Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

A BIM Based Framework for Damage Segmentation, Storage, and Visualization

Mathias Artus [1,*]
Mohamed Alabassy [1]
Christian Koch [1]

[1] *Bauhaus-Universität Weimar, Marienstraße 13a, 99423 Weimar, Germany*
[*] *corresponding author: mathias.artus@uni-weimar.de*

Paper-based data acquisition and manual transfer between incompatible software or data formats during inspections of bridges, as done currently, is time-consuming, error-prone, cumbersome, and leads to information loss. A fully digitized workflow using open data formats would reduce data loss, efforts, and costs of future inspections. On the one hand, existing studies proposed methods to automatize data acquisition and visualization for inspections. These studies lack an open standard to make the gathered data available for other processes. On the other hand, several studies discuss data structures for exchanging damage information among different stakeholders. However, those studies do not cover the process of automatic data acquisition and transfer. This study focused on a framework that incorporates automatic damage data acquisition, transfer, and a damage information model for data exchange. This enables inspectors to use damage data for subsequent analyses and simulations. The proposed framework shows the potentials for a comprehensive damage information model and related (semi-)automatic data acquisition and processing.

Keywords: Building Information Modeling; Defects; Damage Information Modeling; Life cycle; Bridges; Inspection; Maintenance

1 Introduction

Bridges are designed to last for more than 50 years. During this period they consume up to 50% of their life-cycle costs [1, 2]. In conventional bridge inspections data acquisition

© 2022 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

and exchange is performed in the form of paper-based reports as well as proprietary and incompatible data formats, so any data transfer is done manually. Inspections start with an inspector or an engineer performing the visual inspection on-site. All defects and related data are recorded in paper-based reports and the inspector subsequently digitizes the data later in the office. Other stakeholders, such as structural engineers, retrieve this data either as paper-based reports, as exports of databases, or in other proprietary data formats. Then, engineers would have to integrate or import this data again into their digital models. Such repetition of manual data digitization is cumbersome, time consuming, and leads to information loss [3, 4]. The concept of Building Information Modeling (BIM) has been designed to overcome these issues. BIM should cover the entire life-cycle of structures including the operation phase with inspections and maintenance [4]. Defects and related information are vital for the inspection process. However, Sacks et al. remarked that "There is currently no accepted, consistent or thorough way to represent the defects that may occur in bridges." [5], p. 144]. This leads to the conclusion that BIM requires an extension to support the inspection process.

Hereinafter, the concept of modeling defects and related information will be called Damage Information Modeling (DIM) and the associated model as Damage Information Model. Through out the life cycle of a BIM model, several states, such as as-planned, as-designed, as-built, and as-is are known. The operation phase includes inspections to ensure the safety of a structure. Inspections deliver additional information of a structure, for instance, measurements, report documents, or photos. Based on this information, defects are identified. Models including damage information will be called as-damaged models. The as-damaged model provides required information to the related domains, for example, inspection, simulation, and material analysis. Figure 1 shows a conceptual overview of the as-damaged model and some related processes. This study aims to provide a framework that covers data acquisition and visualization. The frame work is designed with open data formats in mind enabling integration of further domains in future work.

2 Background

Several aspects are relevant for digitizing bridge inspection procedures. First, a comprehensive model of the bridge is required. BIM is an established concept in the Architecture Engineering and Construction (AEC) sector. Hence, efforts done in this sector have been investigated. On the basis of the BIM model, DIM is defined. Several studies regarding DIM have been published. Last, the automatic damage segmentation has been addressed by several studies.

2.1 Building Information Modeling

BIM is defined as the concept supporting the overall facilities' life-cycle. Figure 2 shows the building model for design, construction, operation, modification and conceptual design. The operation phase includes facility management, maintenance, and repair.

