: the Clifton Suspension Bridge (UK), Int. Conf. Struct. Heal. Monit. Intell. Infrastruct. Transf. Res. into Pract. SHMII. 2021-June (2021) 1561–1566.
[104] V. Saback de Freitas Bello, C. Popescu, T. Blanksvärd, B. Täljsten, Framework for Bridge Management Systems (BMS) Using Digital Twins, Lect. Notes Civ. Eng. 200 LNCE (2022) 687–694. https://doi.org/10.1007/978-3-

030-91877-4_78.

[105] C. Ye, L. Butler, B. Calka, M. Iangurazov, Q. Lu, A. Gregory, M. Girolami, C. Middleton, A digital twin of bridges for structural health monitoring, Struct. Heal. Monit. 2019 Enabling Intell. Life-Cycle Heal. Manag. Ind. Internet Things - Proc. 12th Int. Work. Struct. Heal. Monit. 1 (2019) 1619–1626. https://doi.org/10.12783/shm2019/32287.

[106] J.S. Kang, K. Chung, E.J. Hong, Multimedia knowledge-based bridge health monitoring using digital twin, Multimed. Tools Appl. 80 (2021) 34609–34624. https://doi.org/10.1007/s11042-021-10649-x.

[107] F. Hidayat, S.H. Supangkat, K. Hanafi, Digital Twin of Road and Bridge Construction Monitoring and Maintenance, (2022) 1–7. https://doi.org/10.1109/isc255366.2022.9922473.

[108] C. Ramonell, R. Chacón, Towards Automated Pipelines for Processing Load Test Data on a HS Railway Bridge in Spain using a Digital Twin, Proc. Int. Symp. Autom. Robot. Constr. 2022-July (2022) 231–237. https://doi.org/10.22260/isarc2022/0033.

[109] S. Yu, D. Li, J. Ou, Digital twin-based structure health hybrid monitoring and fatigue evaluation of orthotropic steel deck in cable-stayed bridge, Struct. Control Heal. Monit. 29 (2022) 7–13. https://doi.org/10.1002/stc.2976.

[110] J. Cervenka, L. Jendele, J. Zalsky, R. Pukl, D. Novak, Digital twin approach for durability and reliability assessment of bridges, in: Proc. Fib Symp. 2020 Concr. Struct. Resilient Soc., 2020: pp. 1840–1848.

[111] F. Jiang, Y. Ding, Y. Song, F. Geng, Z. Wang, An Architecture of Lifecycle Fatigue Management of Steel Bridges Driven by Digital Twin, Struct. Monit. Maint. 8 (2021) 187–201. https://doi.org/10.12989/smm.2021.8.2.187.

[112] F. Jiang, Y. Ding, Y. Song, F. Geng, Z. Wang, Digital Twin-driven framework for fatigue life prediction of steel bridges using a probabilistic multiscale model: Application to segmental orthotropic steel deck specimen, Eng. Struct. 241 (2021) 112461. https://doi.org/10.1016/j.engstruct.2021.112461.

[113] S. Kaewunruen, J. Sresakoolchai, W. Ma, O. Phil-Ebosie, Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions, Sustain. 13 (2021) 1–19. https://doi.org/10.3390/su13042051.

[114] F. Jiang, Y. Ding, Y. Song, F. Geng, Z. Wang, Digital Twin-driven framework for fatigue lifecycle management of steel bridges, Struct. Infrastruct. Eng. 0 (2022) 1–21. https://doi.org/10.1080/15732479.2022.2058563.

[115] S. Yoon, S. Lee, S. Kye, I.H. Kim, H.J. Jung, B.F. Spencer, Seismic fragility analysis of deteriorated bridge structures employing a UAV inspection-based updated digital twin, Struct. Multidiscip. Optim. 65 (2022) 1–17. https://doi.org/10.1007/s00158-022-03445-0.

[116] K. Lin, Y.L. Xu, X. Lu, Z. Guan, J. Li, Digital twin-based collapse fragility assessment of a long-span cable-stayed bridge under strong earthquakes, Autom. Constr. 123 (2021) 103547. https://doi.org/10.1016/j.autcon.2020.103547.

[117] K. Lin, Y.L. Xu, X. Lu, Z. Guan, J. Li, Digital twin-based life-cycle seismic performance assessment of a long-span cable-stayed bridge, Bull. Earthq. Eng. (2022). https://doi.org/10.1007/s10518-022-01567-w.

[118] F. Ghahari, N. Malekghaini, H. Ebrahimian, E. Taciroglu, Bridge Digital Twinning Using an Output-Only Bayesian Model Updating Method and Recorded Seismic Measurements, Sensors. 22 (2022). https://doi.org/10.3390/s22031278.

[119] N. Rojas-Mercedes, K. Erazo, L. Di Sarno, Seismic fragility curves for a concrete bridge using structural health monitoring and digital twins, Earthq. Struct. 22 (2022) 503–515. https://doi.org/10.12989/eas.2022.22.5.503.

[120] I.M. Giorgadze, F. Vahdatikhaki, J.H. Voordijk, Conceptual Modeling of Lifecycle Digital Twin Architecture for Bridges: A Data Structure Approach, Proc. Int. Symp. Autom. Robot. Constr. 2022-July (2022) 199–206. https://doi.org/10.22260/isarc2022/0029.

[121] M.S. Mafipour, S. Vilgertshofer, A. Borrmann, Deriving Digital Twin Models of Existing Bridges from Point Cloud Data Using Parametric Models and Metaheuristic Algorithms, EG-ICE 2021 Work. Intell. Comput. Eng. Proc. (2021) 464–474.

[122] R. Lu, I. Brilakis, Digital twinning of existing reinforced concrete bridges from labelled point clusters, Autom. Constr. 105 (2019) 102837. https://doi.org/10.1016/j.autcon.2019.102837.

[123] H. Sofia, E. Anas, O. Faiz, Mobile mapping, machine learning and digital twin for road infrastructure monitoring and maintenance: Case study of mohammed VI bridge in Morocco, Proc. - 2020 IEEE Int. Conf. Moroccan Geomatics, MORGEO 2020. (2020). https://doi.org/10.1109/Morgeo49228.2020.9121882.

[124] M. Mohammadi, M. Rashidi, V. Mousavi, A. Karami, Y. Yu, B. Samali, Case study on accuracy comparison of digital twins developed for a heritage bridge via UAV photogrammetry and terrestrial laser scanning, Int. Conf. Struct. Heal. Monit. Intell. Infrastruct. Transf. Res. into Pract. SHMII. 2021-June (2021) 1713–1720.

[125] M.S. Mafipour, S. Vilgertshofer, A. Borrmann, Creating digital twins of existing bridges through AI-based methods, IABSE Symp. Prague, 2022 Challenges Exist. Oncoming Struct. - Rep. (2022) 727–734. https://doi.org/10.2749/prague.2022.0727.

[126] D. Dan, Y. Ying, L. Ge, Digital Twin System of Bridges Group Based on Machine Vision Fusion Monitoring of Bridge Traffic Load, IEEE Trans. Intell. Transp. Syst. 23 (2021) 22190–22205. https://doi.org/10.1109/TITS.2021.3130025.

[127] H.H. Hosamo, M.H. Hosamo, Digital Twin Technology for Bridge Maintenance using 3D Laser Scanning: A Review, Adv. Civ. Eng. 2022 (2022). https://doi.org/10.1155/2022/2194949.

[128] V.S. de Freitas Bello, C. Popescu, T. Blanksvärd, B. Täljsten, C. Popescu, Framework for facility management of bridge structures using digital twins, IABSE Congr. Ghent 2021 Struct. Eng. Futur. Soc. Needs. (2021) 629–637. https://doi.org/10.2749/ghent.2021.0629.

[129] United Nations Conference on Trade and Development, Handbook of Statistics 2021, 2021.

[130] D.G. Broo, J. Schooling, Digital twins in infrastructure: definitions, current practices, challenges and strategies, Int. J. Constr. Manag. 0 (2021) 1–10. https://doi.org/10.1080/15623599.2021.1966980.

[131] Global Aliance for Buildings and Construction, Global Status Report 2021 For Buildings and Construction: Towards a Zero-emission, Efficient and Resilient Buidlings and Construction Sector, 2021.

[132] Y. Pan, L. Zhang, Roles of artificial intelligence in construction engineering and management: A critical review and future trends, Autom. Constr. 122 (2021) 103517. https://doi.org/10.1016/j.autcon.2020.103517.

[133] F. Barbosa, J. Woetzel, J. Mischke, M.J. Ribeirinho, M. Sridhar, M. Parsons, N. Bertram, S. Brown, Reinventing Construction: A Route To Higher Productivity, Mckinsey Glob. Insititute. (2017) 20. http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/reinventing-construction-through-a-productivity-revolution.

[134] M. Salamak, BIM w cyklu życia mostów (BIM in the bridges' lifecycle), Wydawnictwo Naukowe PWN, 2020.

[135] C. Ye, S.C. Kuok, L.J. Butler, C.R. Middleton, Implementing bridge model updating for operation and maintenance purposes: examination based on UK practitioners' views, Struct. Infrastruct. Eng. (2021). https://doi.org/10.1080/15732479.2021.1914115.

[136] J. Bennetts, P.J. Vardanega, C.A. Taylor, S.R. Denton, Survey of the use of data in UK bridge asset management, Proc. Inst. Civ. Eng. Bridg. Eng. 173 (2020) 211–222. https://doi.org/10.1680/jbren.18.00050.

[137] E. Jeong, J. Seo, J. Wacker, Literature Review and Technical Survey on Bridge Inspection Using Unmanned Aerial Vehicles, J. Perform. Constr. Facil. 34 (2020) 04020113. https://doi.org/10.1061/(asce)cf.1943-5509.0001519.

[138] Q.T.T. Huong, E.C.W. Lou, N. Le Hoai, Enhancing bim diffusion through pilot projects in Vietnam, Eng. J. 25 (2021) 167–176. https://doi.org/10.4186/ej.2021.25.7.167.

[139] M. Mayouf, M. Gerges, S. Cox, 5D BIM: an investigation into the integration of quantity surveyors within the BIM process, J. Eng. Des. Technol. 17 (2019) 537–553. https://doi.org/10.1108/JEDT-05-2018-0080.

[140] B. Manzoor, I. Othman, J.M. Kang, Z.W. Geem, Influence of building information modeling (BIM) implementation in high-rise buildings towards sustainability, Appl. Sci. 11 (2021). https://doi.org/10.3390/app11167626.

[141] R. Jin, C. Hancock, L. Tang, C. Chen, D. Wanatowski, L. Yang, Empirical Study of BIM Implementation–Based Perceptions among Chinese Practitioners, J. Manag. Eng. 33 (2017) 04017025. https://doi.org/10.1061/(asce)me.1943-5479.0000538.

[142] K. Korus, J. Marcin, T. Płaszczyk, M. Salamak, Digital twins and innovative techniques for bridge inspections, E-Mosty. 7 (2021) 24–31. https://e-mosty.cz/wp-content/uploads/e-mosty-Sept21.pdf.

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

[144] C. Eastman, An Outline of the Building Description System. Research Report No. 50, Carnegie-Mellon Univ., Pittsburgh, Pa. Inst. Phys. Planning. (1974) 23.

[145] J. Weseli, Teoretyczne podstawy opisu i analizy uogólnionego modelu układu most - środowisko (Theoretical basis for description and analysis of the generalized bridge-environment system), Politechnika Śląska, 1987.

[146] J. Lorimer, M. Bew, A report for the Government Construction Client Group Building Information Modelling (BIM) Working Party Strategy Paper, 2011. https://www.cdbb.cam.ac.uk/Resources/ResoucePublications/BISBIMstrategyReport.pdf.

[147] F.E. Jernigan, Big BIM, Little Bim: The Practical Approach to Building Information Modeling : Integrated Practice Done the Right Way!, 4Site Press, 2007.

[148] K. Graham, L. Chow, S. Fai, Level of detail, information and accuracy in building information modelling of existing and heritage buildings, J. Cult. Herit. Manag. Sustain. Dev. 8 (2018) 495–507. https://doi.org/10.1108/JCHMSD-09-2018-0067.

[149] J. Bedrick, W. Ikerd, J. Reinhardt, Level of Development (LOD) Specification Part I & Commentary For Building Information Models and Data, 2020. www.bimforum.org/lod.

[150] ISO 19650-3:2020. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 3: Operational phase of the ass, n.d.

[151] B. Schleich, N. Anwer, L. Mathieu, S. Wartzack, Shaping the digital twin for design and production engineering, CIRP Ann. - Manuf. Technol. 66 (2017) 141–144. https://doi.org/10.1016/j.cirp.2017.04.040.

[152] T.H.J. Uhlemann, C. Lehmann, R. Steinhilper, The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0, Procedia CIRP. 61 (2017) 335–340. https://doi.org/10.1016/j.procir.2016.11.152.

[153] J. Boyd, Build It Twice: If You Want a Successful Building Project, Split Rock Pub Co, 1998. https://doi.org/978-0966362008.

[154] A. Madni, C. Madni, S. Lucero, Leveraging Digital Twin Technology in Model-Based Systems Engineering, Systems. 7 (2019) 7. https://doi.org/10.3390/systems7010007.

[155] R. Rosen, G. Von Wichert, G. Lo, K.D.D. Bettenhausen, About the importance of autonomy and digital twins for the future of manufacturing, IFAC-PapersOnLine. 28 (2015) 567–572. https://doi.org/10.1016/j.ifacol.2015.06.141.

[156] buildingSMART International, (n.d.). https://www.buildingsmart.org/.

[157] ISO 16739-1:2018. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema, n.d.

[158] ISO 10303-11:2004. Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual, n.d.

[159] A. Borrmann, M. König, C. Koch, J. Beetz, Building Information Modeling. Technology Foundations and Industry Practice, Springer International Publishing, Cham, 2018. https://doi.org/10.1007/978-3-319-92862-3.

[160] Industry Foundation Classes, Build. Smart. (n.d.). https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/.

[161] buildingSMART, IFC 4.3 TC1 (4.3.0.1), buildingSMART, n.d. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/.

[162] A. Borrmann, S. Muhic, J. Hyvärinen, T. Chipman, S. Jaud, C. Castaing, C. Dumoulin, T. Liebich, L. Mol, The IFC-Bridge project – Extending the IFC standard to enable high-quality exchange of bridge information models, Proc. 2019 Eur. Conf. Comput. Constr. 1 (2019) 377–386. https://doi.org/10.35490/ec3.2019.193.

[163] P. Benning, C. Dumoulin, T. Chipman, J. Hyvärinen, T. Liebich, L. Mol, S. Muhič, H. Moon, H. Myllymaki, N. Suo, IFC - Bridge Fast Track Project Report WP2, 2018.

[164] S. Lockley, C. Benghi, M. Černý, Xbim.Essentials: a library for interoperable building information applications, J. Open Source Softw. 2 (2017) 473. https://doi.org/10.21105/joss.00473.

[165] usBIM.viewer+, (n.d.). https://www.accasoftware.com/en/ifc-viewer.

[166] J.M. Ko, Y.Q. Ni, Technology developments in structural health monitoring of large-scale bridges, Eng. Struct. 27 (2005) 1715–1725. https://doi.org/10.1016/j.engstruct.2005.02.021.

[167] S. Bas, N.M. Apaydin, A. Ilki, F.N. Catbas, Structural health monitoring system of the long-span bridges in Turkey, Struct. Infrastruct. Eng. 14 (2018) 425–444. https://doi.org/10.1080/15732479.2017.1360365.

[168] L. Zhang, G. Qiu, Z. Chen, Structural health monitoring methods of cables in cable-stayed bridge: A review, Meas. J. Int. Meas. Confed. 168 (2021). https://doi.org/10.1016/j.measurement.2020.108343.

[169] M. Vagnoli, R. Remenyte-Prescott, J. Andrews, Railway bridge structural health monitoring and fault detection: State-of-the-art methods and future challenges, Struct. Heal. Monit. 17 (2018) 971–1007. https://doi.org/10.1177/1475921717721137.

[170] L.J. Prendergast, M.P. Limongelli, N. Ademovic, A. Anžlin, K. Gavin, M. Zanini, Structural health monitoring for performance assessment of bridges under flooding and seismic actions, Struct. Eng. Int. 28 (2018) 296–307. https://doi.org/10.1080/10168664.2018.1472534.

[171] P.F. Giordano, M.P. Limongelli, The value of structural health monitoring in seismic emergency management of bridges, Struct. Infrastruct. Eng. 18 (2022) 537–553. https://doi.org/10.1080/15732479.2020.1862251.

[172] C. Bedon, E. Bergamo, M. Izzi, S. Noè, Prototyping and validation of MEMS accelerometers for structural health monitoring—the case study of the Pietratagliata cable-stayed bridge, J. Sens. Actuator Networks. 7 (2018). https://doi.org/10.3390/jsan7030030.

[173] D. Reagan, A. Sabato, C. Niezrecki, Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges, Struct. Heal. Monit. 17 (2018) 1056–1072. https://doi.org/10.1177/1475921717735326.

[174] L. Ngeljaratan, M.A. Moustafa, Structural health monitoring and seismic response assessment of bridge structures using target-tracking digital image correlation, Eng. Struct. 213 (2020). https://doi.org/10.1016/j.engstruct.2020.110551.

[175] J. Winkler, C. Specialist, M.D. Hansen, Innovative Long-Term Monitoring of the Great Belt Bridge Expansion Joint Using Digital Image Correlation, Https://Doi.Org/10.1080/10168664.2018.1461539. 28 (2018) 347–352. https://doi.org/10.1080/10168664.2018.1461539.

[176] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, S. Nagarajaiah, Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection, J. Struct. Eng. 146 (2020). https://doi.org/10.1061/(asce)st.1943-541x.0002535.

[177] W.H. Hu, D.H. Tang, J. Teng, S. Said, R.G. Rohrmann, Structural health monitoring of a prestressed concrete bridge based on statistical pattern recognition of continuous dynamic measurements over 14 years, Sensors. 18 (2018). https://doi.org/10.3390/S18124117.

[178] D.P. Milanoski, G.K. Galanopoulos, T.H. Loutas, Digital-twins of composite aerostructures towards structural health monitoring, 2021 IEEE Int. Work. Metrol. AeroSpace, Metroaerosp. 2021 - Proc. (2021) 613–618. https://doi.org/10.1109/MetroAeroSpace51421.2021.9511653.

[179] A. Mondoro, B. Grisso, On the Integration of SHM and Digital Twin for the Fatigue Assessment of Naval Surface Ships, Struct. Heal. Monit. 2019. (2019). https://doi.org/10.12783/SHM2019/32203.

[180] T. Rølvåg, Ø. Stranden, Digital Twin Based Structural Health Monitoring of Offshore Crane, Proc. Int. Conf. Offshore Mech. Arct. Eng. - OMAE. 2 (2022). https://doi.org/10.1115/OMAE2022-85896.

[181] I.H. Sarker, Machine Learning: Algorithms, Real-World Applications and Research Directions, SN Comput. Sci. 2 (2021) 1–21. https://doi.org/10.1007/s42979-021-00592-x.

[182] A. Darko, A.P.C. Chan, M.A. Adabre, D.J. Edwards, M.R. Hosseini, E.E. Ameyaw, Artificial intelligence in the AEC industry: Scientometric analysis and visualization of research activities, Autom. Constr. 112 (2020) 103081. https://doi.org/10.1016/j.autcon.2020.103081.

[183] K. Korus, M. Salamak, M. Jasiński, Optimization of geometric parameters of arch bridges using visual programming FEM components and genetic algorithm, Eng. Struct. 241 (2021). https://doi.org/10.1016/j.engstruct.2021.112465.

[184] D.C. Nguyen, M. Salamak, A. Katunin, M. Gerges, Finite Element Model Updating of RC Bridge Structure with

Static Load Testing: A Case Study of Vietnamese ThiThac Bridge in Coastal and Marine Environment, Sensors. 22 (2022). https://doi.org/10.3390/s22228884.

[185] M. Żarski, B. Wójcik, K. Książek, J.A. Miszczak, Finicky transfer learning—A method of pruning convolutional neural networks for cracks classification on edge devices, Comput. Civ. Infrastruct. Eng. 37 (2022) 500–515. https://doi.org/10.1111/MICE.12755.

[186] T. Li, D. Harris, Automated construction of bridge condition inventory using natural language processing and historical inspection reports, (2019) 28. https://doi.org/10.1117/12.2514006.

[187] B.A. Myers, Visual programming, programming by example, and program visualization: a taxonomy, ACM SIGCHI Bull. 17 (1986) 59–66. https://doi.org/10.1145/22339.22349.

[188] M.A. Kuhail, S. Farooq, R. Hammad, M. Bahja, Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review, IEEE Access. (2021). https://doi.org/10.1109/ACCESS.2021.3051043.

[189] T.R.G. Green, M. Petre, Usability analysis of visual programming environments: A "cognitive dimensions" framework, J. Vis. Lang. Comput. 7 (1996) 131–174. https://doi.org/10.1006/jvlc.1996.0009.

[190] S.C. Chase, Generative design tools for novice designers: Issues for selection, Autom. Constr. 14 (2005) 689–698. https://doi.org/10.1016/j.autcon.2004.12.004.

[191] K. Shea, R. Aish, M. Gourtovaia, Towards integrated performance-driven generative design tools, Autom. Constr. 14 (2005) 253–264. https://doi.org/10.1016/j.autcon.2004.07.002.

[192] C. Preidel, A. Borrmann, Towards code compliance checking on the basis of a visual programming language, J. Inf. Technol. Constr. 21 (2016) 402–421.

[193] M. Häußler, S. Esser, A. Borrmann, Code compliance checking of railway designs by integrating BIM, BPMN and DMN, Autom. Constr. 121 (2021). https://doi.org/10.1016/j.autcon.2020.103427.

[194] C. Preidel, S. Daum, A. Borrmann, Data retrieval from building information models based on visual programming, Vis. Eng. 5 (2017). https://doi.org/10.1186/s40327-017-0055-0.

[195] E. Elbeltagi, H. Wefki, S. Abdrabou, M. Dawood, A. Ramzy, Visualized strategy for predicting buildings energy consumption during early design stage using parametric analysis, J. Build. Eng. 13 (2017) 127–136. https://doi.org/10.1016/j.jobe.2017.07.012.

[196] T. Suesser, T. Dogan, Campus Energy Model: Using a Semi-Automated Workflow To Build Spatially Resolved Campus Building Energy Models for Climate Change and Net-Zero Scenario Evaluation, Build. Simul. 2017 15th IBPSA Conf. (2017) 1720–1729.

[197] K. Kensek, Visual programming for building information modeling: Energy and shading analysis case studies, J. Green Build. 10 (2015) 28–43. https://doi.org/10.3992/jgb.10.4.28.

[198] M.R. Asl, M. Bergin, AdamMenter, W. Yan, BIM-based Parametric Building Energy Performance MultiObjective Optimization, 32nd ECAADe Conf. 224 (2014) 10. http://autodeskresearch.com/pdf/bimparametric.pdf.

[199] T.E. Seghier, Y.W. Lim, M.H. Ahmad, W.O. Samuel, Building Envelope Thermal Performance Assessment Using Visual Programming and BIM, based on ETTV requirement of Green Mark and GreenRE, Int. J. Built Environ. Sustain. 4 (2017) 227–235. https://doi.org/10.11113/ijbes.v4.n3.216.

[200] K. Lagios, J. Niemasz, C.F. Reinhart, Animated Building Performance Simulation (Abps) – Linking Rhinoceros/Grasshopper With Radiance/Daysim, SimBuild 2010. (321AD) 321–327. http://www.ibpsa.us/pub/simbuild2010/papers/SB10-DOC-TS06A-03-Lagios.pdf.

[201] E. Touloupaki, T. Theodosiou, Energy Performance Optimization as a Generative Design Tool for Nearly Zero Energy Buildings, Procedia Eng. 180 (2017) 1178–1185. https://doi.org/10.1016/j.proeng.2017.04.278.

[202] L. Sardone, R. Greco, A. Fiore, C. Moccia, D. De Tommasi, N.D. Lagaros, A preliminary study on a variable section beam through Algorithm-Aided Design: A way to connect architectural shape and structural optimization, Procedia Manuf. 44 (2020) 497–504. https://doi.org/10.1016/j.promfg.2020.02.264.

[203] P. Dong, Q. Chen, LiDAR Remote Sensing and Applications, 2018.

[204] Y. Xie, J. Tian, X.X. Zhu, Linking Points With Labels in 3D, IEEE Geosci. Remote Sens. Mag. (2020).

[205] G. Vosselman, H.-G. Maas, Airborne and Terrestrial Laser Scanning, 2010.

[206] Historic England, 3D Laser Scanning for Heritage: Advice and Guidance on the Use of Laser Scanning in

Archaeology and Architecture, 2018. https://historicengland.org.uk/images-books/publications/3d-laser-scanning-heritage/heag155-3d-laser-scanning/.

[207] B. Muralikrishnan, Performance evaluation of terrestrial laser scanners - A review, Meas. Sci. Technol. 32 (2021). https://doi.org/10.1088/1361-6501/abdae3.

[208] Q. Wang, Y. Tan, Z. Mei, Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications, Arch. Comput. Methods Eng. 27 (2020) 479–499. https://doi.org/10.1007/s11831-019-09320-4.

[209] M.A. Fischler, R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography, Graph. Image Process. 24 (1981) 381–395.

[210] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum. 26 (2007) 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x.

[211] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design, 3D Res. 2 (2011) 1–13. https://doi.org/10.1007/3DRes.02(2011)3.

[212] C. Dalitz, T. Schramke, M. Jeltsch, Iterative hough transform for line detection in 3D point clouds, Image Process. Line. 7 (2017) 184–196. https://doi.org/10.5201/ipol.2017.208.

[213] J.W. Ma, T. Czerniawski, F. Leite, Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds, Autom. Constr. 113 (2020) 103144. https://doi.org/10.1016/j.autcon.2020.103144.

[214] C. Rausch, C. Haas, Automated shape and pose updating of building information model elements from 3D point clouds, Autom. Constr. 124 (2021) 103561. https://doi.org/10.1016/j.autcon.2021.103561.

[215] B. Gao, Y. Pan, C. Li, S. Geng, H. Zhao, Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods, IEEE Trans. Intell. Transp. Syst. 23 (2021) 6063–6081. https://doi.org/10.1109/TITS.2021.3076844.

[216] B. Wu, A. Wan, X. Yue, K. Keutzer, SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud, Proc. - IEEE Int. Conf. Robot. Autom. (2018) 1887–1893. https://doi.org/10.1109/ICRA.2018.8462926.

[217] B. Wu, X. Zhou, S. Zhao, X. Yue, K. Keutzer, SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud, Proc. - IEEE Int. Conf. Robot. Autom. 2019-May (2019) 4376–4382. https://doi.org/10.1109/ICRA.2019.8793495.

[218] M.C. Güneş, A. Mertan, Y.H. Sahin, G. Unal, M. Özkar, Synthesizing Point Cloud Data Set for Historical Dome Systems, Commun. Comput. Inf. Sci. 1465 CCIS (2022) 538–554. https://doi.org/10.1007/978-981-19-1280-1_33.

[219] Y. Zhou, O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2018) 4490–4499. https://doi.org/10.1109/CVPR.2018.00472.

[220] H.Y. Meng, L. Gao, Y.K. Lai, Di. Manocha, VV-net: Voxel VAE net with group convolutions for point cloud segmentation, Proc. IEEE Int. Conf. Comput. Vis. 2019-Octob (2019) 8499–8507. https://doi.org/10.1109/ICCV.2019.00859.

[221] H. Radi, W. Ali, VolMap: A Real-time Model for Semantic Segmentation of a LiDAR surrounding view, (2019). http://arxiv.org/abs/1906.11873.

[222] R.Q. Charles, H. Su, M. Kaichun, L.J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: 2017 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2017: pp. 77–85. https://doi.org/10.1109/CVPR.2017.16.

[223] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, PointCNN: Convolution on X-transformed points, Adv. Neural Inf. Process. Syst. 2018-Decem (2018) 820–830.

[224] W. Wu, Z. Qi, L. Fuxin, PointCONV: Deep convolutional networks on 3D point clouds, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2019-June (2019) 9613–9622. https://doi.org/10.1109/CVPR.2019.00985.

[225] H. Thomas, C.R. Qi, J.E. Deschaud, B. Marcotegui, F. Goulette, L. Guibas, KPConv: Flexible and deformable convolution for point clouds, Proc. IEEE Int. Conf. Comput. Vis. 2019-Octob (2019) 6410–6419. https://doi.org/10.1109/ICCV.2019.00651.

[226] Y. Zhou, A. Ji, L. Zhang, Sewer defect detection from 3D point clouds using a transformer-based deep learning model, Autom. Constr. 136 (2022) 104163. https://doi.org/10.1016/j.autcon.2022.104163.

[227] T. Czerniawski, F. Leite, Automated digital modeling of existing buildings: A review of visual object recognition methods, Autom. Constr. 113 (2020) 103131. https://doi.org/10.1016/j.autcon.2020.103131.

[228] L. Yang, J.C.P. Cheng, Q. Wang, Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data, Autom. Constr. 112 (2020) 103037. https://doi.org/10.1016/j.autcon.2019.103037.

[229] B. Wang, C. Yin, H. Luo, J.C.P. Cheng, Q. Wang, Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data, Autom. Constr. 125 (2021) 103615. https://doi.org/10.1016/j.autcon.2021.103615.

[230] H. Tran, K. Khoshelham, A. Kealy, L. Díaz-Vilariño, Shape Grammar Approach to 3D Modeling of Indoor Environments Using Point Clouds, J. Comput. Civ. Eng. 33 (2019) 1–14. https://doi.org/10.1061/(asce)cp.1943-5487.0000800.

[231] D. Stutz, A. Geiger, Learning 3D Shape Completion Under Weak Supervision, Int. J. Comput. Vis. 128 (2020) 1162–1181. https://doi.org/10.1007/s11263-018-1126-y.

[232] Y. Xu, S. Tuttas, L. Hoegner, U. Stilla, Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor, Autom. Constr. 85 (2018) 76–95. https://doi.org/10.1016/j.autcon.2017.09.014.

[233] T. Czerniawski, J.W. Ma, Fernanda Leite, Automated building change detection with amodal completion of point clouds, Autom. Constr. 124 (2021) 103568. https://doi.org/10.1016/j.autcon.2021.103568.

[234] T.T. Tran, V.T. Cao, D. Laurendeau, Extraction of cylinders and estimation of their parameters from point clouds, Comput. Graph. 46 (2015) 345–357. https://doi.org/10.1016/j.cag.2014.09.027.

[235] Z. Zhang, J. Li, X. Li, Y. Lin, S. Zhang, C. Wang, A fast method for measuring the similarity between 3d model and 3d point cloud, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 2016-Janua (2016) 725–728. https://doi.org/10.5194/isprsarchives-XLI-B1-725-2016.

[236] H. Tran, T.N. Nguyen, P. Christopher, D.K. Bui, K. Khoshelham, T.D. Ngo, A digital twin approach for geometric quality assessment of as-built prefabricated façades, J. Build. Eng. 41 (2021). https://doi.org/10.1016/j.jobe.2021.102377.

[237] Y. Yan, J.F. Hajjar, Automated extraction of structural elements in steel girder bridges from laser point clouds, Autom. Constr. 125 (2021) 103582. https://doi.org/10.1016/j.autcon.2021.103582.

[238] R. Lu, I. Brilakis, C.R. Middleton, Detection of Structural Components in Point Clouds of Existing RC Bridges, Comput. Civ. Infrastruct. Eng. 34 (2019) 191–212. https://doi.org/10.1111/mice.12407.

[239] F. Jiang, L. Ma, T. Broyd, K. Chen, Digital twin and its implementations in the civil engineering sector, Autom. Constr. 130 (2021) 103838. https://doi.org/10.1016/j.autcon.2021.103838.

[240] C. Zhuang, J. Liu, H. Xiong, Digital twin-based smart production management and control framework for the complex product assembly shop-floor, Int. J. Adv. Manuf. Technol. 96 (2018) 1149–1163. https://doi.org/10.1007/s00170-018-1617-6.

[241] T. Deng, K. Zhang, Z.J. (Max) Shen, A systematic review of a digital twin city: A new pattern of urban governance toward smart cities, J. Manag. Sci. Eng. 6 (2021) 125–134. https://doi.org/10.1016/j.jmse.2021.03.003.

[242] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 2020-Decem (2020).

[243] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, J. Schulman, WebGPT: Browser-assisted

question-answering with human feedback, (2021). http://arxiv.org/abs/2112.09332.

[244] Chip Huyen, Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications, O'Reilly Media, 2022.

[245] B. Fuller, Critical Path, St. Martin's Press, 1981.

[246] W. Kinsner, R. Saracco, Towards Evolving Symbiotic Cognitive Education Based on Digital Twins, Proc. 2019 IEEE 18th Int. Conf. Cogn. Informatics Cogn. Comput. ICCI*CC 2019. (2019) 13–21. https://doi.org/10.1109/ICCICC46617.2019.9146095.

[247] D. Remund, D. "Deb" Aikat, Drowning in Data: A Review of Information Overload within Organizations and the Viability of Strategic Communication Principles, in: Inf. Overload, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2012: pp. 231–250. https://doi.org/10.1002/9781118360491.ch11.

[248] C. Newport, Deep Work: Rules for Focused Success in a Distracted World, Grand Central Publishing, 2016.

[249] C. Newport, Digital Minimalism: Choosing a Focused Life in a Noisy World, Portfolio, 2019.

[250] Paresh Chandra Deka, A Primer on Machine Learning Applications in Civil Engineering, CRC Press, 2019. https://doi.org/https://doi.org/10.1201/9780429451423.

[251] L. Atzori, A. Iera, G. Morabito, SIoT: Giving a social structure to the internet of things, IEEE Commun. Lett. 15 (2011) 1193–1195. https://doi.org/10.1109/LCOMM.2011.090911.111340.

[252] L. Atzori, A. Iera, G. Morabito, M. Nitti, The social internet of things (SIoT) - When social networks meet the internet of things: Concept, architecture and network characterization, Comput. Networks. 56 (2012) 3594–3608. https://doi.org/10.1016/j.comnet.2012.07.010.

[253] D. Guinard, M. Fischer, V. Trifa, Sharing using social networks in a composable Web of Things, 2010 8th IEEE Int. Conf. Pervasive Comput. Commun. Work. PERCOM Work. 2010. (2010) 702–707. https://doi.org/10.1109/PERCOMW.2010.5470524.

[254] K.M. Alam, M. Saini, A. El Saddik, tNote: A Social Network of Vehicles under Internet of Things, in: Internet Veh. – Technol. Serv., 2014: pp. 227–236. https://doi.org/10.1007/978-3-319-11167-4_23.

[255] R. Vinuesa, H. Azizpour, I. Leite, M. Balaam, V. Dignum, S. Domisch, A. Felländer, S.D. Langhans, M. Tegmark, F. Fuso Nerini, The role of artificial intelligence in achieving the Sustainable Development Goals, Nat. Commun. 11 (2020) 1–10. https://doi.org/10.1038/s41467-019-14108-y.

[256] F. Arena, S. Curto, S. Trubia, L. Puleo, A. Severino, The use of BIM for infrastructures, in: 2021: p. 110011. https://doi.org/10.1063/5.0047871.

[257] ISO 19650-1:2018. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles, n.d.

[258] ISO 19650-2:2018. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 2: Delivery phase of the assets, n.d.

[259] ISO 19650-4:2022. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 4: Information exchange, n.d.

[260] ISO 19650-5:2020. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 5: Security-minded approach to , n.d.

[261] ISO/CD 19650-6. Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 6: Health and Safety, n.d.

[262] National BIM Standard - United States V3, n.d.

[263] H. Chen, L. Hou, G. (Kevin) Zhang, S. Moon, Development of BIM, IoT and AR/VR technologies for fire safety and upskilling, Autom. Constr. 125 (2021). https://doi.org/10.1016/j.autcon.2021.103631.

[264] J. Barnabas, P. Raj, The human body: A digital twin of the cyber physical systems, in: Adv. Comput., 1st ed., Elsevier Inc., 2020: pp. 219–246. https://doi.org/10.1016/bs.adcom.2019.09.004.

[265] M. Ryan, The social and ethical impacts of artificial intelligence in agriculture: mapping the agricultural AI literature, AI Soc. (2022). https://doi.org/10.1007/s00146-021-01377-9.

[266] P.G. Kelley, Y. Yang, C. Heldreth, C. Moessner, A. Sedley, A. Kramm, D.T. Newman, A. Woodruff, Exciting, Useful, Worrying, Futuristic: Public Perception of Artificial Intelligence in 8 Countries, Association for Computing Machinery, 2021. https://doi.org/10.1145/3461702.3462605.

[267] I. Štulec, K. Petljak, D. Naletina, Weather impact on retail sales: How can weather derivatives help with adverse weather deviations?, J. Retail. Consum. Serv. 49 (2019) 1–10. https://doi.org/10.1016/j.jretconser.2019.02.025.

[268] Y. Sun, S. Wang, Weather Impacts on Stock Market, Int. Conf. Educ. Econ. Manag. Res. 95 (2017) 491–494.

[269] R. Söderberg, K. Wärmefjord, J.S. Carlson, L. Lindkvist, Toward a Digital Twin for real-time geometry assurance in individualized production, CIRP Ann. - Manuf. Technol. 66 (2017) 137–140. https://doi.org/10.1016/j.cirp.2017.04.038.

[270] M. Fujiu, T. Minami, J. Takayama, Environmental Influences on Bridge Deterioration Based on Periodic Inspection Data from Ishikawa Prefecture, Japan, Infrastructures. 7 (2022). https://doi.org/10.3390/infrastructures7100130.

[271] R.C. Martin, Agile Software Development, Principles, Patterns, and Practices, Pearson, 2002.

[272] R.C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Financial Times Prentice Hall, 2008.

[273] A. Liew, DIKIW: Data, information, knowledge, intelligence, wisdom and their interrelationships, Bus. Manag. Dyn. 2 (2013) 49–62.

[274] H. Adeli, K.C. Sarma, Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing, John Wiley & Sons, Ltd, Chichester, UK, 2006. https://doi.org/10.1002/0470867353.

[275] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proc. ICNN'95 - Int. Conf. Neural Networks, IEEE, 1995: pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[276] S. Kirkpatrick, C.D.J. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Science (80-.). 220 (1983) 671–680. https://doi.org/10.1126/science.220.4598.671.

[277] D.E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, 1989.

[278] Z. Aydın, Y. Ayvaz, Optimum topology and shape design of prestressed concrete bridge girders using a genetic algorithm, Struct. Multidiscip. Optim. 41 (2010) 151–162. https://doi.org/10.1007/s00158-009-0404-2.

[279] O. Skoglund, J. Leander, R. Karoumi, Optimizing the steel girders in a high strength steel composite bridge, Eng. Struct. 221 (2020) 110981. https://doi.org/10.1016/j.engstruct.2020.110981.

[280] R.L. Pedro, J. Demarche, L.F.F. Miguel, R.H. Lopez, An efficient approach for the optimization of simply supported steel-concrete composite I-girder bridges, Adv. Eng. Softw. 112 (2017) 31–45. https://doi.org/10.1016/j.advengsoft.2017.06.009.

[281] A.M.B. Martins, L.M.C. Simões, J.H.J.O. Negrão, Optimization of cable-stayed bridges: A literature survey, Adv. Eng. Softw. 149 (2020) 102829. https://doi.org/10.1016/j.advengsoft.2020.102829.

[282] V. Lute, A. Upadhyay, K.K. Singh, Genetic Algorithms-based Optimization of Cable Stayed Bridges, J. Softw. Eng. Appl. 4 (2011) 571–578. https://doi.org/10.4236/jsea.2011.410066.

[283] J.H.J.O. Negrão, L.M.C. Simões, Optimization of cable-stayed bridges with three-dimensional modelling, Comput. Struct. 64 (1997) 741–758. https://doi.org/10.1016/S0045-7949(96)00166-6.

[284] A. Baldomir, S. Hernandez, F. Nieto, J.Á. Jurado, Cable optimization of a long span cable stayed bridge in la Coruña (Spain), Adv. Eng. Softw. 41 (2010) 931–938. https://doi.org/10.1016/j.advengsoft.2010.05.001.

[285] M.M. Hassan, A.A. El Damatty, A.O. Nassef, Database for the optimum design of semi-fan composite cable-stayed bridges based on genetic algorithms, Struct. Infrastruct. Eng. 11 (2015) 1054–1068. https://doi.org/10.1080/15732479.2014.931976.

[286] M.M. Hassan, Optimization of stay cables in cable-stayed bridges using finite element, genetic algorithm, and B-spline combined technique, Eng. Struct. 49 (2013) 643–654. https://doi.org/10.1016/j.engstruct.2012.11.036.

[287] J. Guo, W. Yuan, X. Dang, M.S. Alam, Cable force optimization of a curved cable-stayed bridge with combined

simulated annealing method and cubic B-Spline interpolation curves, Eng. Struct. 201 (2019) 109813. https://doi.org/10.1016/j.engstruct.2019.109813.

[288] M.-H. Ha, Q.-A. Vu, V.-H. Truong, Optimum Design of Stay Cables of Steel Cable-stayed Bridges Using Nonlinear Inelastic Analysis and Genetic Algorithm, Structures. 16 (2018) 288–302. https://doi.org/10.1016/j.istruc.2018.10.007.

[289] V. Lute, A. Upadhyay, K.K. Singh, Computationally efficient analysis of cable-stayed bridge for GA-based optimization, Eng. Appl. Artif. Intell. 22 (2009) 750–758. https://doi.org/10.1016/j.engappai.2009.04.001.

[290] V. Srinivas, K. Ramanjaneyulu, An integrated approach for optimum design of bridge decks using genetic algorithms and artificial neural networks, Adv. Eng. Softw. 38 (2007) 475–487. https://doi.org/10.1016/j.advengsoft.2006.09.016.

[291] J. Cheng, Optimum design of steel truss arch bridges using a hybrid genetic algorithm, J. Constr. Steel Res. 66 (2010) 1011–1017. https://doi.org/10.1016/j.jcsr.2010.03.007.