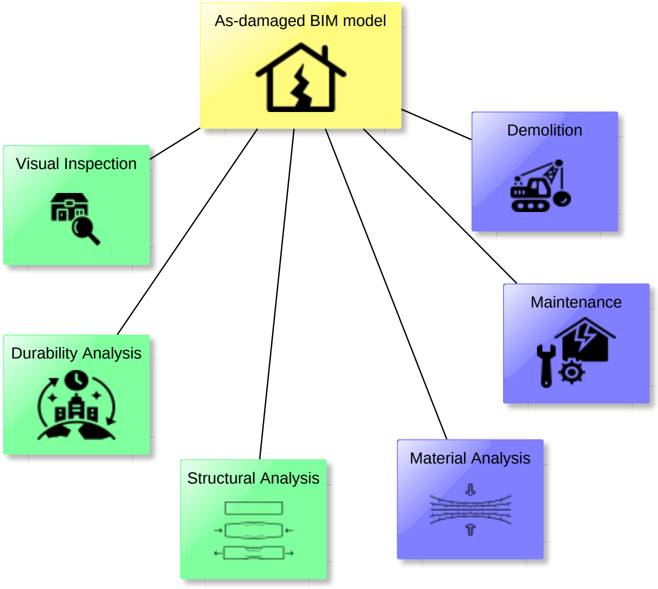


Figure 1: The as damaged BIM model as central point for involved domains during the operation phase.

Numerous studies dealt with design and construction, but less effort was put into the operation phase.

Several stakeholders are involved during the operation phase, which is similar to design, planning and construction processes. These stakeholders have to continuously exchange building information back and forth interactively. The conservative way is to utilize software for a discipline-specific tasks and exchange 2D plans and documents with third party parties; this approach is called Little BIM. Conversely, Big BIM means sharing digital building models between parties instead of 2D plans and documents. The building model may be exchanged via an open standard or proprietary data formats; this is called open or closed BIM, respectively[4]. Proprietary data formats may limit the choice of software products or exclude stakeholders that do not have access to the required software. Hence, an open standard is preferable during operations phase.

Central for the entire BIM concept are the Industry Foundation Classes (IFC). This standard defines an open data format to exchange building data. BIM was originally focused on building construction and not infrastructure. Thus, bridge models could not be properly exchanged using the IFC. This led to the extension of BIM to bridges and roads [6]. Further extensions focus on tunnels, airports, ports and harbours. Although, there is the prospect of exchanging bridge data between the different stakeholders with the help of IFC, 78% of the bridges were built before 1990. For bridges pre-dating that time, no BIM model exists, which is a prerequisite for the application of DIM. A number of studies have been conducted in the domain of 3D model generation of existing buildings via point clouds widely known as scan-to-BIM [7, 8, 9]. In accordance with

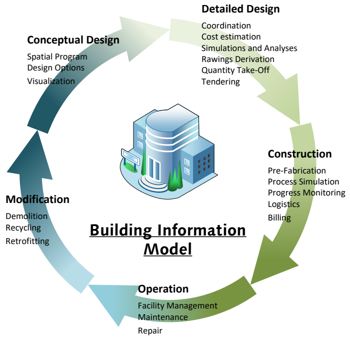


Figure 2: BIM in the life-cycle of a facility. According to [4].

such as-built models, damage information could be added to extend the use of BIM for inspection and assessment.

2.2 Damage Data Model

To define a DIM, Sacks et al. published an Information Delivery Manual (IDM) for the open data exchange during inspection and assessment [3]. The manual provides an overview of the designed inspection process called SeeBridge. The SeeBridge inspection process extends the conservative inspection process, which has been described earlier. Following tasks are part of the process: generation of the bridge model, damage acquisition, point cloud generation, damage segmentation, and the calculation procedure for performance indices. However, the IDM does not describe how to explicitly exchange damage data between the stakeholders.

Several attempts have been made to define data structures for damage information models. First, these models have to incorporate semantic information, such as measurements and relationships. This is covered by existing bridge management systems [10].

Second, visual data for example photos are part of information models as shown by Hüthwohl at al. [11]. They included the photos directly as texture in the model. However, some defects may have multiple photos but a texture with multiple photos would be challenging regarding visualization. A DIM must cover both scenarios: including a texture and including several photos.

Last, geometric data is part of DIMs. Hamdan et al. developed a ontological model to store damage data including defect geometry with relation to a damaged element. The study did not cover the generation of the geometry and related effects to the component geometry [12].