[292] V. Toğan, A. Daloğlu, Design and Reliability Based Optimization of a 2D Arch Bridge, J. Eng. Nat. Sci. 25 (2006) 17–26.

[293] J. Cheng, H. Jin, Reliability-based optimization of steel truss arch bridges, Int. J. Steel Struct. 17 (2017) 1415–1425. https://doi.org/10.1007/s13296-017-1212-y.

[294] N. Islam, S. Rana, R. Ahsan, S. Ghani, An Optimized Design of Network Arch Bridge using Global Optimization Algorithm, Adv. Struct. Eng. 17 (2014) 197–210. https://doi.org/10.1260/1369-4332.17.2.197.

[295] M.A. Latif, M.P. Saka, Optimum design of tied-arch bridges under code requirements using enhanced artificial bee colony algorithm, Adv. Eng. Softw. 135 (2019) 102685. https://doi.org/10.1016/j.advengsoft.2019.102685.

[296] M. Pouraminian, M. Ghaemian, Optimalizacija oblika lučnih mostova s otvorenim nadlučnim sklopom, J. Croat. Assoc. Civ. Eng. 67 (2015) 1177–1185. https://doi.org/10.14256/jce.1223.2015.

[297] D.M. Peng, C.A. Fairfield, Optimal design of arch bridges by integrating genetic algorithms and the mechanism method, Eng. Struct. 21 (1999) 75–82. https://doi.org/10.1016/S0141-0296(97)00124-7.

[298] H. Furuta, K. Maeda, E. Watanabe, Application of Genetic Algorithm to Aesthetic Design of Bridge Structures, Microcomput. Civ. Eng. 10 (1995) 415–421.

[299] M.Z. Abd Elrehim, M.A. Eid, M.G. Sayed, Structural optimization of concrete arch bridges using Genetic Algorithms, Ain Shams Eng. J. 10 (2019) 507–516. https://doi.org/10.1016/j.asej.2019.01.005.

[300] J. Park, Y.H. Chun, J. Lee, Optimal design of an arch bridge with high performance steel for bridges using genetic algorithm, Int. J. Steel Struct. 16 (2016) 559–572. https://doi.org/10.1007/s13296-016-6024-y.

[301] D. Bruno, P. Lonetti, A. Pascuzzo, An optimization model for the design of network arch bridges, Comput. Struct. 170 (2016) 13–25. https://doi.org/10.1016/j.compstruc.2016.03.011.

[302] V. Houšt', J. Eliáš, L. Miča, Shape optimization of concrete buried arches, Eng. Struct. 48 (2013) 716–726. https://doi.org/10.1016/j.engstruct.2012.11.037.

[303] M. Whelan, N. Salas Zamudio, T. Kernicky, Structural identification of a tied arch bridge using parallel genetic algorithms and ambient vibration monitoring with a wireless sensor network, J. Civ. Struct. Heal. Monit. 8 (2018) 315–330. https://doi.org/10.1007/s13349-017-0266-z.

[304] K. Deb, A. Kumar, Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems, Complex Syst. 9 (1995) 431–454.

[305] K. Deb, An Efficient Constraint Handling Method for Genetic Algorithms, Comput. Methods Appl. Mech. Eng. 186 (2000) 311–338. https://doi.org/10.1016/S0045-7825(99)00389-8.

[306] A. Reese, Random number generators in genetic algorithms for unconstrained and constrained optimization, Nonlinear Anal. Theory, Methods Appl. 71 (2009) e679–e692. https://doi.org/10.1016/j.na.2008.11.084.

[307] K. Deb, R.B. Agrawal, Simulated Binary Crossover for Continuous Search Space, Complex Syst. 9 (1994) 115–148.

[308] K. Deb, D. Deb, Analysing mutation schemes for real-parameter genetic algorithms, Int. J. Artif. Intell. Soft Comput. 4 (2014) 1–15. https://doi.org/10.1504/ijaisc.2014.059280.

[309] epsi1on, BriefFiniteElement.Net, (n.d.).

[310] Eurocode 0: Basis of structural design, n.d.

[311] Eurocode 3: Design of steel structures - Part 1-5: General rules - Plated structural elements, n.d.

[312] Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings, n.d.

[313] Eurocode 3: Design of steel structures - Part 2: Steel bridges, n.d.

[314] I. Caetano, L. Santos, A. Leitao, Computational design in architecture: Defining parametric, generative, and algorithmic design, Front. Archit. Res. (2019) 1–14.

[315] T. Czerniawski, F. Leite, 3DFacilities: Annotated 3D reconstructions of building facilities, Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-91635-4_10.

[316] R. Pierdicca, M. Mameli, E.S. Malinverni, M. Paolanti, E. Frontoni, Automatic generation of point cloud synthetic dataset for historical building representation, Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-25965-5_16.

[317] J. Fang, D. Zhou, F. Yan, T. Zhao, F. Zhang, Y. Ma, L. Wang, R. Yang, Augmented LiDAR simulator for autonomous driving, IEEE Robot. Autom. Lett. 5 (2020) 1931–1938. https://doi.org/10.1109/LRA.2020.2969927.

[318] C. Morbidoni, R. Pierdicca, M. Paolanti, R. Quattrini, R. Mammoli, Learning from synthetic point cloud data for historical buildings semantic segmentation, J. Comput. Cult. Herit. 13 (2020). https://doi.org/10.1145/3409262.

[319] R. Zhai, J. Zou, Y. He, L. Meng, BIM-driven data augmentation method for semantic segmentation in superpoint-based deep learning network, Autom. Constr. 140 (2022) 104373. https://doi.org/10.1016/j.autcon.2022.104373.

[320] Y. Xu, S. Arai, F. Tokuda, K. Kosuge, A Convolutional Neural Network for Point Cloud Instance Segmentation in Cluttered Scene Trained by Synthetic Data without Color, IEEE Access. 8 (2020) 70262–70269. https://doi.org/10.1109/ACCESS.2020.2978506.

[321] L. Ma, R. Sacks, R. Zeibak-Shini, A. Aryal, S. Filin, Preparation of Synthetic As-Damaged Models for Post-Earthquake BIM Reconstruction Research, J. Comput. Civ. Eng. 30 (2016) 1–12. https://doi.org/10.1061/(asce)cp.1943-5487.0000500.

[322] Y. Xu, U. Stilla, Toward Building and Civil Infrastructure Reconstruction from Point Clouds: A Review on Data and Key Techniques, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14 (2021) 2857–2885. https://doi.org/10.1109/JSTARS.2021.3060568.

[323] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-Decem (2016) 1534–1543. https://doi.org/10.1109/CVPR.2016.170.

[324] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Nießner, ScanNet: Richly-annotated 3D reconstructions of indoor scenes, Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017-Janua (2017) 2432–2443. https://doi.org/10.1109/CVPR.2017.261.

[325] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, C. Stachniss, Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset, Int. J. Rob. Res. 40 (2021) 959–967. https://doi.org/10.1177/02783649211006735.

[326] D. Griffiths, J. Boehm, SynthCity: A large scale synthetic point cloud, (2019) 1–6. https://doi.org/https://doi.org/10.48550/arXiv.1907.04758.

[327] J.-E. Deschaud, KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator, (2021) 1–2. https://doi.org/https://doi.org/10.48550/arXiv.2109.00892.

[328] J.E. Deschaud, D. Duque, J.P. Richa, S. Velasco-Forero, B. Marcotegui, F. Goulette, Paris-carla-3d: A real and synthetic outdoor point cloud dataset for challenging tasks in 3d mapping, Remote Sens. 13 (2021) 1–24. https://doi.org/10.3390/rs13224713.

[329] G. Uggla, M. Horemuz, Towards synthesized training data for semantic segmentation of mobile laser scanning point clouds: Generating level crossings from real and synthetic point cloud samples, Autom. Constr. 130 (2021) 103839. https://doi.org/10.1016/j.autcon.2021.103839.

[330] T.J. Purcell, I. Buck, W.R. Mark, P. Hanrahan, Ray tracing on programmable graphics hardware, ACM Trans.

Graph. 21 (2002) 703–712. https://doi.org/10.1145/566654.566640.

[331] J.W. Ma, B. Han, F. Leite, An Automated Framework for Generating Synthetic Point Clouds from as-Built BIM with Semantic Annotation for Scan-to-BIM, Proc. - Winter Simul. Conf. 2021-Decem (2021). https://doi.org/10.1109/WSC52266.2021.9715301.

[332] BlenSor. Blender Sensor Simulation, (n.d.). blensor.org.

[333] Blender. Open source 3D creation suite, (n.d.). blender.org.

[334] B. Han, J.W. Ma, F. Leite, A framework for semi-automatically identifying fully occluded objects in 3D models: Towards comprehensive construction design review in virtual reality, Adv. Eng. Informatics. 50 (2021) 101398. https://doi.org/10.1016/j.aei.2021.101398.

[335] S. Bechtold, B. Höfle, Helios: a Multi-Purpose Lidar Simulation Framework for Research, Planning and Training of Laser Scanning Operations With Airborne, Ground-Based Mobile and Stationary Platforms, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. III–3 (2016) 161–168. https://doi.org/10.5194/isprsannals-iii-3-161-2016.

[336] F. Noichl, A. Braun, A. Borrmann, “BIM-to-Scan” for Scan-to-BIM: Generating Realistic Synthetic Ground Truth Point Clouds based on Industrial 3D Models, Proc. 2021 Eur. Conf. Comput. Constr. 2 (2021) 164–172. https://doi.org/10.35490/ec3.2021.166.

[337] Gazebo. Robotic design, development and simulation suite, (n.d.). gazebosim.org.

[338] J.L. Martínez, M. Morán, J. Morales, A. Robles, M. Sánchez, Supervised learning of natural-terrain traversability with synthetic 3D laser scans, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10031140.

[339] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An Open Urban Driving Simulator, in: 1st Conf. Robot Learn., 2017: pp. 1–16. http://arxiv.org/abs/1711.03938.

[340] F. Wang, Y. Zhuang, H. Gu, H. Hu, Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis, IEEE Trans. Instrum. Meas. 68 (2019) 2671–2673. https://doi.org/10.1109/TIM.2019.2906416.

[341] Unity. Real-Time Development Platform, (n.d.). unity.com.

[342] G.F. Gusmão, C.R.H. Barbosa, A.B. Raposo, Development and validation of lidar sensor simulators based on parallel raycasting, Sensors (Switzerland). 20 (2020) 1–18. https://doi.org/10.3390/s20247186.

[343] K.S. Henriksen, M.S. Lynge, M.D.B. Jeppesen, M.M.J. Allahham, I.A. Nikolov, J.B. Haurum, T.B. Moeslund, Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation, Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-58468-9_26.

[344] C. Suchocki, Comparison of time-of-flight and phase-shift TLS intensity data for the diagnostics measurements of buildings, Materials (Basel). 13 (2020). https://doi.org/10.3390/ma13020353.

[345] P. Vacek, O. Jasek, K. Zimmermann, T. Svoboda, Learning to Predict Lidar Intensities, IEEE Trans. Intell. Transp. Syst. 23 (2022) 3556–3564. https://doi.org/10.1109/TITS.2020.3037980.

[346] X. Yue, B. Wu, S.A. Seshia, K. Keutzer, A.L. Sangiovanni-Vincentelli, A LiDAR point cloud generator: From a virtual world to autonomous driving, ICMR 2018 - Proc. 2018 ACM Int. Conf. Multimed. Retr. (2018) 458–464. https://doi.org/10.1145/3206025.3206080.

[347] R. Lu, I. Brilakis, Recursive Segmentation for As-Is Bridge Information Modelling, (2017) 209–217. https://doi.org/10.24928/jc3-2017/0020.

[348] R. Schmidt, geometry3Sharp. Open-Source c# library for geometric computing, (n.d.). github.com/gradientspace/geometry3Sharp.

[349] Dynamo. Open Source Graphical Programming for Design, (n.d.). github.com/DynamoDS/Dynamo.

[350] Autodesk Revit. BIM software, (n.d.). autodesk.com/products/revit.

[351] B. Riveiro, R. Lindenbergh, Laser Scanning. An Emerging Technology in Structural Engineering, CRC Press, 2019. https://doi.org/10.1201/9781351018869.

[352] T.P. Kersten, M. Lindstaedt, Geometric accuracy investigations of terrestrial laser scanner systems in the laboratory and in the field, Appl. Geomatics. 14 (2022) 421–434. https://doi.org/10.1007/s12518-022-00442-2.

[353] B. Höfle, N. Pfeifer, Correction of laser scanning intensity data: Data and model-driven approaches, ISPRS J. Photogramm. Remote Sens. 62 (2007) 415–433. https://doi.org/10.1016/j.isprsjprs.2007.05.008.

[354] N. Pfeifer, P. Dorninger, A. Haring, H. Fan, Investigating terrestrial laser scanning intensity data: Quality and functional relations, 8th Conf. Opt. 3-D Meas. Tech. (2007) 328–337.

[355] R. Blaskow, D. Schneider, Analysis and correction of the dependency between laser scanner intensity values and range, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 40 (2014) 107–112. https://doi.org/10.5194/isprsarchives-XL-5-107-2014.

[356] CloudCompare. 3D point cloud and mesh processing software. Open Source Project, (n.d.). danielgm.net/cc.

[357] Z+F Imager 5010 scanner specification, (n.d.). geo-matching.com/terrestrial-laser-scanners/z-f-imager-5010.

Acknowledgments

My PhD studies were a time of great professional and personal growth. During these four years, I had the luxury of exploring areas of my deep interest and that I just enjoy. I have also been surrounded by people that have been forming me as a scientist, professionalist, and person. Therefore, I will take this unusual opportunity and try to express my gratitude with a (wide) circle of thanks.

I want to thank my supervisor, Marek Salamak, who, as an open scientist, is an important figure in my research and, as a kind and reliable person, in my personal development. I would also like to thank my additional supervisor, Jan Winkler, as well as colleagues from the Department of Mechanics and Bridges of the Silesian University of Technology, especially those with whom I have been working most: Marcin Jasiński, Piotr Łaziński, Stefan Pradelok, Piotr Bętkowski, Grzegorz Poprawa, Tomasz Płaszczyk, Duc Nguyen, Muhammad Fawad, Wiktoria Drzyzga, Mateusz Uściłowski, Jerzy Weseli. Given your expertise and kindness, you are a great group to work with.

During my doctoral studies, I had a chance to work with international researchers. I want to thank Thomas Czerniawski, Mohit Gupta, Chialing Wei, and Hala Sanboskani for my semester at Arizona State University. I would also like to thank the authorities of the Budapest University of Technology and Economics and the Mientrung University of Civil Engineering for their willingness to host me in past and future projects.

I had a chance to cooperate, both within professional and academic activities, with several companies. I want to thank friends from CADmost, BIMind, Autodesk, Digital Engineering Solutions, and 3DFormat. The industrial, practical perspective is of great value for academic research.

I want to thank the people who are there for me. I would like to thank my family, especially the closest: mother Sabina, father Grzegorz; grandparents Anna, Polcia, and Bolek; Aneta, Karina, Zosia, Jacek, and particularly my sisters: Daga (the greatest lifelong companion) and Magda (a wonderful young woman, making me proud of being an older brother). I would also like to thank all my fantastic friends, both those who would be hosted in Domek Babci Gosi for games and those who would be met in Ziemowit's 116 for karaokes.

Finally, I want to thank Kinga - the first reviewer and greatest supporter of each of my ideas.

I am grateful that I can list so many people here.

Thank you a lot,

Kamil

Abstract

The ongoing digitization presents both challenges and opportunities. Industries are transforming at an unprecedented pace. Sustainable development goals, ecological aims, and a growing population motivate the introduction of modern and integrated techniques also in civil engineering. The need for digital transformation is expressed by the problems of aging infrastructure, the civil engineering productivity margin, and the expectations of engineers and managers of facilities. Bridges, as complex, unique structures of logistic and strategic importance, need a new strategy of management. Digital twins, announced as crucial components of Industry 4.0, have become an enabler of holistic digital transformation.

This dissertation proposes principles for creating and utilizing the digital twins of bridges. The proposed concept addresses the general multi-industrial digital twinning fundamentals (identified with the literature study), the specificity of bridges, and the demands of designers, engineers, and managers of civil engineering facilities. The proposed digital twin is a virtual counterpart of the bridge in its entire lifecycle characterized by actuality, intelligence and autonomy, interaction, interoperability, modularity, expansibility and scalability, accessibility and security, and uniqueness. From the technological perspective, it is an evolution of current practices, which have not yet been properly integrated. The proposed digital twin utilizes BIM (Building Information Modeling), SHM (Structural Health Monitoring), and AI (Artificial Intelligence). It states IFC (Industry Foundation Classes) as the base for the central model. It is also enhanced by other techniques (e.g., visual programming, point clouds). This approach enables the evolutionary process of adopting digital twins in civil engineering. Civil engineering digital twinning does not have to be perfect from the start. It should mature naturally, alluring designers, engineers, and managers of civil engineering facilities with its benefits.

To make this evolutionary development of digital twins possible, digital twinning must provide practical use cases giving clear benefits to engineers and managers of facilities. Therefore, this dissertation proposes two solutions created by the author, utilizing techniques identified as components of civil engineering digital twinning. The solutions regard various lifecycle phases and provide benefits on the design and operational phases of the objects' lifecycle.

The first proposed solution is the optimization using visual programming and genetic algorithm. Visual programming is used to link BIM modeling, FEM analysis, and optimization algorithm. The solution is a step towards utilizing digital twins' techniques in the design phase. The system is an effective geometrical optimization approach, assisting designers in their day-to-day tasks. Moreover, the integration of BIM and FEM models is crucial for creating a fully-functional digital twin of a bridge.

The second proposed technique is the generation of synthetic point clouds' datasets. Point clouds can be a base for modeling actual geometrical conditions in both operational and building phases. To automate the extraction of point clouds' information or even the generation of geometrical models, often machine learning algorithms are used. However, the available point clouds' datasets are not sufficient for the training of machine learning algorithms. The alternative is synthetic data. A scanning simulator created by the author

allows for generating synthetic point clouds from BIM models. Acquiring the data is the first but indispensable step in developing machine learning solutions automating the utilization of point clouds, including systems to monitor geometrical changes in the digital twin models. The scheduled scanning of the physical object and then updating the model with trained algorithms will create a model-based history of geometrical changes. Monitoring, identifying, and comparing the geometrical changes can help identify structural malfunctions. As for bridges, monitoring the geometrical changes is crucial, especially for objects on the ground deformation areas (e.g., mining or post-seismic activities, tunneling). The model updating can therefore lead to expert systems alarming about hazards.

The proposed in this dissertation concept of a digital twin of a bridge is one of the first approaches to establish the foundations for these data-rich models of bridges. The literature study revealed that research on digital twins of bridges is still in its infancy. The ongoing digital transformation is undoubtedly a challenge – but also an opportunity. Given the increasingly complex interconnections of scientific and industrial activities of various sectors, digital twinning improves the effectiveness of civil engineering and enables interdisciplinary cooperation in the digital dimension. The here-proposed principles and solutions are a brick towards the infrastructure of the future: reaching the modern world’s goals and delivering everyday value.

Extended abstract

1 Motivations

1.1 Adapt (or shape) to thrive

The world is always changing. This statement – although trivial – is a purpose of digital twinning research. The statement can be expanded. The world is always changing at an increasing pace. This is being validated especially in the last decade with the digital transformation following the concept of Industry 4.0.

Changes divide people into enthusiasts and pessimists, not only for moral and cultural concepts but also for technical innovations. During the Industrial Revolutions, people perceived the coming automation differently: some were excited about more comfortable lives; others were afraid of losing jobs and struggling in the new environment. As we know today, the groups that took the opportunities of technical advances shaped our world.

The ongoing digitization comes with many shifts: reality is being virtualized into a parallel world; people are constantly connected to the web, producing thoroughly analyzed data; artificial intelligence is taking over tasks formerly reserved for humans. The changes bring excitement and fear – both feelings are justified. However – following the maxim "you must develop not to stay in place" – people and industries, whether enthusiasts or pessimists, should be prepared for the changes of the digitization of our lives.

When juxtaposing "bridges" with terms like "artificial intelligence" or "virtual reality", bridges seem to be a technological relict of ancient times. After all, we are talking about static, monumental structures that, once built, can be left alone to stand for decades. But is it so? Not exactly. Bridges are constantly reacting to varying environmental conditions and forces. The resulting gradual degradation of bridges' condition can be unnoticed by users but is essential for managers and inspectors. Because of that, bridges must be monitored and maintained. Also, with new materials, structural solutions, building techniques, and complex equipment, bridges are still an area of civil engineering innovations. Nonetheless, the modern world's conditions push bridges to take the next steps into technical advances. Civil engineering is transforming to adapt to sustainable development goals, ecological aims, and a growing population.

To thrive, one must adapt to changes – especially in technological contexts. But adapting does not indicate a passive and mindless following. It may also mean shaping. In the digital world, industries become interconnected. This cooperation requires new principles and paradigms. Digital twins, a concept perceived as the key component of Industry 4.0, are becoming this multi-industrial paradigm. Digital twinning is already being applied, but its principles are still forming. The principles should be formed not only by programmers or IT engineers but, in the case of bridges, also by the civil engineering community. In that way, civil engineers will contribute to forming the principles of worldwide digitization, and digital twins of bridges will address the needs of designers, engineers, and managers of infrastructure.

1.2 Connectors of people and data

Bridges' impact is visible both in the day-to-day lives of individuals and on the scale of the entire countries and industries, which can be expressed with numbers; the numbers also reveal problems of the existing infrastructure. Only in the USA, there are 619 thousand of bridges, of which 224 thousand require major repair work or replacement [2]. The problem is expanded by its economic scale: the estimated cost of identified repairs is USD 260 billion – a value comparable to the gross domestic product of the Czech Republic (USD 282 billion), New Zealand (USD 250 billion), or Portugal (USD 250 billion).

The number and complexity of bridges increased the importance of managing the growing and also aging infrastructure network. It resulted in bridge management systems and standardization of maintenance, as well as processes of processing facilities' data. These should be adjusted to the novel technical capabilities of digital transformation. However, infrastructure managers still use older practices, despite their flaws: improper data management diminishes efficiency; condition assessment relies on periodic human inspections, displaying risks of omitting defects; aging infrastructure is unproperly maintained, leading to economic losses and posing the danger of catastrophes. The problems of existing infrastructure convince that civil engineering needs the plan to ensure resilience and safety through the entire lifecycle. The plan must consider the specificity of bridges and the requirements of today's world and also benefit from the technical advances.

In the digitized world, bridges evolve from static constructions into services providing multidimensional connection: physical – of communities, and digital – of data. This shift results from the requirements of facilities' managers and the need to ensure the safety and comfort of users. This forces connectivity and integration in the digital dimension, especially in acquiring, processing, and sharing data. Bridges will cooperate with other objects to empower global holistic management and to constitute the digital networks of smart cities. To enable that, we need a multi-industrial paradigm – the world is shifting towards digital twins.

1.3 Aims of the dissertation

The aim of this dissertation is to provide principles for creating and utilizing digital twins of bridges – the virtual counterparts of objects in their whole lifecycle. The proposed concept addresses the general principles of digital twins, the specificity of bridges, and the practices of the civil engineering industry. The practicality of the proposed solutions is upheld with novel use cases utilizing the techniques identified as components of the proposed digital twinning concept.

To reach the general aim of the dissertation, several intermediate aims have been realized.

- Assessment of the current stage of development of digital twins through the study of literature and analysis of the implementations in bridges.
- Identification of the primary techniques to create and utilize digital twins of bridges.
- The proposal of the primary characteristics of digital twins of bridges regarding their lifecycle.

- The implementation of a practical integration of BIM and FEM models in a visual programming environment, aiming to create a partial digital twin in the design phase with the use of an optimization algorithm to automate the design process.
- The implementation of a technique to generate synthetic point clouds, aiming to create and update geometric models of digital twins in the operational phase with machine learning algorithms.

The listed intermediate aims undoubtedly do not fulfill all the general digital twins' requirements defined in the literature. However, digital twins, in the current development stage, are still a forming concept, continually enhanced with new techniques and solutions. The approach presented in the dissertation enables the implementation of digital twins in an evolutionary way. Digital twins do not have to be complete and perfect from the beginning; the concept should mature naturally, alluring practitioners with its benefits. Other important factors are the limitations of time and the scope of the dissertation. The proposed description of the digital twins' characteristics states the primary principles, not the details of technical implementations. Also, the presented techniques for creating and utilizing digital twins are limited to show the potential benefits. Used technologies and solutions regard the selected needs of the phases of design (automation and optimization of the design processes) and maintenance (changes of the facilities' geometries in time). Nonetheless, the use case of generating the synthetic point clouds does not describe the whole process of creation and update of the models with machine learning algorithms but focuses on the first key step to the implementation of such solutions – the acquisition of data. Future work will regard the creation of machine learning algorithms to create and utilize digital twins.

2 Digital twins – the next wave in simulations

Digital twins are virtual counterparts of objects throughout their entire lifecycle. First adopted by NASA [3], today are being implemented cross-industrially, becoming a component of the digital transformation.

A digital twin, as a model coexisting with an object, should be initialized on a design stage, first reflecting the object's idea. With advances in design, it expands into a high-fidelity central data-management system. Then, it is used to plan and optimize production. Utilizing models for simulation and visualization during design and production is already a multi-industrial practice. But digital twinning aims to provide benefits through the entire object's lifecycle. Due to synchronization with the physical counterpart also on the operational stage, a digital twin exceeds the capabilities of isolated models. The virtual and physical counterparts collaborate. Data from the physical object update the virtual system. The virtual system affects the operation of the physical object.

The digital twin collects data to map processes. It registers how the object reacts to different conditions and how the reactions alter over time. Analyses of such big data lead to a better understanding of the object operation. The data also allows training intelligent algorithms to detect patterns and anomalies, evolving the digital twin into a system of growing autonomy in decision-making. The model reflecting the actual state of an element allows extended reliable simulations. Since digital twinning is a cross-sectoral framework, unified data systems allow holistic management, considering relationships of cooperating elements. It leads to global optimizations of economic, ecological, and social costs, keeping the safety of the objects.

Digital twins' interest increases in the academic environment, resulting in a dynamic growth of publications, and in the industry, leading to new implementations. The research is dualistic: both theoretical frameworks and definitions are being established, and practical use cases are being explored. Both aspects have been regarded in this dissertation.

3 The concept of digital twins of bridges

Establishing the foundations for bridge digital twins must start with a vision. But equally important are the enablers for realization. There is no point in forming idealized concepts that are perfect in theory but inapplicable in practice. To develop, digital twinning must be adopted by civil engineering practitioners, so designers, engineers, and facilities' managers. The digital twinning framework should, therefore, rather guide into practical benefits than restrain with idealized rules.

The first civil engineering applications of digital twins utilize modern civil engineering techniques (e.g., BIM, SHM, 3D reconstruction), often blending them and integrating with, e.g., artificial intelligence. The use cases are still partial. They do not implement complete digital twinning but focus on selected functionalities (e.g., isolated semantic models or acquiring SHM data). Nonetheless, the applications promote the automation and digitization of civil engineering, making a ground for digital twins.

This dissertation proposes principles for creating and utilizing the digital twins of bridges (Fig. 1). The proposed concept addresses the general multi-industrial digital twinning fundamentals (identified with the literature study), the specificity of bridges, and the demands of designers, engineers, and managers of civil engineering facilities. The proposed digital twin is a virtual counterpart of the bridge in its entire lifecycle characterized by actuality, intelligence and autonomy, interaction, interoperability, modularity, expansibility and scalability, accessibility and security, and uniqueness. From the technological perspective, it is an evolution of current practices, which have not yet been properly integrated. The proposed digital twin utilizes BIM (Building Information Modeling), SHM (Structural Health Monitoring), and AI (Artificial Intelligence). It states IFC (Industry Foundation Classes) as the base for the central model. It is also enhanced by other techniques (e.g., visual programming, point clouds). This approach enables the evolutionary process of adopting digital twins in civil engineering. Civil engineering digital twinning does not have to be perfect from the start. It should mature naturally, alluring designers, engineers, and managers of civil engineering facilities with its benefits.

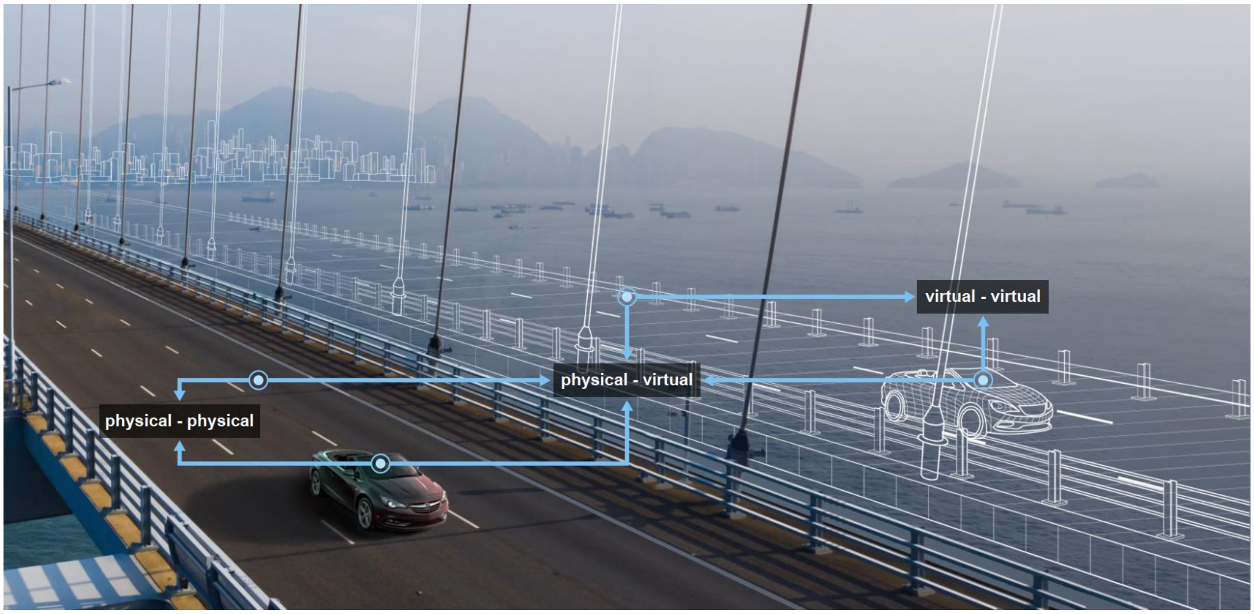


Fig. 1. A bridge digital twin vision

3.2 Techniques of civil engineering digital twinning

This subchapter lists techniques identified as crucial for creating and utilizing digital twins of civil engineering facilities. The listed techniques are:

- Building Information Modeling (BIM),
- Industry Foundation Classes (IFC),
- structural health monitoring (SHM),
- artificial intelligence (AI),
- visual programming (VP),
- point clouds.

The authors describing the implementations of the listed techniques often refer to digital twins. This is understandable, especially given the initial phase of the development of this concept. Nonetheless, calling the implementation of an isolated technique a complete digital twin is not entirely correct. The techniques can be components of civil engineering digital twins, but only their conscious interoperable-oriented mixture enables extensive digital twinning. The BIM model or SHM system is not yet a digital twin. This subchapter describes the differences between these techniques and their use in creating and utilizing digital twins of bridges.

3.3 Notions towards digital twinning of bridges

The dissertation presents notions towards the digital twinning of bridges. The notions complement techniques and characteristics and together can be perceived as an extended definition to guide creating, utilizing, and interpreting digital twins of bridges. The notions (elaborated in this subchapter) are:

- Digital twins allow business thriving in the future;

- Digital twins’ frameworks should be object-specific;
- Ultra-high fidelity is not always required;
- Data synchronization does not need to be perfect;
- Digital twin should not neglect any data freely;
- Digital twin provides data interfaces;
- Digital twin includes processes;
- Digital twin is intelligent, autonomous;
- Digital twin is a system;
- Digital twins widen empirical science;
- Digital twins must provide practical benefits;
- It is the right time to establish digital twinning frameworks.

3.4 Characteristics of digital twins of bridges

The dissertation describes characteristics that complement notions in defining the digital twinning framework for bridges. The characteristics (elaborated in the dissertation) are:

- actual reflection: fidelity and synchronization,
- intelligence and autonomy,
- interaction,
- interoperability,
- modularity,
- expansibility and scalability,
- accessibility and security,
- uniqueness.

The characteristics are interconnected. Actuality, intelligence and autonomy, as well as interaction, are the foundation of the digital twinning concept. Interaction (the ability to cooperate with other objects in the digital dimension) is enabled by interoperability (the compatibility with general principles and standards of data exchange). Modularity enables practical implementations; not all the functionalities of digital twins must be implemented from the very beginning. The expansibility of functionalities and scalability are the results of previous characteristics. Accessibility for users and other instances, but with the security of data, as well as uniqueness (e.g., of identifiers), are the indispensable factors for effective implementations.

3.5 The need for practical use cases

Digital twins, as an emerging concept, still requires extensive research. Stating the object-suited principles is the first step. However, for development, digital twinning must provide practical use cases giving clear benefits to designers, engineers, and managers of facilities.

This dissertation proposes two solutions created by the author, utilizing techniques identified as components of civil engineering digital twinning. The solutions regard various lifecycle phases and provide benefits on the design and operational phases of the objects' lifecycle.

The first proposed solution is the optimization using visual programming and genetic algorithm [183]. Visual programming is used to link BIM modeling, FEM analysis, and optimization algorithm. The solution is a step towards utilizing digital twins' techniques in the design phase. The system is an effective geometrical optimization approach, assisting designers in their day-to-day tasks. Moreover, the integration of BIM and FEM models is crucial for creating a fully-functional digital twin of a bridge.

The second proposed technique is the generation of synthetic point clouds' datasets. Point clouds can be a base for modeling actual geometrical conditions in both operational and building phases. To automate the extraction of point clouds' information or even the generation of geometrical models, often machine learning algorithms are used. However, the available point clouds' datasets are not sufficient for the training of machine learning algorithms. The alternative is synthetic data. A scanning simulator created by the author allows for generating synthetic point clouds from BIM models. Acquiring the data is the first but indispensable step in developing machine learning solutions automating the utilization of point clouds, including systems to monitor geometrical changes in the digital twin models. The scheduled scanning of the physical object and then updating the model with trained algorithms will create a model-based history of geometrical changes. Monitoring, identifying, and comparing the geometrical changes can help identify structural malfunctions. As for bridges, monitoring the geometrical changes is crucial, especially for objects on the ground deformation areas (e.g., mining or post-seismic activities, tunneling). The model updating can therefore lead to expert systems alarming about hazards.

4 Optimization using visual programming and genetic algorithm

The multitude of geometric parameters makes bridges a challenging task for designers. Traditional iterative optimization is time-consuming and usually consists of an insufficient number of analyzed variants, leading to non-optimal solutions. Optimization techniques for bridges are the subject of extensive research. Many of the approaches use text-based programming languages. However, they are not easy to use by non-programmers. The alternative can be visual programming (VL), a more accessible form of developing algorithms. Visual programming is closely linked with BIM environments, where parametricism and quick modifications are essential. Visual programming scripts typically automate parametric geometry modeling but can also serve other engineering tasks. Merged with an optimization algorithm, they constitute an automated design process – generative design. In this approach, a user sets the constraints and ranges of selected parameters, and the algorithm optimally adjusts the values.

A generative design system based on visual programming can perform iterative, computationally-expensive tasks. However, as revealed within the literature study, its utility for bridges is still not sufficiently recognized, especially in the field of finite element analysis; not all civil-engineering visual programming environments provide proper FEM (finite element modeling) analysis tools. And the integration of BIM and FEM models is crucial for creating fully-functional digital twins of civil engineering facilities.

To address this gap, the author proposed [183] a generative design geometry optimization process (Fig. 2) for bridges using Dynamo, an open-source visual programming language popular among civil engineers. BIM and FEM models have been integrated with an optimization algorithm in an automated manner. To enable FEM analysis, Dynamo has been enhanced with a FEM package created by the author. It empowers construction analysis entirely in the visual programming environment, which additionally automates BIM modeling.

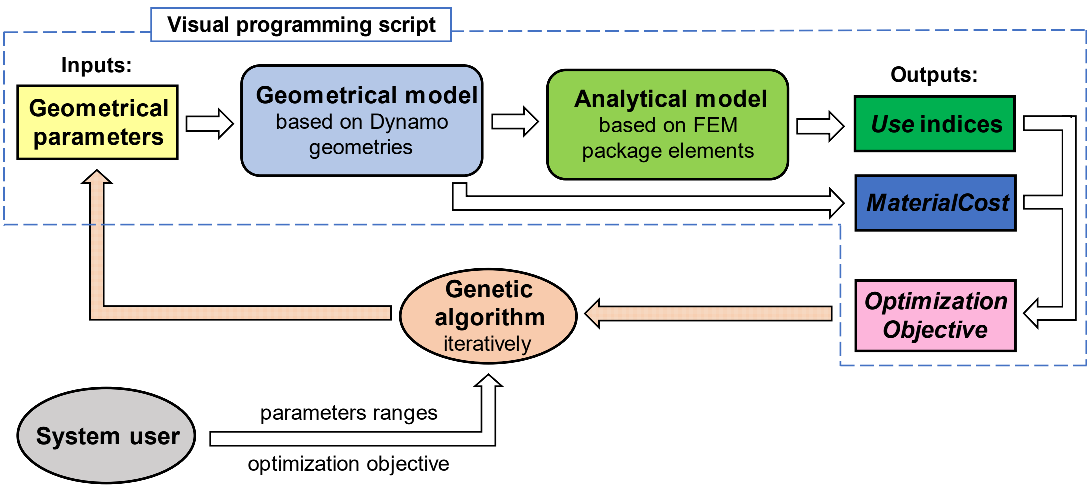


Fig. 2. Generative design optimization system operation flowchart

In a performed experiment, Dynamo scripts generated numerous models as variants of a reference arch bridge. Fig. 3 shows visualizations of basic FEM and BIM models of the bridge. The varying parameters include the hangers' system (vertical, radial, oblique, network), number of elements (e.g., hangers), arch rise, cross-sections' dimensions, and material parameters.

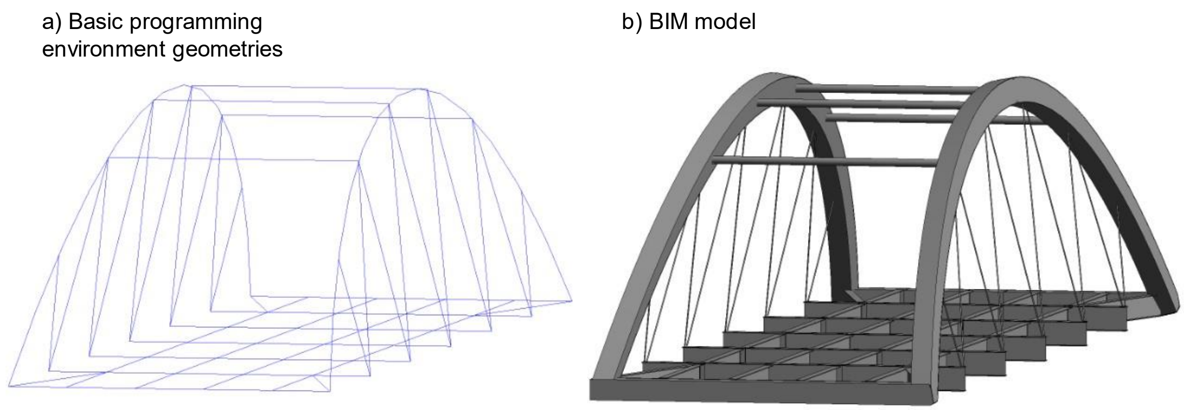


Fig. 3. Visualizations of a) basic FEM model geometry and b) generated BIM Model

The generated variants have been analyzed by FEM methods. The linked genetic algorithm took the analysis' results as inputs for the next generation of variants. The algorithm iteratively steered the geometric and material parameters aiming to minimize the objective function. The objective function was relative material cost depending on material volume and its cost factor. The results of the initial analysis are shown in Fig. 4. The initial analysis regarded variants of varying hangers' systems, hangers number, and arch rise. The task was to determine the range of these parameters to be regarded in further analyses. The ranges have been determined by searching for variants of permissible use (Use ≤ 1) and lower relative cost than in the reference structure (MaterialCost < 87,24).