Another required information is the time relation of a defect. A defect is registered during an inspection and the geometry may change over time. Hence, the defect or defect data need a relation to time, i.e., a relation to an inspection process. This has

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

been covered by the work of Tanaka et al. [13]

Most researchers focus on the inclusion of defect geometry in general disregarding that physical defects affect the geometry of a component. Furthermore, the generation of defect geometry and transferring the damage information to other applications are missing.

2.3 Damage Data Acquisition

German et al. have proposed an image-based method to automatically detect spalled regions and related properties at reinforced concrete columns [14]. The region of spalling was first isolated by way of a local entropy-based thresholding algorithm, then the exposure of longitudinal reinforcement (i.e., depth of spalling in column) and length of spalling along the column were measured using a global adaptive thresholding algorithm in conjunction with image processing methods for template matching and morphological operations. That method was tested on a set of images for damaged RC columns, indicating its validity against manual measurements.

They later improved upon their work by adapting the aforementioned algorithms for spalling segmentation and property retrieval to sufficiently detect the absence of spalled regions on concrete surfaces, detecting transverse reinforcement and distinguishing it from longitudinal reinforcement [15]. Those enhancements enabled a better classification based on contextual information from depth retrieval pertaining to the amount and type of reinforcement which is exposed into one of five respective categories, namely: no spalling, spalling of concrete cover, no exposure of reinforcement, spalling which exposes transverse reinforcement, spalling which exposes longitudinal reinforcement, and spalling of concrete which exposes both transverse and longitudinal reinforcement.

Dawood et al. developed an integrated model based on image processing techniques and machine learning to automate consistent spalling segmentation and numerical representation of distress in subway networks [16]. A hybrid algorithm including with the support of regression analysis allows a prediction of spalling depth. A spalling processor detects distress attributes from noise-reduced images and creates 3D visualisation models of the defect. Subsequently, she regression analysis model and image processing techniques measure depth and severity of the spalling distress. The overall process and implementation were able to quantify the spalling depth in 75 images with an average validity of 93%

Wu et al. proposed spalling segmentation method by analysing surface roughness reconstructed from point clouds acquired by laser scanning [17]. In the proposed method, points on ancillary facilities are filtered via circular scan-line fitting and large residual error filtering. A roughness descriptor was used to identify high rough patches. Next, high rough areas on the tunnel surface, such as bolt holes, or segment seams were filtered as well after classifying them usig Hough transformation and similarity analysis to verify its classification. The remaining patches were presumed to be concrete spalling.

Hoang et al. proposed another approach to identify image texture for features extraction [18]. Image textures obtained from statistical properties of colour channels, grey-level cooccurrence matrix and grey-level run lengths were used as features to characterize surface condition of a concrete wall. These extracted features were classified into spalling and nonspalling classes.

A Mask R-CNN architecture developed by Facebook AI Research [19] was utilised by Borin and Cazzini [20] to segment spalling images and automatically back-project the segmented spalling patches from image plane into 3D space using the available information about the pose of the camera for each captured image and its EXIF metadata as a 2D texture laid over host elements in the 3D information model.

2.4 Damage Data Applications

Based on the damage data, different applications process the given data. The processing of data is divided into visualization and simulation. This study focuses on visualization. Visualization is necessary for assessment and planning. Planning covers several subtasks: planning of non-destructive testing, inspection, and maintenance planning.

Damage and bridge data have to be visualized for different use cases. Most studies address the visualization for the task of assessment. Bruno et al. developed a framework to assess historic buildings. The Historic Building Information Model (HBIM) includes inspection and survey results to provide a structured overview of defects to engineers [21]. The visualization mainly consists of tabular structured text and related photos.

Based on SfM and laser scans, 3D point clouds of damaged bridges may be generated. Torok et al. have shown how to generate those point clouds and highlight damaged sections in such a point cloud [22]. This approach provides a 3D surface model of the damaged bridge. The assessment needs overviews of registered defects and related components. A labeled point cloud does not provide a view like that.