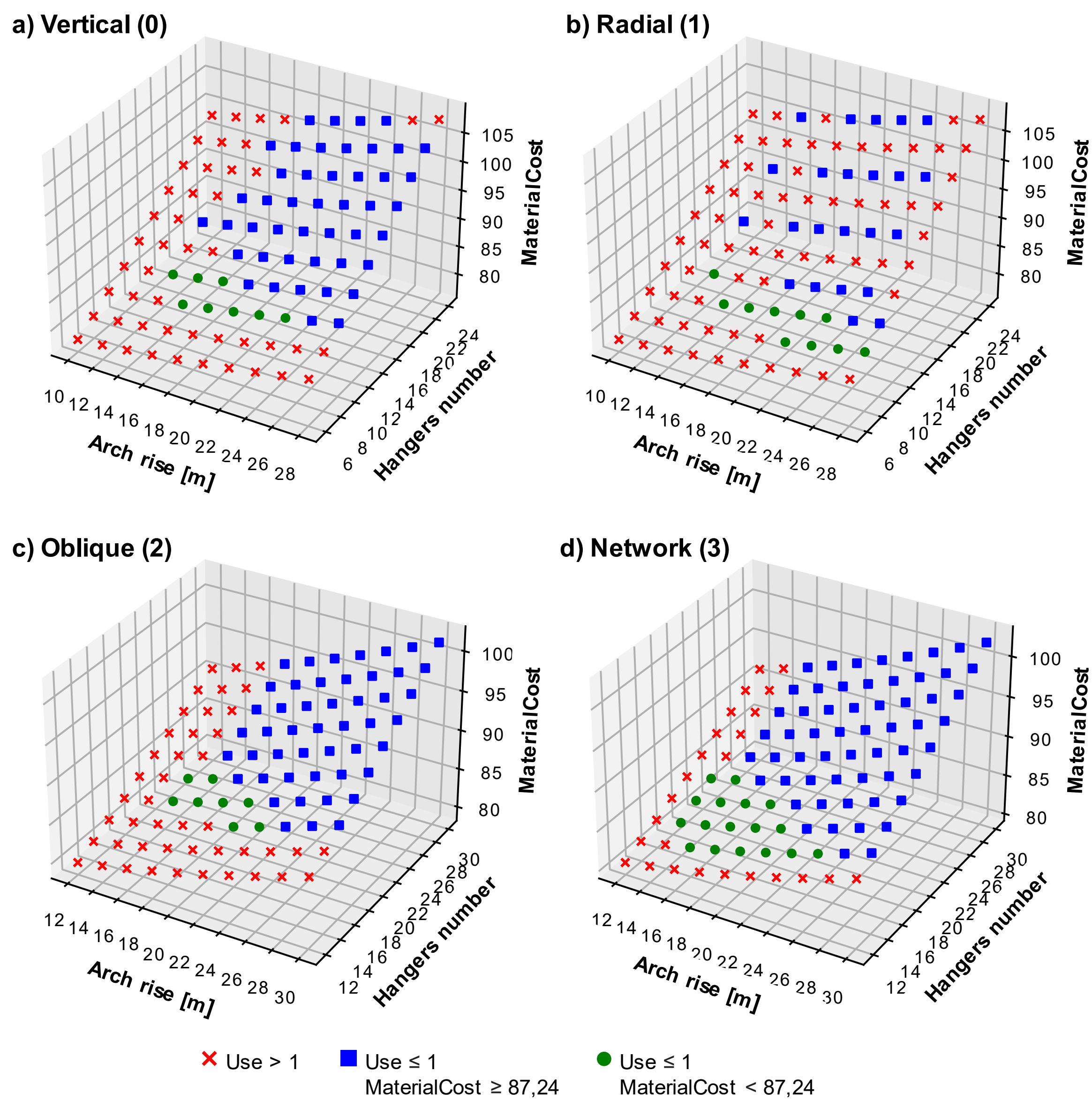


Fig. 4. The initial analysis of the variants of differing parameters' values (hangers' system type, hangers number, and arch rise) with reference to maximum structural use and relative cost

5 Generating synthetic data for point clouds' machine learning automation

Point clouds are becoming basic sources of geometrical data for physical infrastructure. However, data about the construction's shape is insufficient for digitizing facilities in the form of data-rich digital twins and performing advanced analysis – for that, semantic data is also required. Since manual extraction of semantic data from point clouds and its interpretation is time-consuming, error-prone, and inefficient [315–

319], much research effort is put into automation. The automation of point clouds is often based on machine learning (ML).

A properly operating ML system is based on two fundamentals: an algorithm suited for the task and a dataset for training. With the increasing computational power of algorithms, often the data is a bottleneck [244]. A proper dataset should be diverse, reliable, and contain a sufficient number of examples. Also, if the selected algorithm is a supervised one, the data examples must be labeled. If the examples' number is insufficient for the task complexity, ML solutions suffer from the data hunger effect [215] – insufficient training data inhibit achieving optimal algorithms' performance, hindering automation.

The acquisition of point clouds is time-consuming and expensive. Scanning some areas states practical challenges (e.g., closing a road to scan the lower part of a bridge). Moreover, manual labeling of points is tedious. Due to that, large datasets of point clouds are rare. The data hunger problem (the problem of insufficient data) is frequent in point cloud-dedicated ML solutions. One of the methods to overcome the data hunger problem is producing synthetic data. Synthetic data have some advantages compared to real data. Primarily, it is easier to acquire since there is no need for numerous scanning. Synthetic data is also scalable (the amount of generated data is limited only by computation power) and gives control over examples' features and distribution. Moreover, synthetic data ensures reliable annotations. However, to be used as an enhancement or replacement of real data in the process of training the machine learning algorithm, synthetic data properties should mimic the properties of real data.

Synthetic point clouds can be generated by geometrical models' sampling or by scanners' simulators. These techniques allow for preparing synthetic machine learning datasets acting as enhancements or replacements of real data. Several attempts have been made to create simulating tools [331,332,334–336,338,340,342,343]. However, there is still a lack of publicly available solutions that give sufficient control over simulation parameters and acquired point clouds' properties, emulate the operation of scanners, and do not require dedicated preparatory work (e.g., creating models only for the scanning simulation).

The author decided to address the need to generate synthetic point clouds. For this purpose, the author developed a simulator of laser scanning called DynamoPCSim. The simulator generates synthetic point clouds from geometrical and BIM models. Fig. 5 shows examples of the generated synthetic point clouds. The examples present a column of a bridge on which in-situ experiments have been performed. The simulator is based on ray tracing, a technique matching the operation of scanning lasers. Due to that, the produced point clouds' characteristics are closer to the real ones. The simulator is deployed as an add-in (a package) to Dynamo, an open-source visual programming environment. Since Dynamo enables linkage with Autodesk Revit modeling software, the simulator can utilize BIM models directly without requiring dedicated models prepared only for the purpose of scanning simulation. Due to the modular structure of the simulator and the flexibility of visual programming, it can be adjusted to different scanning properties and point clouds' parameters to be acquired.

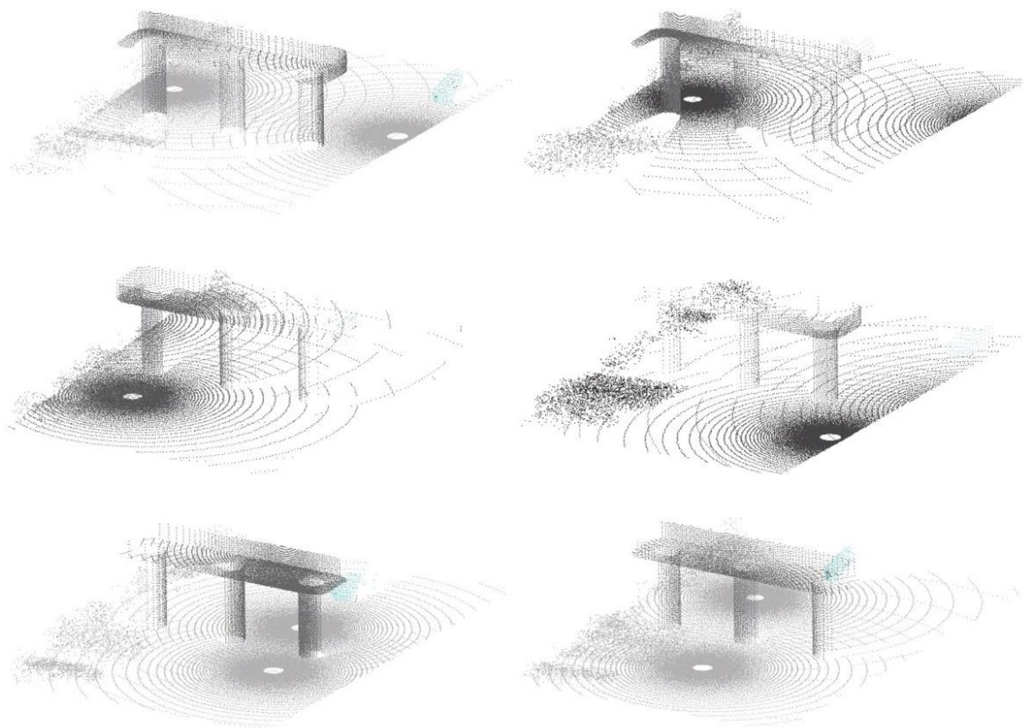


Fig. 5. Examples of synthetic point clouds generated with DynamoPCSim

A performed in the chapter experiment of semantic segmentation using a neural network confirmed the usability of generated synthetic data for machine learning. In the experiment, the PointNet neural network [222] was trained on synthetic point clouds. Then, it was validated on test sets. The first test set contained scenes of a synthetic point cloud, and the second – scenes of a real one. Fig. 6 shows confusion matrices of the predictions on both sets. The results are comparable in the recall metric. It indicates that generating synthetic point clouds can replace acquiring real point clouds in the training of a neural network.

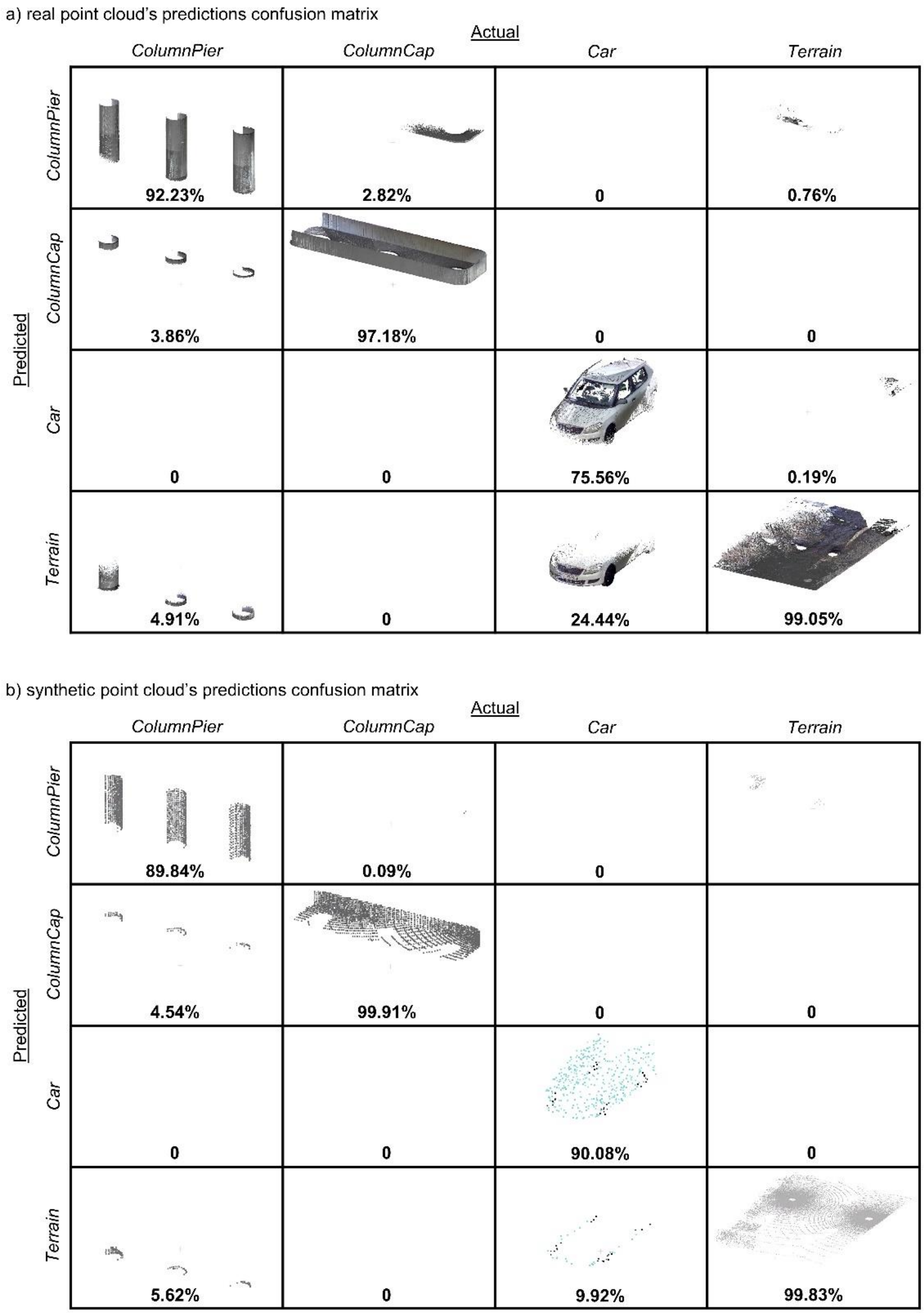


Fig. 6. Confusion matrices (with graphical outcomes) of neural network's operation on a) the real point cloud and b) a synthetic point cloud

The created by the author simulator addresses the first but indispensable step in developing machine learning solutions automating the utilization of point clouds – acquiring the data to train the algorithms. The trained systems can extract point clouds' information, segment them, and finally, also generate and update the point-cloud-based geometrical models of digital twins. The automatically updated models will match the real conditions of the physical objects, realizing the idea of a model coexisting with an object through the whole lifecycle, also in the operational phase. As for bridges, the information about geometrical changes will allow to detect and compare deformations, monitor the structure, and alarm about hazards.

6 Conclusions

The ongoing digital transformation is undoubtedly a challenge – but also an opportunity. Given the increasingly complex interconnections of scientific and industrial activities of various sectors, digital twinning improves the effectiveness of civil engineering and enables interdisciplinary cooperation. It also delivers new, practical tools, assisting designers, engineers, and managers in day-to-day tasks.

The proposed in this dissertation idea of a digital twin of a bridge is one of the first approaches to establish the foundations for these data-rich models of bridges. The dissertation identifies techniques for creating and utilizing digital twins of bridges and proposes their characteristics. The proposed concept addresses the general multi-industrial digital twinning fundamentals (identified with the literature study), the specificity of bridges, and the demands of designers, engineers, and managers of civil engineering facilities. With this approach, a digital twin of a bridge is ready for digital cooperation with other instances but also already delivers practical benefits. The benefits are, among others, the automation of the design, assessment of the structural condition, and effective data management. These benefits are realized by new digital twins' use cases. Two of these approaches have been proposed in the dissertation.

The first proposed technique is the automation of design using visual programming and genetic algorithm [183]. The solution presents how visual programming can integrate BIM and FEM Models. Such integration is crucial for creating fully-functional digital twins of bridges. Moreover, the implemented optimization algorithm allowed for analyzing much more variants than in the traditional design. It leads to optimal structural choices and enhances the effectiveness of the process.

The second proposed solution regards creating and updating geometries of digital twins also in the building and operational phases. It is a technique to generate synthetic point clouds as datasets to train machine learning algorithms. The trained systems can extract point clouds' information, segment them, and finally, also generate and update the point-cloud-based geometrical models of digital twins. The automatically updated models will match the real conditions of the physical objects, realizing the idea of a model coexisting with an object through the whole lifecycle, also in the operational phase. As for bridges, the information about geometrical changes will allow to detect and compare deformations, monitor the structure, and alarm about hazards.

The aspects regarded in the dissertation represent the initial stage of research on digital twins of bridges and should be further developed. The directions of further work concern both the theoretical foundations defining these virtual objects and practical ways of utilizing them. First of all, the technical implementation details of the proposed digital twins' characteristics should be determined. In the future, the concept will also need to comprise new, emerging technologies. As for the automation of the design using visual programming and optimization algorithms, additional aims should be considered. Today, mainly economic costs are taken into account. Nonetheless, environmental and social costs have already started to gain significance. And analyses of these parameters are enabled in BIM environments.

As for generating synthetic point clouds, the proposed technique is only the first step in the creation of digital twins in the operational phase. In further work, the synthetic data should be used to train machine learning systems that generate and update geometric models of digital twins. This will open up the possibility of creating expert systems for detecting geometric changes and alarming about potential dangers.

The conducted analysis of the state-of-the-art of digital twins of bridges, conceptual work, and in-situ experiments allowed for achieving the aims stated at the beginning of the dissertation. The foundations for the creation of digital twins of bridges have been declared by proposing a framework that takes into account the specificity of bridges and their lifecycle. It allows for the virtualization of not only physical objects but also their accompanying processes. The author is aware that the presented material does not fulfill all the assumptions of the digital twins' concept. However, the presented research and the resulting acquired competencies are the catalysts for further activities that will aim at increasingly complete implementations of digital twins of bridges in the future.

Streszczenie (PL)

Cyfrowe bliźniaki mostów: utworzenie podstaw wirtualizacji z praktycznymi sposobami użytkowania

Obserwowana współcześnie cyfryzacja niesie ze sobą zarówno nowe możliwości jak i poważne wyzwania. Zasady zrównoważonego rozwoju, ambitne cele proekologiczne i rosnąca populacja społeczeństw motywują wprowadzanie nowoczesnych i zintegrowanych technologii również w inżynierii lądowej. Potrzeba cyfrowej transformacji jest wyrażana przez dotykające nas problemy starzejącej się infrastruktury, margines produktywności sektora budowlanego oraz oczekiwania inżynierów i zarządców. Mosty jako złożone i wrażliwe na wpływy środowiska konstrukcje o logistycznym i strategicznym znaczeniu, potrzebują dziś nowych metod zarządzania. Cyfrowe bliźniaki, postrzegane jako kluczowy komponent strategii Industry 4.0 (Przemysł 4.0), stają się katalizatorem cyfrowej transformacji w ujęciu holistycznym.

Rozprawa proponuje pryncypia w zakresie tworzenia i wykorzystywania cyfrowych bliźniaków mostów. Uwzględnia generalne zasady tej idei, ale też specyfikę inżynierii lądowej i doświadczenia inżynierów. Proponowany cyfrowy bliźniak jest zatem wirtualnym odzwierciedleniem obiektu mostowego w całym jego cyklu życia. Jest scharakteryzowany przez swoją aktualność, swoistą inteligencję i autonomiczność, interaktywność i interoperacyjność, modułowość i rozszerzalność, skalowalność i dostępność oraz bezpieczeństwo i unikatowość. Z perspektywy technologicznej jest on ewolucją już dziś stosowanych technologii, rozwiązań i metod, które jeszcze nie zostały w wystarczający sposób zintegrowane. Proponowany cyfrowy bliźniak wykorzystuje bowiem elementy takie jak BIM (Building Information Modeling), SHM (Structural Health Monitoring) oraz AI (Artificial Inteligence). Przyjmuje standard IFC (Industry Foundation Classes) jako bazę dla modelu centralnego. Jest też wzbogacony przez dodatkowe techniki (np. programowanie graficzne, chmury punktów). Takie podejście umożliwia realizację procesu wdrażania w sposób ewolucyjny. Cyfrowe bliźniaki nie muszą przecież być perfekcyjne od samego początku. Koncepcja ta powinna dojrzewać w naturalny sposób, zachęcając użytkowników korzyściami, jakie wynikają z ich stosowania.

Aby ten ewolucyjny rozwój był możliwy, paradygmat cyfrowych bliźniaków musi dostarczać praktycznych sposobów użytkowania zapewniających realne korzyści inżynierom oraz właścicielom i zarządcom infrastruktury. Rozprawa omawia więc dwa utworzone przez autora praktyczne sposoby użytkowania wykorzystujące techniki wskazane jako podstawowe komponenty cyfrowych bliźniaków. Sposoby te dotyczą różnych faz cyklu życia.

Optymalizacja z wykorzystaniem programowania graficznego i algorytmu genetycznego automatyzuje głównie początkową fazę projektową. Programowanie graficzne zostało tu wzbogacone funkcjonalnościami MES tworząc jednolite środowisko optymalizacji geometrycznej oraz integracji z modelami BIM. Automatyzujące optymalizację algorytmy pozwalają na analizę zdecydowanie większej liczby wariantów

niż w tradycyjnym, iteracyjnym sposobie projektowania. Taka integracja modeli BIM i MES jest kluczowa do uzyskania pełnej funkcjonalności cyfrowego bliźniaka mostu.

Kolejne fazy cyklu życia obiektu mostowego dotyczą stosunkowo krótkiego procesu budowy oraz najdłuższej fazy operacyjnej. Tutaj autor zaproponował wykorzystanie chmur punktów, które można pozyskać technikami rekonstrukcji 3D, co w inżynierii lądowej najczęściej realizowane jest przez skanowanie laserowe lub fotogrametrię. Chmury punktów mogą być bazą dla modelowania aktualnej geometrii obiektu i to nie tylko podczas jego użytkowania, ale też w trakcie wznoszenia. W celu automatyzacji procesu pozyskiwania wartościowych informacji z chmur punktów, a nawet prób automatycznego generowania modeli, wykorzystywane są często algorytmy uczenia maszynowego. Niestety, dostępne zbiory chmur punktów nie pozwalają na efektywne trenowanie takich algorytmów. Alternatywą mogą być sztucznie wygenerowane dane w postaci syntetycznych chmur punktów. Uzyskanie danych do treningu algorytmów to pierwszy krok do tworzenia systemów monitorujących zmiany geometrii modeli cyfrowych bliźniaków z użyciem chmur punktów. Powtarzane okresowo skanowanie fizycznego obiektu, a następnie użycie zaproponowanych algorytmów analizy nowej chmury punktów pozwoli na aktualizację modelu geometrii. Zaktualizowane i porównywane ze sobą kolejne wydania modelu będą mogły być wykorzystane do identyfikacji zmian w geometrii, które z kolei mogą wskazywać na ewentualne nieprawidłowości pracy fizycznego obiektu. W przypadku mostów zlokalizowanych na obszarach z deformacją terenu (np. aktywność górnicza, sejsmiczna lub tunelowanie), taka automatyczna aktualizacja modelu może być wykorzystana do monitorowania stanu deformacji konstrukcji i wynikających z tego zagrożeń bezpieczeństwa.