BIM aims to incorporate raw data, such as photos, measurements, and point clouds; furthermore, conclusions and deductions from stake holders, for example, task planning or conditions states. Chan et al. have used a BIM model of a bridge and added the condition state of the individual components. Additionally, they highlighted the benefits of proprietary software that includes images and textual information [23]. These two different visualizations are the first step towards virtual model based inspections.

A reasonable assessment requires also geometric information, such as the defect position. McGuire et al. [24] target this problem by adding damage cubes to the BIM model. Visualizing damage and bridge data according to the requirements of national inspection manuals has been covered very well. However, the proposed approaches are not open for additional tasks, views, and use cases. This is necessary because it is not possible to predict future use cases. Hence, this studies focused on addressing assessment requirements and using open standards.

Apart from visualization, using 3D damage data may be utilized for structural analyses. Based on geometric damage information, the FEA model may be generated by using adapted material parameters [25]. However, a data model to exhange damage data with third party software is not defined in this study. Hamdan et al. developed a framework for bridge assessment based on a linked model approach [26]. Such linked models have the disadvantage of data dispersion and for later data exchange it is beneficial if all data is stored in a single file. Isailović proposed an open framework on the basis of point

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

clouds [27]. However, the photos for damage detection are generated based on the point cloud that may lead to information loss. To omit this limitation, this study used photos of the inspected bridge as input.

2.5 Problem Statements and Objectives

Several frameworks that generate point clouds and damage information from images and perform FEAs exist [26, 25]. However, these models focus on supporting FEA only; other applications or domains have not been considered. Other frameworks, which are focused on the conservative inspection process, lack supporting functionality for FEA, durability simulations, or special planning tasks, e.g., for non-destructive testing [21, 28]. The aim of this study is to provide a framework utilizing open standards enabling the integration of further domains in future applications. The Framework is based on a DIM developed in prior studies [29, 30]. The following research questions are addressed by this study:

- How are damage geometries generated on the basis of photos?
- How are geometric as-damaged BIM models generated?
- How could additional data, for instance documents or photos, be added to the geometric as-damaged BIM model?

3 Methodology

Figure 3 shows the entire framework with raw damage data on the left side in blue. The steps for data processing are depicted in yellow second from left, and the resultant BIM data on the second from right in red. Possible subsequent processes are depicted in green on the right side. The bridge inspection delivers the required damage data, such as photos, measurements, and textual descriptions. To detect defects and generate the related geometry, the damage segmentation and geometry generation processes take the visual data. Subsequent, the damage alignment defines the position of the defects at their related components using the defect geometry and the BIM Model of the bridge. Finally, semantic damage data is added manually to the geometric as-damaged BIM model. The resultant model is used for later planning, analysis and assessment. Planning is necessary for further surveys, for example, ultrasonic or impact-echo.Furthermore, with the 3D as-damaged BIM model, 3D Finite Element Analyses (FEA) may be setup easier because the geometry of damaged components may be directly imported into the simulation environment. Furthermore, probabilistic simulations may utilize semantic and geometric data. Last, the assessment of the bridge is based on all data incorporated by the as-damaged BIM model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