Przedstawiona w rozprawie koncepcja cyfrowego bliźniaka mostu jest jedną z pierwszych prób zdefiniowania tych wirtualnych modeli dla konstrukcji mostowych. Studium literatury pokazało, że w światowej inżynierii mostowej prace nad takim podejściem są dopiero na początkowych etapach. Nabierająca rozpędu cyfrowa transformacja niemal wszystkich dziedzin naszego życia jest bezsprzecznie dużym wyzwaniem – ale i szansą. W kontekście coraz bardziej przenikających się różnych dyscyplin naukowych i sektorów gospodarki, paradygmat cyfrowych bliźniaków zwiększa efektywność w obszarze inżynierii lądowej i umożliwia międzybranżową współpracę na cyfrowych platformach wymiany danych. Dostarcza też nowych, praktycznych narzędzi, które wspomagają realizację codziennych zadań wykonywanych przez inżynierów i zarządców infrastruktury. Przedstawione w tej rozprawie pryncypia i sposoby użytkowania cyfrowych bliźniaków mostów są jednymi z pierwszych kroków w kierunku infrastruktury przyszłości. Dzięki temu będzie możliwa wirtualizacja już nie tylko samych fizycznych obiektów mostowych, ale również towarzyszących im procesów.

Poszerzone streszczenie (PL)

Cyfrowe bliźniaki mostów: utworzenie podstaw wirtualizacji z praktycznymi sposobami użytkowania

1 Motywacje

1.2 Adaptuj się (lub kreuj) by efektywnie funkcjonować

Świat nieustannie się zmienia. To stwierdzenie – jakkolwiek trywialne – stało się też powodem do nowej aktywności inżynierów, która dotyczy pojęcia cyfrowych bliźniaków (ang. digital twins). Ale to stwierdzenie może być jeszcze mocniej zaakcentowane. Otóż, świat nieustannie się zmienia w ciągle rosnącym tempie. Walidacja tej tezy jest szczególnie widoczna w ostatniej dekadzie, która przez wielu nazywana jest czwartą rewolucją przemysłową. Wpisuje się też w trendy obserwowanej i osobiście przez nas doświadczanej transformacji cyfrowej z postępującą wirtualizacją kolejnych obszarów naszego życia. Cyfrowa transformacja stanowi główny element strategii rozwoju opisywanej w świecie jako Industry 4.0 (Przemysł 4.0).

Wszelkie zmiany dzielą ludzi na entuzjastów oraz przeciwników tych procesów. I to nie tylko w kontekście zmieniających się standardów moralnych i norm obyczajowych, ale też technologicznych innowacji, które wpływają na sposób naszego życia. Podczas rewolucji przemysłowych, ludzie postrzegali nadchodzącą automatyzację bardzo odmiennie. Niektórzy byli podekscytowani poprawą jakości życia, inni bali się utraty pracy i potrzeby odnalezienia się w nowych realiach. Dzisiaj jednak już wiemy, że grupy, które wykorzystały możliwości płynące z tamtych innowacji, nadały kształt naszej dzisiejszej rzeczywistości.

Trwająca obecnie transformacja cyfrowa niewątpliwie niesie ze sobą wiele zmian. Rzeczywistość jest coraz mocniej wirtualizowana. Ludzie są nieustannie podłączeni do sieci, produkując przy tym coraz większe ilości wnikliwie analizowanych danych. Sztuczna inteligencja przejmuje zadania zarezerwowane wcześniej tylko dla ludzi. Te zmiany budzą podekscytowanie i lęk. Oba te odczucia są oczywiście zasadne. Niemniej jednak, zgodnie z maksymą “musisz się rozwijać, żeby nie stać w miejscu”, pojedynczy ludzie i społeczeństwa, biznes i całe sektory przemysłu, wspólnoty narodowe i globalne korporacje, niezależnie od swojego indywidualnego nastawienia, muszą być przygotowani na zmiany wynikające z powszechnej cyfryzacji życia.

Zestawiając tradycyjne słowa, jak *most* czy *filar* z takimi hasłami, jak *sztuczna inteligencja* lub *wirtualna rzeczywistość*, mosty wydają się być technologicznym reliktem zamierzchłych czasów. Niemal zabytkiem. Myśląc o mostach, wyobrażamy sobie przecież raczej monumentalne i statyczne konstrukcje, które powinny posiadać techniczną sprawność przez wiele dziesiątków lat. I to bez kosztownych ingerencji. Ale czy na pewno tak jest? Nie do końca. Niestety, mosty są cały czas narażone na destrukcyjne oddziaływanie środowiska i pojazdów. Ta stopniowa i powolna degradacja stanu technicznego jest często nawet niewidoczna dla użytkowników. Natomiast dostrzec ją mogą zarządcy i terenowi inspektorzy, którzy wykonują na nich cykliczne oceny stanu technicznego. Dlatego konstrukcje mostowe powinny być przez nich odpowiednio utrzymywane i monitorowane. Co więcej, mosty jako najbardziej wymagające i odpowiedzialne konstrukcje budowlane, od zawsze stanowią obszary poszukiwania innowacji w inżynierii lądowej. To tu najczęściej stosowane są nowe materiały budowlane, rozwiązania konstrukcyjne, uprzemysłowione metody budowy i złożone elementy wyposażenia. Zbierane w inżynierii mostowej doświadczenia przenoszone są później do innych sektorów budownictwa. Wymagania współczesnego świata pchają inżynierów mostowych w kierunku poszukiwania innowacji. Wynika to choćby z konieczności adaptacji do zasad zrównoważonego rozwoju i ochrony środowiska oraz spełnienia potrzeb rosnącej i starzejącej się populacji.

Aby efektywnie funkcjonować, konieczna jest adaptacja do zmian, jakie serwuje nam otoczenie. Również w kontekście technologicznym. Ale adaptowanie nie oznacza pasywnego i bezkrytycznego naśladowania. Może to oznaczać również kreowanie. W cyfrowym świecie, sektory gospodarki są coraz mocniej połączone i interdyscyplinarne. Ta kooperacja wymaga nowych zasad i paradygmatów. Cyfrowe bliźniaki, które postrzegane są jako kluczowy składnik strategii Przemysł 4.0, stają się właśnie takim interdyscyplinarnym paradygmatem. Cyfrowe bliźniaki w wielu dziedzinach są już wykorzystywane, ale ich definicje i praktyki są wciąż formowane. Powinni w tym brać udział nie tylko programiści i automatycy, ale w przypadku mostów również środowisko inżynierów lądowych. W ten sposób będą mieli swój udział w formowaniu zasad globalnej cyfryzacji dla branży budowlanej, a cyfrowe bliźniaki mostów będą lepiej odpowiadać na potrzeby zarządców i właścicieli infrastruktury.

1.2 Łączniki ludzi i danych

Oddziaływanie mostów widoczne jest zarówno w codziennym życiu jednostek, jak i w skali całych państw i gospodarek, co można przecież wyrazić liczbami. Tylko w Stanach Zjednoczonych użytkowanych jest ponad 600 tys. mostów [2], z których aż 220 tys. wymaga znacznych prac remontowych lub w skrajnych przypadkach nawet wymiany. Ten problem staje się jeszcze bardziej złożony, jeśli weźmie się pod uwagę czynnik ekonomiczny. Szacowany koszt napraw lub wymiany wspomnianych mostów to 260 miliardów dolarów. Jest to wartość porównywalna z PKB Czech (282 mld USD), Nowej Zelandii (250 mld USD) czy Portugalii (250 mld USD). Dla porównania, PKB Polski to 674 mld USD (dane Banku Światowego na rok 2022).

Ogromna liczba i złożoność mostów wymusiły potrzebę poprawy efektywności gospodarowania tymi wciąż rosnącymi, a jednocześnie starzejącymi się zasobami. To poskutkowało opracowaniem dedykowanych systemów zarządzania mostami (ang. bridge management system) i standaryzacją w zakresie realizowanych procedur utrzymaniowych, ale również procesów przetwarzania gromadzonych przy tej okazji informacji. A te dziś muszą już być dopasowane do aktualnych możliwości technicznych, jakie niesie ze sobą transformacja cyfrowa. Niestety, zarządcy infrastruktury wciąż jeszcze wykorzystują starsze tradycyjne metody zarządzania swoimi zasobami. Jest to szczególnie widoczne w odniesieniu do nieefektywnego przetwarzania danych. Okresowe oceny stanu technicznego mostów dalej wykonuje się tradycyjnymi metodami wizualnymi, w których terenowy inspektor mostowy ręcznie wypełnia papierowy formularz. Proces jest nieefektywny i obarczony subiektywizmem. Istnieje ryzyko przeoczenia lub pominięcia defektów, a to w przypadku starzejącej się infrastruktury, może skutkować stratami ekonomicznymi, a nawet awariami.

Cyfrowa transformacja sprawia, że obiekty mostowe powoli ewoluują ze statycznych konstrukcji w systemy zapewniające wielowymiarowe relacje. Proces ten można właściwie nazwać cybernetyzacją, w którym te dotychczas techniczne obiekty będą stawać się obiektami coraz bardziej cyber-fizycznymi. Będą to inteligentne obiekty działające zgodnie z założeniami Internetu rzeczy (Internet of Things, IoT). Wynikać to będzie z potrzeb ich właścicieli i zarządców, ale też bezpieczeństwa i komfortu fizycznych użytkowników. A to będzie wymuszało łączność i integrację już w wymiarze wirtualnym. Głównie w zakresie gromadzenia, przetwarzania i udostępniania danych. Mosty będą współpracować z innymi obiektami tak, by umożliwić holistyczne zarządzanie i formować w przyszłości cyfrowe sieci inteligentnych miast (ang. smart city). Aby było to możliwe, potrzebny jest globalny paradygmat cyfrowej współpracy. Takie możliwości kryje w sobie koncept cyfrowych bliźniaków.

1.3 Cele i założenia rozprawy

Zasadniczym celem rozprawy było zdefiniowanie podstaw tworzenia cyfrowych bliźniaków mostów i wskazanie sposobów ich wykorzystywania. Wpisuje się to w ogólne trendy wirtualizacji fizycznych obiektów i procesów w ramach trwającej cyfrowej transformacji. Cyfrowe bliźniaki rozumiane są tutaj jako wirtualne odwzorowania realnych obiektów w ich całym cyklu życia. Zaproponowany w pracy szablon tych wirtualnych modeli uwzględnia ogólne zasady idei cyfrowych bliźniaków, które są już od pewnego czasu stosowane w inżynierii. Zostały one jednak rozwinięte o specyfikę obiektów mostowych i warunki funkcjonowania w obszarze inżynierii lądowej. Praktyczność zaproponowanych rozwiązań została uzasadniona konkretnymi przypadkami użycia, które wykorzystują techniki zidentyfikowane jako komponenty proponowanego szablonu cyfrowych bliźniaków. Osiągnięcie tego zasadniczego celu rozprawy było możliwe przez realizację kilku poniższych celów pośrednich.

- Ocena aktualnego stanu rozwoju cyfrowych bliźniaków w innych dziedzinach poprzez studium dostępnej literatury oraz sprawdzenie przypadków implementacji w obszarze mostownictwa.
- Identyfikacja najważniejszych technik, które mogą być wykorzystane do tworzenia oraz użytkowania cyfrowych bliźniaków mostów.
- Opracowanie poszerzonej definicji cyfrowych bliźniaków mostów z ich zasadniczymi cechami i z uwzględnieniem faz cyklu życia.
- Przeprowadzenie praktycznej integracji modeli BIM i MES w środowisku programowania graficznego na potrzeby tworzenia cyfrowego bliźniaka w fazie projektowej, ale przy ograniczonym zakresie i z wykorzystaniem algorytmu optymalizacyjnego do automatyzacji procesu projektowania.
- Propozycja metody generowania syntetycznych chmur punktów w celu tworzenia i aktualizacji modeli geometrycznych cyfrowych bliźniaków w fazie operacyjnej z użyciem algorytmów uczenia maszynowego.

Zestawione powyżej cele pośrednie z pewnością nie wypełniają wszystkich definiowanych dziś w literaturze założeń, jakimi powinny cechować się ogólnie rozumiane cyfrowe bliźniaki. Jednak pracując nad opisem cyfrowych bliźniaków mostów wzięto pod uwagę początkowy etap rozwoju tego konceptu, który w przyszłości integrować przecież będzie wiedzę i rozwiązania stosowane w wielu dyscyplinach. Takie podejście umożliwia więc realizację procesu wdrażania w sposób ewolucyjny. Cyfrowe bliźniaki nie muszą przecież być kompletne i perfekcyjne od samego początku. Koncepcja ta powinna dojrzewać w naturalny sposób, zachęcając użytkowników korzyściami, jakie wynikają z ich stosowania. Nie bez znaczenia są także ograniczenia niniejszej pracy wynikające z czasu jej wykonywania i samej objętości. Zaproponowany opis cech cyfrowych bliźniaków wskazuje więc ogólne pryncypia, a nie szczegóły technicznych implementacji. Natomiast zawarte w pracy przykłady tworzenia i użytkowania nawet niekompletnego cyfrowego bliźniaka mają jedynie pokazać potencjalne możliwości takiego podejścia. Wykorzystane w przykładach technologie i rozwiązania dotyczą określonych potrzeb fazy projektowej (automatyzacja i optymalizacja procesu projektowego) oraz najdłuższej fazy operacyjnej (zmiany geometrii obiektu w trakcie długotrwałego użytkowania). Przykład z generowaniem syntetycznych chmur punktów nie omawia pełnego procesu tworzenia i aktualizacji geometrii modeli cyfrowych bliźniaków. Na tym początkowym etapie skupia się on na pierwszym, nieodzownym kroku do implementacji takich rozwiązań, jakim jest pozyskiwanie danych do trenowania algorytmów uczenia maszynowego. W dalszych pracach wytrenowane algorytmy będą mogły być wykorzystane przy implementacji cyfrowych bliźniaków mostów w powiązaniu z ich rzeczywistymi odpowiednikami. A w szczególności obiektami mostowymi, które podlegają wpływom deformacji terenu i narażone są na niebezpieczne zmiany geometrii.

2 Cyfrowy bliźniak jako kolejny etap rozwoju procesów symulacji

Cyfrowy bliźniak to wirtualne odwzorowanie obiektu w całym cyklu jego życia. Podejście takie początkowo stosowane było przez NASA [3] na potrzeby księżycowego programu Apollo, a dziś wykorzystywane jest w wielu obszarach, stając się elementem cyfrowej transformacji.

Cyfrowy bliźniak, jako model współistniejący z rzeczywistym obiektem, powinien zostać utworzony już podczas pierwszej fazy projektowania fizycznego obiektu. W tych początkowych etapach, które wynikają głównie z faktu użycia do zarządzania informacją o obiekcie budowlanym zasad metodyki BIM, cyfrowy bliźniak wirtualnie odzwierciedla formującą się koncepcję fizycznego obiektu. Wraz z rozwojem projektu, staje się on modelem informacyjnym z centralną bazą wiedzy o stosunkowo wysokim poziomie szczegółowości. Następnie odgrywa ważną rolę w planowaniu i optymalizacji fazy wykonawczej, podczas której uwzględniane są realizowane w terenie procesy budowlane. Wykorzystywanie modeli do symulacji i wizualizacji podczas projektowania i wykonawstwa jest coraz częstszą praktyką. Jednak paradygmat cyfrowych bliźniaków zakłada korzystanie z wirtualnych modeli w całym cyklu życia fizycznego obiektu. Dzięki synchronizacji z fizycznym odpowiednikiem także w fazie użytkowania, cyfrowy bliźniak przekracza możliwości izolowanych dotąd modeli, jakie tworzył projektant. Obiekt i jego wirtualny odpowiednik zaczynają współistnieć wymieniając się między sobą informacjami. Dane z fizycznego obiektu zasilają wirtualny model, a ten może wpływać na działanie fizycznego obiektu.

Cyfrowy bliźniak gromadzi dane, służące odwzorowaniu procesów zachodzących w obiekcie fizycznym. Rejestrowany jest sposób w jaki obiekt reaguje na zmieniające się warunki oraz jak te reakcje zmieniają się w czasie. Analizy takiego złożonego i obszernego zbioru danych pozwalają lepiej zrozumieć działanie samego obiektu. Dane pozwalają również trenować inteligentne algorytmy do rozpoznawania wzorców (ang. pattern recognition) i anomalii (ang. anomaly detection). Możliwa jest sukcesywna transformacja cyfrowego bliźniaka w system o coraz większej autonomii w zakresie podejmowania decyzji i przewidywania ich następstw. Cyfrowy model odzwierciedlający rzeczywisty aktualny stan obiektu pozwala na prowadzenie kompleksowych i wiarygodnych symulacji. Jednolite systemy danych umożliwiają holistyczne zarządzanie uwzględniające współoddziaływanie obiektów. Dzięki temu możliwa jest optymalizacja kosztów ekonomicznych, ekologicznych czy społecznych i to bez utraty poziomu bezpieczeństwa tych obiektów.

Zainteresowanie cyfrowymi bliźniakami wzrasta zarówno w środowiskach akademickich, co widoczne jest w dynamicznym wzroście publikacji na ten temat, ale też w zastosowaniach przemysłowych. Wciąż jednak są to głównie aktywności badawcze. Z jednej strony tworzone są teoretyczne podstawy, szablony i definicje cyfrowych bliźniaków, a z drugiej wprowadzane są już praktyczne sposoby ich użytkowania. Oba te aspekty zostały opisane w niniejszej rozprawie.

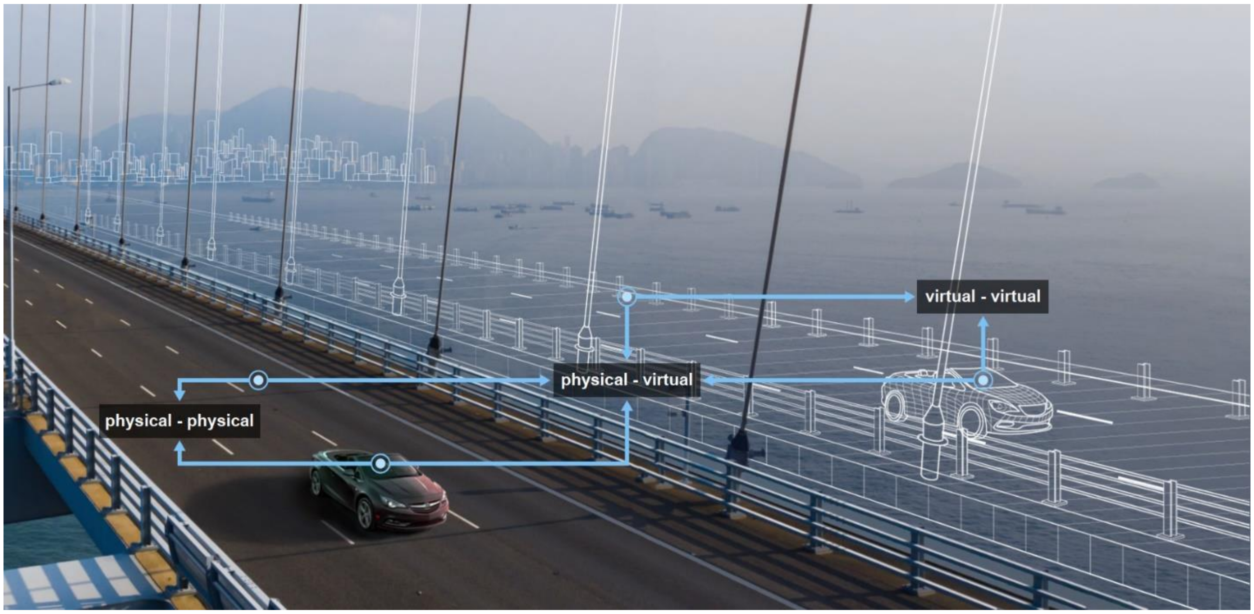
3 Idea cyfrowego bliźniaka mostu

3.1 Ogólne założenia i potrzeby

Tworzenie podstaw teoretycznych cyfrowych bliźniaków mostów musi być zapoczątkowane pewną, być może nawet, wyidealizowaną wizją. Ale równie ważne są czynniki pozwalające wdrożyć tę wizję w praktyce. Nie mają bowiem sensu działania zmierzające do formowania samych wyidealizowanych koncepcji, które nie nadają się do praktycznego wdrożenia. Aby cyfrowe bliźniaki mogły się dalej rozwijać, takie podejście musi być zaakceptowane przez stosujących je praktyków branży budowlanej. Do tej grupy użytkowników zaliczyć można na pewno inżynierów, którzy na kolejnych etapach cyklu życia obiektu budowlanego pełnią różnorodne funkcje, czyli np. przedstawiciele projektantów, wykonawców, zarządców. Z pewnością nie mogą to być jedynie środowiska akademickie czy osoby z branży IT. Nawet jeśli to oni właśnie odpowiadają za zasady tworzenia wirtualnego modelu. Szablon cyfrowych bliźniaków powinien więc służyć osiągnięciu praktycznych korzyści dla wszystkich interesariuszy, a nie utrudniać i ograniczać ich działania poprzez konieczność stosowania wyidealizowanych zasad i modeli.

Pierwsze praktyczne próby wdrażania cyfrowych bliźniaków wykorzystują współczesne techniki i podejścia stosowane już inżynierii lądowej (np. MES, SHM, BIM, rekonstrukcja 3D itd.). Do ich integracji i wzbogacenia coraz częściej wykorzystywane są algorytmy sztucznej inteligencji. Próby te są jednak wciąż częściowe. Nie wprowadzają kompleksowych cyfrowych bliźniaków, a jedynie uwzględniają poszczególne funkcjonalności. Mogą to być na przykład wyizolowane modele geometrii 3D (i to nawet z elementami semantyki w opisie komponentów) lub gromadzenie danych z systemów monitoringu typu SHM. Niemniej jednak, te działania promując automatyzację i cyfryzację w inżynierii lądowej, stawiają fundamenty pod przyszłość cyfrowych bliźniaków. Również obiektów mostowych.

Rozprawa proponuje pryncypia w zakresie tworzenia i wykorzystywania cyfrowych bliźniaków mostów (Rys. 1). Uwzględnia generalne zasady tej idei, ale też specyfikę inżynierii lądowej i doświadczenia inżynierów. Proponowany cyfrowy bliźniak jest zatem wirtualnym odzwierciedleniem obiektu mostowego w całym jego cyklu życia. Jest scharakteryzowany przez swoją aktualność, swoistą inteligencję i autonomiczność, interaktywność i interoperacyjność, modułowość i rozszerzalność, skalowalność i dostępność oraz bezpieczeństwo i unikatowość. Z perspektywy technologicznej jest on ewolucją już dziś stosowanych technologii, rozwiązań i metod, które jeszcze nie zostały w wystarczający sposób zintegrowane. Proponowany cyfrowy bliźniak wykorzystuje bowiem takie elementy, jak BIM (Building Information Modeling), SHM (Structural Health Monitoring) oraz AI (Artificial Inteligence). Przyjmuje standard IFC (Industry Foundation Classes) jako bazę dla modelu centralnego. Jest też wzbogacony przez dodatkowe techniki (np. programowanie graficzne, chmury punktów). Takie podejście umożliwia realizację procesu wdrażania w sposób ewolucyjny. Cyfrowe bliźniaki nie muszą przecież być perfekcyjne od samego początku. Koncepcja ta powinna dojrzewać w naturalny sposób, zachęcając użytkowników korzyściami, jakie wynikają z ich stosowania.



Rys. 1. Wizja cyfrowego bliźniaka mostu

3.2 Techniki stosowane w cyfrowych bliźniakach w inżynierii lądowej

Ten rozdział zestawia techniki i podejścia, które zidentyfikowane zostały jako kluczowe przy tworzeniu i wykorzystywaniu cyfrowych bliźniaków obiektów w inżynierii lądowej. Techniki te to:

- Building Information Modeling (BIM), czyli modelowanie informacji o obiekcie budowlanym;
- Industry Foundation Classes (IFC), czyli otwarty standard opisu obiektu budowlanego;
- Structural Health Monitoring (SHM), czyli monitorowanie stanu technicznego konstrukcji;

- Artificial Intelligence (AI), czyli sztuczna inteligencja;
- Visual Programming (VP), czyli programowanie graficzne;
- Point clouds, czyli chmury punktów.

Autorzy opisujący implementacje cyfrowych bliźniaków często nadużywają tego pojęcia. Jest to zrozumiałe w sytuacji, gdy nie zostało ono jeszcze w pełni zdefiniowane. Autorzy często używają tylko jednej z wymienionych technik lub metod, nazywając utworzony system kompletnym cyfrowym bliźniakiem. Wydaje się to jednak nie w pełni poprawne. Techniki te mogą być komponentami paradygmatu cyfrowych bliźniaków, ale jedynie ich świadome i interoperacyjne połączenie umożliwia kompleksowe wdrożenie takiej wizji. Model BIM lub system monitoringu SHM to jeszcze nie jest cyfrowy bliźniak. W tym rozdziale omówione zostały różnice pomiędzy tymi technikami i sposoby ich użycia przy tworzeniu i wykorzystywaniu cyfrowych bliźniaków mostów.

3.3 & 3.4 Studium cyfrowych bliźniaków mostów

Studium cyfrowych bliźniaków mostów odnosi się do technik i metod, jakie już dziś są stosowane w projektowaniu, budowie i utrzymaniu obiektów mostowych. Ich integracja oraz coraz większy stopień ich cyfryzacji i automatyzacji prowadzą do utworzenia nowej i rozszerzonej koncepcji cyfrowego bliźniaka mostu. Towarzyszą też temu pewne przemyślenia, które obejmują poniższe stwierdzania lub założenia.

- Wdrożenie cyfrowych bliźniaków jest owocne zarówno z perspektywy technicznej, jak i biznesowej.
- Szablony cyfrowych bliźniaków powinny być dedykowane konkretnym typom obiektów, również obiektów mostowych.
- Nie zawsze jest potrzebna bardzo wysoka wierność odwzorowania geometrii czy wszystkich właściwości konstrukcji.
- Synchronizacja danych pomiędzy obiektem fizycznym i wirtualnym nie zawsze musi odbywać się w czasie rzeczywistym.
- System cyfrowego bliźniaka powinien zostać zaprojektowany tak, by uwzględniać różnego rodzaju dane.
- Cyfrowy bliźniak wyposażony jest w dedykowane interfejsy użytkowników, które zależne są od etapu życia, w jakim znajduje się fizyczny obiekt i fazy rozwoju jego wirtualnego odpowiednika.
- Cyfrowy bliźniak zawiera dane pozwalające na analizę i symulowanie procesów, jakie zachodzą w rzeczywistym obiekcie.
- Cyfrowy bliźniak jest inteligentny i wraz z rozwojem zyskuje coraz większą autonomię.
- Cyfrowy bliźniak jest systemem systemów (systemem modeli i modułów), a nie pojedynczym modelem.
- Cyfrowe bliźniaki otwierają nowe możliwości empirycznego poznawania natury fizycznych obiektów.

- Cyfrowe bliźniaki powinny dostarczać właścicielom i użytkownikom realnych i wymiernych korzyści.

Studium omawia charakterystyczne cechy cyfrowych bliźniaków mostów, na które składają się między innymi:

- aktualność odwzorowania obiektu z odpowiednim poziomem szczegółowość i synchronizacji,
- inteligencję i autonomiczność podejmowania decyzji,
- interaktywność,
- interoperacyjność,
- modułowość,
- rozszerzalność i skalowalność,
- dostępność i bezpieczeństwo,
- unikatowość.

Opisane charakterystyki są ze sobą powiązane. Aktualność odwzorowania, inteligencja i autonomiczność oraz interaktywność to fundamenty idei cyfrowych bliźniaków. Interaktywność, czyli współpraca z innymi obiektami w cyfrowym wymiarze, jest możliwa przez interoperacyjność, czyli kompatybilność z ogólnymi zasadami i standardami wymiany danych. Modułowość umożliwia praktyczną implementację, gdyż nie wszystkie funkcjonalności cyfrowego bliźniaka muszą być implementowane od samego początku. Rozszerzalność i skalowalność to efekty implementacji poprzednich charakterystyk. Na przykład, modularność pozwala na implementację nowych modeli i funkcjonalności, które potem mogą podlegać skalowaniu w zakresie rozmiarów modelowanych obiektów i ilości przetwarzanych danych. Dwie kolejne powiązane cechy to dostępność dla użytkowników lub cyberfizycznych obiektów, ale przy jednoczesnym zapewnieniu bezpieczeństwa danych. Natomiast unikatowość sprawia, że utworzony cyfrowy bliźniak staje się osobnym cyberfizycznym bytem z unikalnym systemem identyfikacji i dostępu.

3.5 Potrzeba praktycznych sposobów tworzenia i użytkowania cyfrowych bliźniaków

Cyfrowe bliźniaki mostów to wciąż jeszcze słabo rozwinięta technologia, która wymaga szeroko zakrojonych badań. Pierwszym krokiem jest konceptualizacja zasad ich tworzenia i użytkowania. Jednak aby rozwój idei cyfrowych bliźniaków był możliwy, musi ona dostarczać nie tylko teoretycznych zasad, ale też praktycznych sposobów użytkowania, które zapewnią realne korzyści inżynierom.

Rozprawa omawia dwa zaproponowane przez autora praktyczne sposoby takiego użytkowania wykorzystujące techniki wskazane jako podstawowe komponenty cyfrowych bliźniaków. Sposoby te dotyczą różnych faz cyklu życia. Optymalizacja z wykorzystaniem programowania graficznego i algorytmu genetycznego automatyzuje głównie początkową fazę projektową [183]. Programowanie graficzne zostało tu wzbogacone funkcjonalnościami MES, tworząc jednolite środowisko optymalizacji geometrycznej oraz integracji z modelami BIM. Automatyzujące optymalizację algorytmy pozwalają na analizę zdecydowanie większej liczby wariantów niż w tradycyjnym sposobie projektowania. Z kolei integracja modeli BIM i MES jest kluczowa dla w pełni funkcjonalnego cyfrowego bliźniaka mostu.

Kolejny zaproponowany przez autora sposób użytkowania, to wykorzystanie chmur punktów do modelowania aktualnej geometrii obiektu, co może być wykorzystane przy ocenie zmian geometrycznych wpływających na bezpieczeństwo konstrukcji. Na przykład przemieszczenia łożysk i urządzeń dylatacyjnych, wychylenia filarów, deformacji przęseł itp. Chmury punktów mogą być bazą do modelowania aktualnej geometrii obiektu i to nie tylko podczas jego użytkowania, ale też w trakcie wznoszenia. W celu automatyzacji procesu pozyskiwania wartościowych informacji z chmur punktów, a nawet prób automatycznego generowania modeli, wykorzystywane są często algorytmy uczenia maszynowego. Niestety, dostępne zbiory chmur punktów nie pozwalają na efektywne trenowanie takich algorytmów. Alternatywą mogą być sztucznie wygenerowane dane w postaci syntetycznych chmur punktów. Taka metoda została zaproponowana w pracy i zwalidowana na przykładzie filara rzeczywistego obiektu mostowego.

Uzyskanie danych do treningu algorytmów to pierwszy krok do tworzenia systemów monitorujących zmiany geometrii modeli cyfrowych bliźniaków z użyciem chmur punktów. Powtarzane okresowo skanowanie fizycznego obiektu, a następnie użycie zaproponowanych algorytmów analizy nowej chmury punktów pozwoli na aktualizację modelu geometrii. Zaktualizowane i porównywane ze sobą kolejne wydania modelu będą mogły być wykorzystane do identyfikacji zmian w geometrii, które z kolei mogą wskazywać na ewentualne nieprawidłowości pracy fizycznego obiektu. W przypadku mostów zlokalizowanych na obszarach z deformacją terenu (np. aktywność górnicza, sejsmiczna lub tunelowanie), taka automatyczna aktualizacja modelu może być wykorzystana do monitorowania stanu deformacji konstrukcji i wynikających z tego zagrożeń bezpieczeństwa. Ale ten etap będzie dopiero przedmiotem dalszych prac.

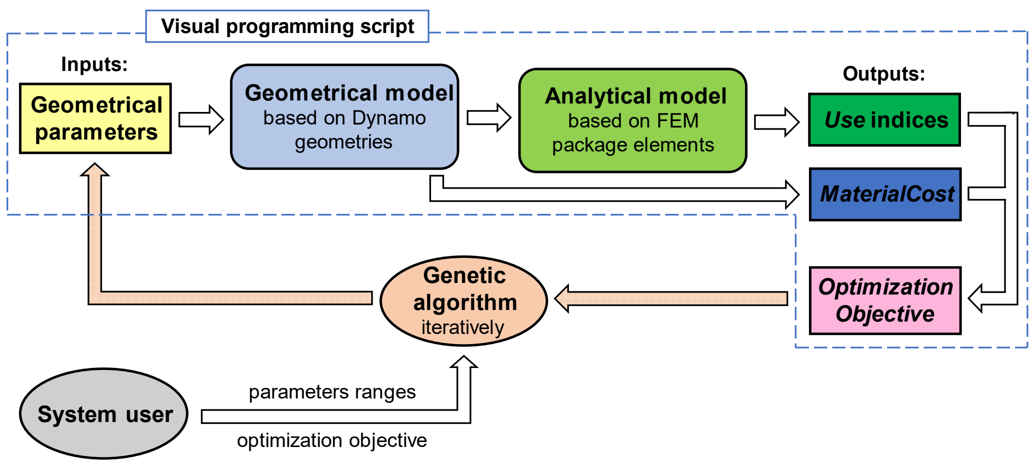
4 Optymalizacja z użyciem programowania graficznego i algorytmu genetycznego

Mnogość zmiennych czyni projektowanie mostów bardzo złożonym zadaniem. Stosowana w tradycyjnym procesie projektowania optymalizacja z kilkoma krokami iteracji jest czasochłonna i zwykle obejmuje zbyt małą liczbę analizowanych wariantów. Towarzyszą temu liczne ograniczenia, np. czasowe (narzucony harmonogram projektu) i związane z dostępem do danych projektowych na określonym etapie realizacji procesu (projekt koncepcyjny, wykonawczy itd.). Skutkuje to nieoptymalnymi wyborami ostatecznego wariantu.

Techniki optymalizacji konstrukcji mostów są przedmiotem licznych i obszernych badań. Ze względu na funkcję tych obiektów i często istotne aspekty estetyczne, optymalizacja nie może ograniczać się już tylko do wielokrotnego użycia modelu analizowanego w środowisku programu MES. Coraz częściej te procesy są automatyzowane dzięki technikom programistycznym. Rozwiązania te zwykle wykorzystują typowe języki programowania tekstowego (np. Python, C#). Nie są one jednak łatwe w użyciu dla użytkowników nie będących programistami, a więc również inżynierów budowlanych. Odpowiedzią na te potrzeby może być programowanie graficzne (ang. visual programming). Języki programowania graficznego są często implementowane w środowiskach BIM, gdzie parametryczność i możliwość szybkiej edycji są kluczowe. Skrypty (algorytmy) programowania graficznego najczęściej automatyzują parametryczne modelowanie geometrii, ale mogą też pomagać w rozwiązywaniu innych zadań. Połączone z algorytmami optymalizacyjnymi tworzą zautomatyzowany proces projektowania, który nazywany jest projektowaniem generatywnym (ang. generative design). W tym podejściu użytkownik ustala cele i obostrzenia parametrów zadania, a algorytm dobiera optymalne wartości.

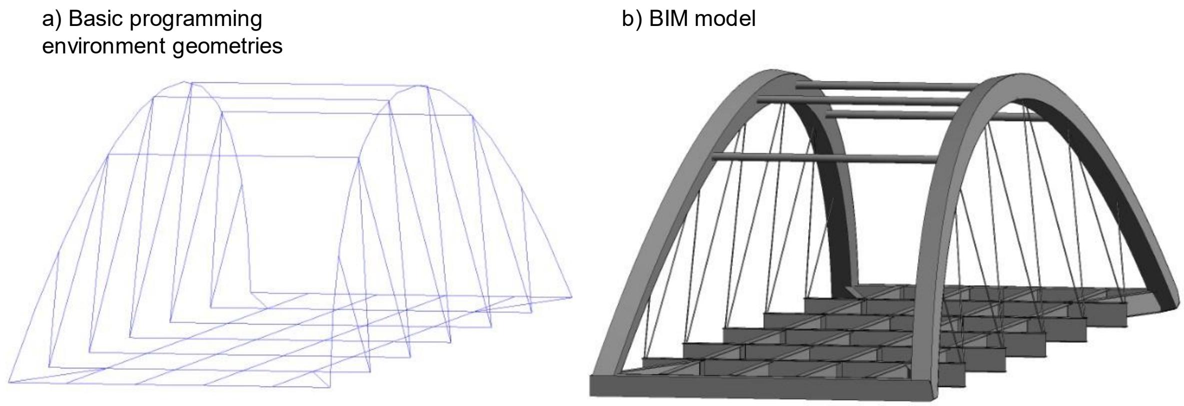
Projektowanie generatywne w powiązaniu z programowaniem graficznym może skutecznie wspomagać realizację czasochłonnych iteracyjnych zadań optymalizacyjnych. Przegląd literatury wykazał jednak, że implementacja tego podejścia do projektowania mostów jest wciąż jeszcze niedostatecznie rozpoznana, szczególnie w zakresie analiz konstrukcyjnych. Wynika to również z ograniczeń dostępnych narzędzi. Nie wszystkie środowiska programowania graficznego dedykowane inżynierii lądowej dostarczają odpowiednich narzędzi MES. A bez integracji modeli BIM i MES oraz użycia w nich automatyzujących algorytmów trudno jest zbudować poprawnie działającego cyfrowego bliźniaka mostu.

Autor podjął się próby [183] wykorzystania podejścia generatywnego do optymalizacji konstrukcji mostów z użyciem popularnego wśród inżynierów budownictwa języka programowania graficznego Dynamo. Pokazana została integracja modeli BIM i MES z algorytmami optymalizacyjnymi, ale użytymi w sposób zautomatyzowany. Środowisko Dynamo zostało wzbogacone przez autora o utworzony pakiet metod umożliwiający analizy MES (Rys. 2). Dzięki temu możliwa była analiza konstrukcji bezpośrednio w skrypcie programowania graficznego, który jednocześnie może również automatyzować proces modelowania BIM.



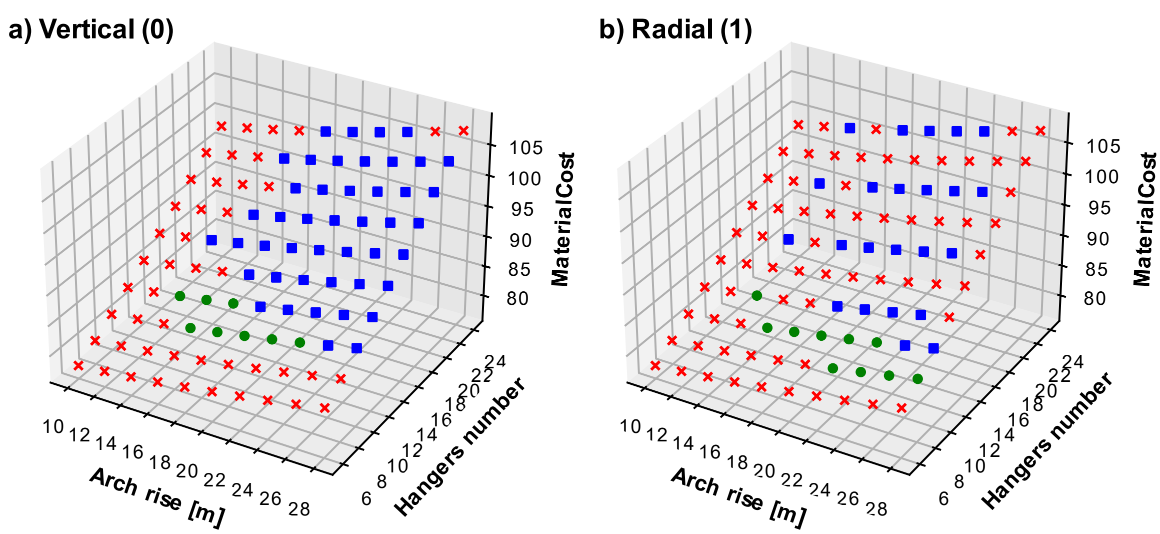
Rys. 2. Schemat procesu optymalizacyjnego projektowania generatywnego

W przeprowadzonym eksperymencie, utworzony skrypt Dynamo generował dużą liczbę sparametryzowanych modeli będących wariantami pewnego mostu łukowego. Wizualizację podstawowego modelu MES i BIM tego mostu pokazano na Rys. 3. Zmieniające się parametry obejmują układ wieszaków w czterech wariantach: pionowym (vertical), promienistym (radial), ukośnym (oblique) i siatkowym (network). Innymi parametrami są również rozstaw wieszaków lub inaczej ich liczba (hangers number), strzałka łuku (arch rise) oraz wymiary podstawowych elementów konstrukcyjnych i właściwości materiału.