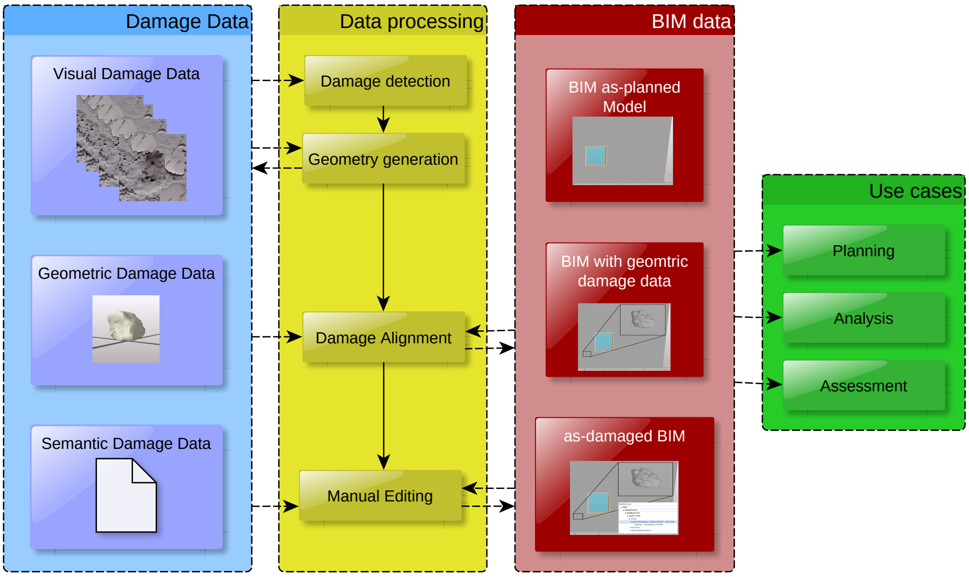


Figure 3: Framework overview

4 Data Input

The primary input data consists of photos This section describes how the point cloud and defect geometry is generated. Additionally, the process of aligning the resultant defect geometry to the original BIM model is explained.

4.1 Defect Geometry Generation

A five-step workflow, as shown by Figure 4, generates the spalling geometry on the basis of photos. First, photos of a checkerboard pattern are used to calibrate the camera. Intrinsic parameters and the radial distortion coefficients, which are required for 3D reconstruction respectively for undistoring imaged and their segmented masks, are defined by this calibration.

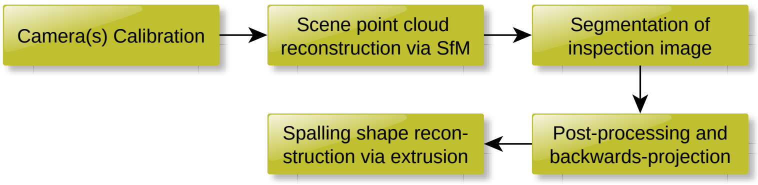

Camera(s) Calibration
Scene point cloud reconstruction via SfM
Segmentation of inspection image
Post-processing and backwards-projection
Spalling shape reconstruction via extrusion

Figure 4: Process for generating the geometry.

Based on inspection images and the intrinsic parameters estimated from calibration, a dense point cloud of the region of interest is reconstructed in the second step. The process of generating the point cloud may be accelerated separating the images into clusters and generate point cloud submodels. Furthermore, this modification limit subsequent segmentation and reduces the size of submodels without defect geometries. Third, the segmentation of the inspection images is performed. Figure 5 shows an exemplary series of images to illustrate the segmentation. 5 (a) shows a spalling in concrete. The resultant damage probability map after damage segmentation is depicted by Figure 5 (b). Finally, Figure 5 (c) shows the defect with the binary damage map superimposed.

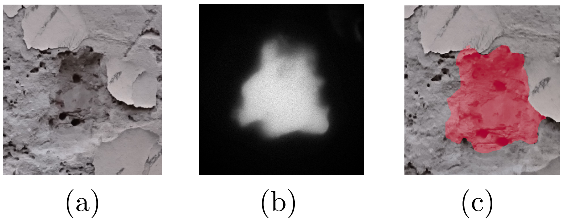


Figure 5: a) cropped part of the inspection photo. b) spalling map. c) overlay of the indentified spalled region and the inspection photo.

Fourth, given the point clouds of the scene reconstructed through OpenSfM and all the depth information entailed from that process, the regions of interest classified as spallings in the prediction maps could be converted from pixel units into 3D world coordinates. Identified spalling pixels from the submodels are scaled to metric units and merged back into one point cloud with separate labels. Figure 6 a) shows the points of the segmented spalling.

Fifth, to reconstruct the shape of the spalling, the unit vector for the extrusion direction is derived from the arithmetic mean of all conformed normals pointing into the direction of the camera for all vertices of each segmented spall patch in the point cloud. Each main patch is defined by merging overlapping labelled patches from different images via a boolean union operation after filtering out duplicated vertices, if they exist, which has resulted in the creation of a significantly larger elongated mesh that extends along the lower external edge of the primary beam. Extruding along that unit vector for a distance significantly larger than the depth of the hosting damaged beam ensures the generation of a valid defect shape with an outer surface always protruding from the surface of the hosting building element. This overdimensioning of the subtraction geometry avoids the possibility of erroneous boolean difference operations when creating the voids that was reported by Isailovic et al. [27].