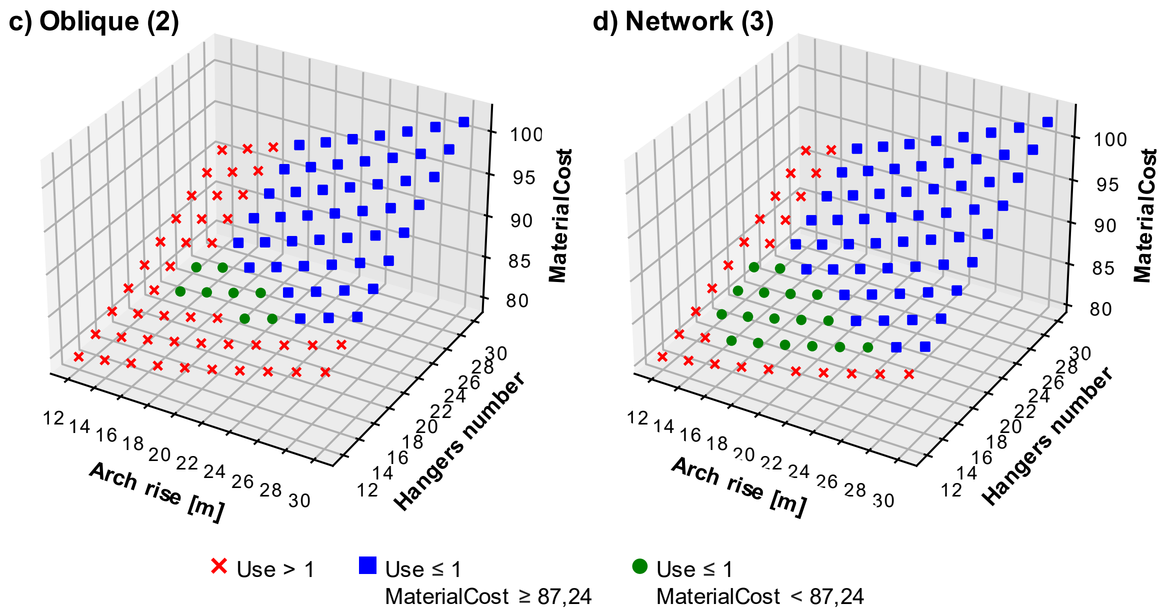


Rys. 3. Wizualizacje modeli mostu łukowego a) podstawowa geometria modelu MES b) wygenerowany modelu BIM

Generowane warianty modelu były następnie analizowane metodą MES. Połączony z systemem algorytm genetyczny pobierał rezultaty analiz jako parametry wejściowe do następnej generacji modeli. Algorytm iteracyjnie sterował parametrami geometrycznymi i materiałowymi, dążąc do minimalizacji przyjętej funkcji celu. Przyjętą funkcją celu był koszt zużytego materiału zależny od jego objętości i współczynnika ceny. Wyniki jednej z przedstawionych analiz pokazane są w postaci wykresów widocznych na Rys. 4. Jest to analiza wstępna rozwiązań o wybranych parametrach. Zadanie polegało na sprawdzeniu jaki zakres tych parametrów powinien zostać uwzględniony w poszerzonych analizach. Zakresy te zostały wyznaczone przez poszukiwania rozwiązań o dopuszczalnym wytężeniu (Use ≤ 1) i mniejszym koszcie materiału w stosunku do konstrukcji porównawczej (MaterialCost < 87,24).



Rys. 4 (cz. 1). Analiza wstępna rozwiązań o różnych parametrach (typ systemu wieszaków i ich liczba oraz strzałka łuku) z odniesieniem do maksymalnego wytężenia i kosztu porównawczego



Rys. 4 (cz. 2). Analiza wstępna rozwiązań o różnych parametrach (typ systemu wieszaków i ich liczba oraz strzałka łuku) z odniesieniem do maksymalnego wytężenia i kosztu porównawczego

5 Generowanie syntetycznych chmur punktów dla uczenia maszynowego

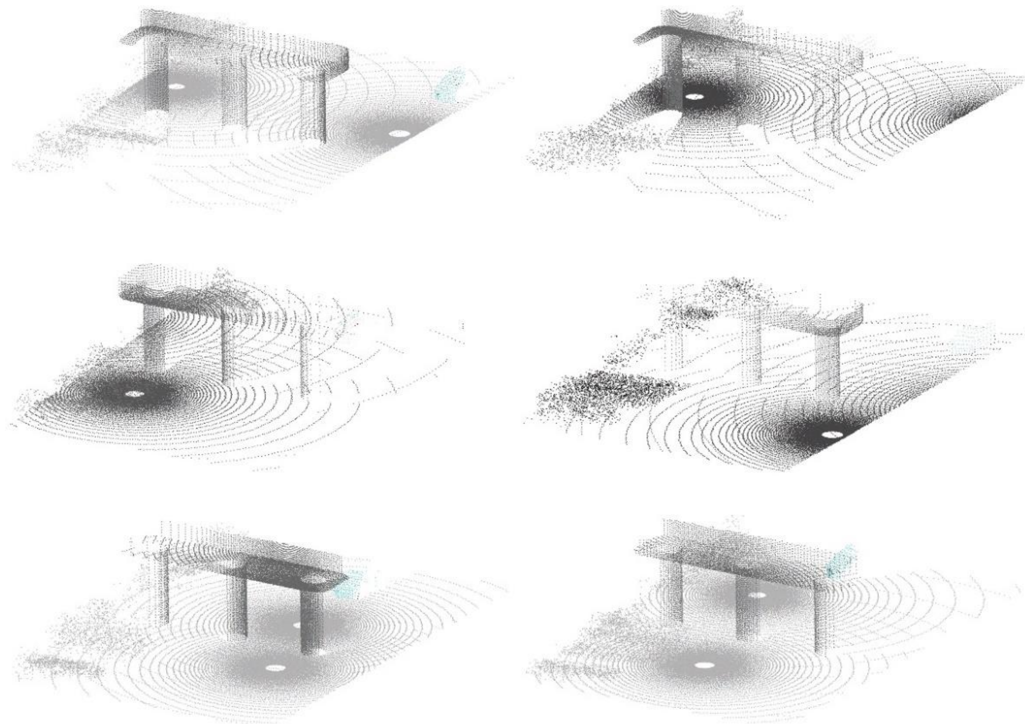
Chmury punktów coraz częściej stają się podstawowym źródłem geometrycznych danych o obiektach infrastruktury i to nie tylko mostowej. Dane geometryczne są jednak niewystarczające do wirtualizacji obiektów w postaci cyfrowych bliźniaków i prowadzenia dalszych kompleksowych analiz. Do tego potrzebne są bowiem również dane semantyczne. Manualna ekstrakcja semantycznych danych z chmur punktów jest czasochłonna, podatna na błędy i nieefektywna. Dlatego automatyzacja tego procesu wciąż jest przedmiotem wielu badań. Sposoby automatyzacji procesów chmur punktów często bazują na technikach uczenia maszynowego (ang. machine learning), co zostało wykorzystane również w tej pracy.

Skutecznie działający system uczenia maszynowego opiera się na dwóch fundamentach: poprawnym algorytmie i odpowiednim zbiorze danych do jego treningu. Wraz ze wzrostem możliwości algorytmów i mocy obliczeniowych, często to właśnie dane do treningu są ograniczeniem w stosowaniu tego podejścia. Odpowiedni zbiór takich przykładów treningowych powinien być zdywersyfikowany, miarodajny i powinien być odpowiednio liczny. Jeśli dodatkowo np. wybrany algorytm należy do grupy uczenia nadzorowanego (ang. supervised learning), to wówczas dane muszą być jeszcze etykietowane (ang. annotated, labeled). Niewystarczająca dla złożoności zadania liczba przykładów zmniejsza bowiem efektywność działania systemów uczenia maszynowego.

Pozyskiwanie chmur punktów jest czasochłonne i drogie. Skanowanie niektórych miejsc niesie trudności praktyczne (np. konieczność zamknięcia drogi przy skanowaniu dolnej części wiaduktu). Z kolei manualne etykietowanie chmur punktów, tak by mogły być użyte w treningu algorytmów, jest nieefektywne. To sprawia, że duże zbiory chmur punktów do treningu algorytmów są rzadkością. Efekt braku odpowiedniej liczby danych jest widoczny w rozwiązaniach uczenia maszynowego, które przeznaczone są do automatyzacji procesów analizy chmur punktów. Jedną z metod na pokonanie tego problemu jest generowanie danych syntetycznych. Dane syntetyczne mają pewne zalety w stosunku do rzeczywistych. Przede wszystkim łatwiej je pozyskać, bo nie ma potrzeby prowadzenia wielokrotnego skanowania prawdziwego obiektu, które jest kosztowne i czasochłonne. Dane te są też skalowalne. Ich liczba jest ograniczona właściwie tylko mocami obliczeniowymi. Dają też kontrolę nad parametrami i dystrybucją przykładów, a także zapewniają efektywne etykietowanie. Niemniej, aby dane syntetyczne mogły zastąpić lub wspomagać dane realne w procesie trenowania algorytmów, muszą one odwzorowywać charakterystyki realnych przykładów.

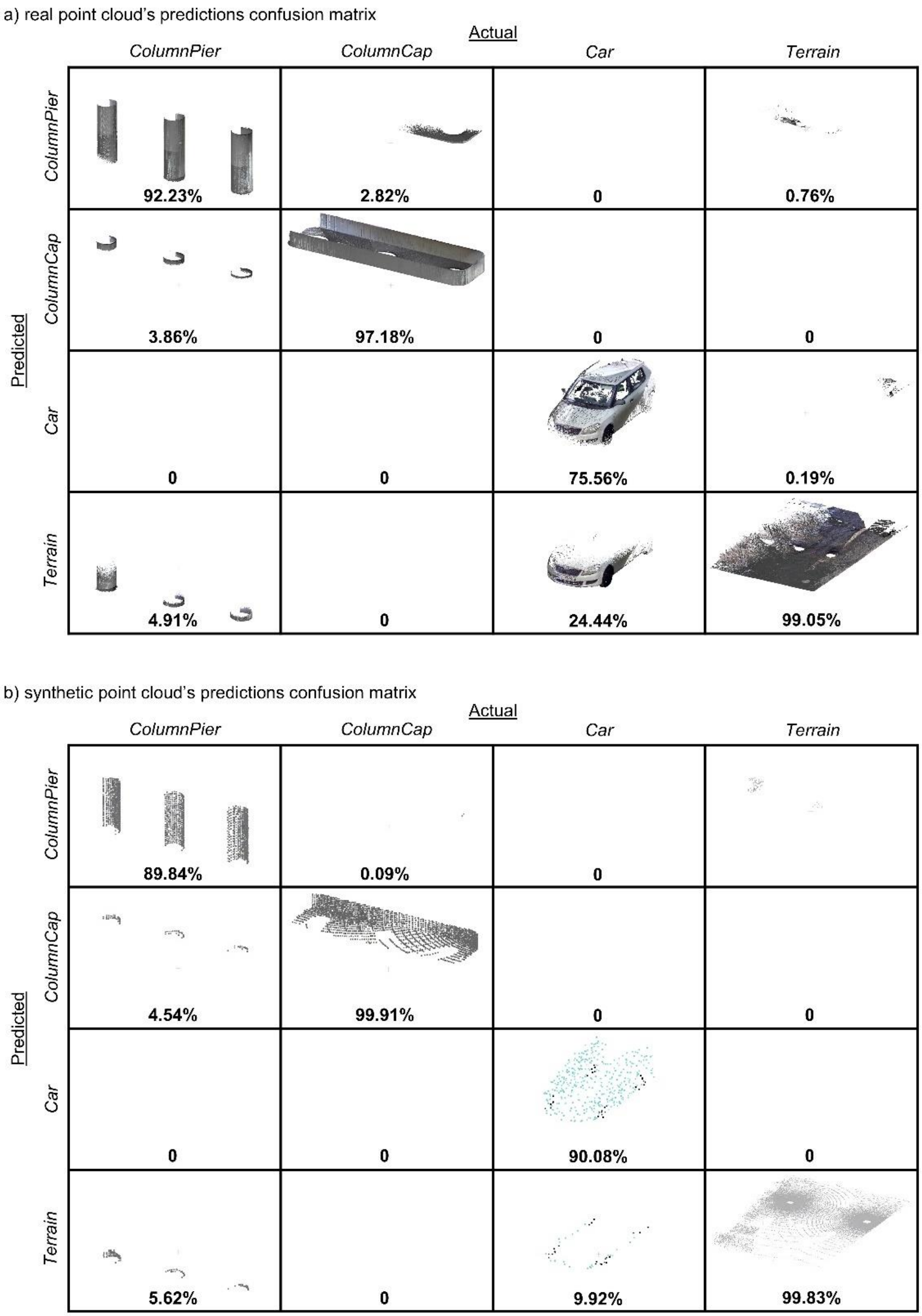
Syntetyczne chmury punktów mogą być pozyskiwane poprzez proste próbkowanie modeli geometrycznych lub symulatory skanerów. Opisy symulatorów są dostępne w literaturze [331,332,334–336,338,340,342,343]. Wciąż jednak brakuje narzędzi dających elastyczną kontrolę nad parametrami symulacji i właściwościami generowanych chmur punktów. Pożądane są więc metody, które odwzorowują działanie prawdziwego skanera i które nie wymagają złożonych prac przygotowawczych. Na przykład polegających na tworzeniu specyficznych modeli jedynie na potrzeby symulacji procesu skanowania.

Autor podjął się więc próby generowania syntetycznych chmur punktów bezpośrednio w środowisku BIM. Na te potrzeby, utworzony został symulator skanowania laserowego nazwany DynamoPCSim. Na Rys. 5 pokazano przykłady wygenerowanych w pracy syntetycznych chmur punktów. Jest to filar mostu drogowego, na którym przeprowadzono eksperymenty terenowe. Opracowany symulator wykorzystuje technikę śledzenia promieni (ang. ray tracing), która naśladuje działanie skanerów laserowych. Dzięki temu utworzone chmury punktów mają charakterystyki zbliżone do tych realnych. Symulator został zaimplementowany w postaci pakietu metod działających w otwartym środowisku programowania graficznego Dynamo. Skrypty Dynamo mogą być łączone z modelami tworzonymi wprost w środowisku BIM programu Autodesk Revit. Dzięki temu, symulator może bazować bezpośrednio na źródłowych modelach BIM bez konieczności tworzenia dodatkowych pośrednich modeli, które miałyby jedynie realizować wirtualny proces skanowania. Modułowa budowa i elastyczność programowania graficznego sprawiają, że symulator może uwzględniać różne parametry skanowania i pożądane charakterystyki chmur punktów.



Rys. 5. Przykłady syntetycznych chmur punktów wygenerowanych przez DynamoPCSim

Przeprowadzony w pracy eksperyment, który polegał na segmentacji semantycznej (ang. semantic segmentation) z użyciem sieci neuronowej potwierdził użyteczność generowanych danych do uczenia maszynowego. Zastosowana architektura sieci neuronowej PointNet [222] została wytrenowana na wygenerowanych syntetycznych chmurach punktów. Następnie przeprowadzona została walidacja na zbiorach testowych. Pierwszy z nich zawierał sceny syntetycznej chmury punktów, a drugi pochodził ze skanowania realnego obiektu. Na Rys. 6 pokazano macierze błędów predykcji sieci na przygotowanych zbiorach testowych. Uzyskane wyniki obu testów są ze sobą porównywalne w zakresie wskaźnika kompletności (ang. recall). Potwierdza to więc, że generowanie syntetycznych chmur punktów może zastąpić w treningu sieci neuronowej stosowanie trudno dostępnych chmur realnych.



Rys. 6. Macierz błędów (ang. confusion matrix) działania sieci neuronowej na a) realnej chmurze punktów oraz b) syntetycznych chmurach punktów

Utworzony przez autora symulator dotyczy pozyskiwania danych w postaci chmur punktów potrzebnych do treningu algorytmów uczenia maszynowego. Jest to pierwszy, ale nieodzowny etap tworzenia systemów, które automatyzują wykorzystywanie chmur punktów na potrzeby ekstrakcji informacji, ich segmentacji, a finalnie nawet generowania i aktualizowania geometrycznych modeli cyfrowych bliźniaków. W ten sposób automatycznie aktualizowane modele będą replikowały rzeczywistą geometrię, realizując ideę modelu współistniejącego z fizycznym obiektem. W przypadku mostów, informacje o zmianach geometrycznych umożliwią między innymi wykrywanie i porównywanie deformacji, monitorowanie stanu technicznego konstrukcji i alarmowanie o niebezpiecznych zjawiskach.

6 Podsumowanie i wnioski

Nabierająca rozpędu cyfrowa transformacja niemal wszystkich dziedzin naszego życia stanowi duże wyzwanie, ale i szansę. W kontekście coraz bardziej przenikających się różnych dyscyplin naukowych i sektorów gospodarki, paradygmat cyfrowych bliźniaków zwiększa efektywność w obszarze inżynierii lądowej i umożliwia międzybranżową współpracę na cyfrowych platformach wymiany danych. Dostarcza też nowych, praktycznych narzędzi, które wspomagają realizację codziennych zadań wykonywanych przez inżynierów i zarządców infrastruktury.

Przedstawiona w rozprawie koncepcja cyfrowego bliźniaka mostu jest jedną z pierwszych prób zdefiniowania tych informacyjnych modeli w odniesieniu do konstrukcji mostowych. Rozprawa omawia techniki, które mogą być wykorzystywane do tworzenia i użytkowania cyfrowych bliźniaków mostów oraz opisuje ich zasadnicze cechy. Zaproponowana koncepcja tych złożonych wirtualnych obiektów obejmuje ogólne zasady tej idei, które przedstawiane są w coraz bogatszej literaturze przedmiotu. Uwzględnia też przy tym specyfikę konstrukcji mostów, a także zwyczajowe wymagania stawiane przez potencjalnych użytkowników cyfrowych bliźniaków, czyli inżynierów mostowych oraz właścicieli i zarządców infrastruktury mostowej. W ten sposób, cyfrowy bliźniak mostu może być przygotowany do interakcji z innymi obiektami w wirtualnym środowisku. Może też dostarczać realnych korzyści swoim użytkownikom. Te korzyści to między innymi automatyzacja prac projektowych, monitorowanie postępów robót na budowie, skuteczna ocena stanu technicznego konstrukcji, czy efektywne zarządzania zasobami. Są one realizowane przez wprowadzenie nowych sposobów użytkowania cyfrowych bliźniaków. Dwa przykłady takiego nowego podejścia zostały zaproponowane i szczegółowe opisane przez autora w rozprawie.

Pierwszy z nich to automatyzacja prac projektowych z użyciem programowania graficznego i optymalizacyjnego algorytmu genetycznego [183]. W tym podejściu pokazano w jaki sposób algorytmy programowania graficznego mogą zostać wykorzystane do praktycznej integracji modeli BIM oraz MES. A taka integracja jest nieodzowna w przypadku kompletnych cyfrowych bliźniaków obiektów mostowych. Z kolei użyty w tym zadaniu algorytm optymalizacyjny pozwolił na analizę zdecydowanie większej liczby wariantów niż w tradycyjnym sposobie projektowania. W rezultacie prowadzi to do uzyskania bardziej optymalnych rozwiązań przy znacznie większej efektywności całego procesu.

Drugi z zaproponowanych sposobów dotyczy generowania i aktualizacji geometrii cyfrowych bliźniaków również w fazach wznoszenia i użytkowania. Jest to propozycja tworzenia zbiorów syntetycznych chmur punktów do trenowania algorytmów uczenia maszynowego. Algorytmy takie mogą analizować chmury punktów, wyciągać z nich semantyczne informacje, a finalnie również generować lub uaktualniać geometryczne modele. Automatycznie aktualizowane modele będą w przyszłości mogły replikować rzeczywistą geometrię mostu, realizując ideę modelu współistniejącego z fizycznym obiektem. Akurat w przypadku mostów, informacje o zmianach geometrycznych umożliwią między innymi wykrywanie i porównywanie deformacji, monitorowanie stanu technicznego konstrukcji i alarmowanie o niebezpiecznych zjawiskach.

Zagadnienia przedstawione w rozprawie stanowią początkowy etap badań nad cyfrowymi bliźniakami mostów i powinny być dalej rozwijane. Kierunki dalszych prac dotyczą zarówno teoretycznych podstaw definiujących te wirtualne obiekty jak również praktycznych sposobów ich użytkowania. Przede wszystkim należy określić techniczne szczegóły implementacji zaproponowanych w pracy właściwości cyfrowych bliźniaków. W przyszłości, koncepcja ta będzie wymagała uwzględniania również nowych, dopiero tworzonych technologii. W odniesieniu do automatyzacji procesu projektowego z użyciem programowania graficznego i algorytmów optymalizacyjnych, należałoby już uwzględnić dodatkowe parametry, które są trudne do stosowania przy tradycyjnym podejściu do projektowania. Dzisiaj są to najczęściej wymierne koszty rozumiane w aspekcie ekonomicznym. Natomiast w przyszłości coraz większą rolę będą odgrywały również koszty środowiskowe i społeczne. A takie analizy umożliwia dopiero wdrożenie środowiska BIM.

Z kolei zaproponowany symulator syntetycznych chmur punktów stanowi jedynie pierwszy krok do tworzenia cyfrowych bliźniaków w fazie operacyjnej. W dalszych pracach syntetyczne dane powinny zostać użyte do trenowania systemów uczenia maszynowego generujących i aktualizujących modele geometryczne cyfrowych bliźniaków. To z kolei otworzy możliwość tworzenia systemów eksperckich wykrywania zmian geometrycznych i alarmowania o potencjalnych niebezpieczeństwach.

Przeprowadzone rozpoznanie stanu wiedzy w zakresie cyfrowych bliźniaków mostów, prace koncepcyjne oraz eksperymenty in-situ pozwoliły zrealizować wszystkie zakładane na początku rozprawy cele. Zdefiniowano podstawy tworzenia cyfrowych bliźniaków mostów proponując szablon tych wirtualnych modeli, który uwzględnia specyfikę obiektów mostowych i fazy ich cyklu życia. Dzięki temu możliwe było pokazanie możliwości wirtualizacji już nie tylko samego fizycznego obiektu, ale również towarzyszących mu procesów. Autor ma świadomość, że przedstawiony materiał nie wypełnia wszystkich założeń koncepcji cyfrowych bliźniaków. Omówione badania i nabyte dzięki nim kompetencje są jednak zaczątkiem do kolejnych działań, które będą zmierzać do coraz bardziej kompletnych implementacji cyfrowych bliźniaków obiektów mostowych w przyszłości.