The described process of geometry generation requires the alignment of the point cloud that has been generated from the BIM model and from the SfM. This alignment is achieved by using the GoICP algorithm [31, 32]. Figure 6 b) shows the resultant extrusion of the points marked as spalling. The multitude of points in the point cloud is not required for subsequent finite element analysis or visualization, and hence, the geometry is simplified to limit the memory required. Therefore, Figure 6 c) shows a geometry with reduced details, which may be observed by decreased roughness of the surface on the front.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

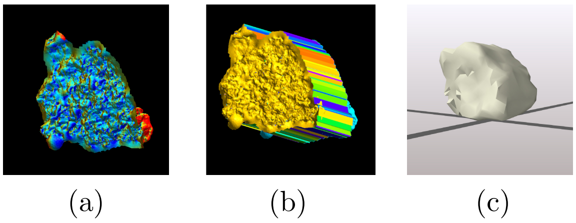

(a) (b) (c)

Figure 6: Steps of construction the spalling geometry. a) points of the spalling in 3D space. b) extrusion of the points in the direction of the mean normal vector. c) resultant subtraction geometry.

4.2 Defect Geometry Alignment

This process requires the as-built BIM model, the point cloud of the damaged model, and the defect geometry. Aligning the defect geometry correctly in the resulting BIM model requires the coordinate transformation between the generated point cloud and the BIM model. For this purpose, a BIM model of the afflicted structure or component is generated and a dense point cloud is calculated based on this model. Using a globally optimal ICP (GoICP) algorithm [31, 32] the coordinate transformation between the two point clouds is generated.

Figure 7 shows an exemplary series of photos during the alignment of the defect at a wall. The front view of the wall with the red region in Figure 7 (a) shows the position of the defect at the wall. Figure 7 (b) and (c) shows a plan view of the aligned point clouds. The red area depicts the point cloud from the SfM and turquoise points show the BIM point cloud.

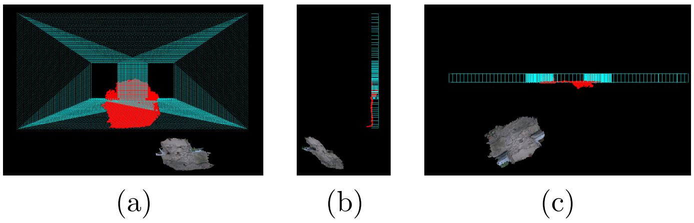

(a) (b) (c)

Figure 7: Transformed source point cloud after applying the estimated transformation displayed in red that results from aligning the 3D reconstructed point cloud of the scene with grey-toned real texture colors to the point cloud of the wall in the 3D model displayed in cyan using GoICP shown from elevation in subfigure (a), side view in (b) and plan in (c).

5 As-Damaged BIM

The data model is the central exchange point for the entire framework. Section 2.2 discussed the requirement of an open standard to support multiple domains. Figure 8 shows an UML class diagram of the final data model. Blue elements are semantic data. Visual data is depicted in green and yellow elements refer to geometric data. Bridge related classes are colored in gray.

A single defect is represented by a *DefectAnnotation* with a name, id, and description. Such a defect needs a relationship to the affected component; this relationship is realized with an objectified relationship either the *DefectProductRelation* or the *DamageGeometryCutout*. The first relationship only represents that a defect affects a component. The latter includes the geometric impact of a defect, such as the subtraction in case of a spalling. Objectified relationships may carry additional information, for example, a name and description. Furthermore, the defect has a *DamageType*; damage types may be crack, spalling or corrosion. Some defects need additional measurements, such as width or depth. This information is stored in a *Measurement* object. Several *Measurements* are grouped into a *MeasurementSet*. Defects may have relationships to other defects, such as a cause or effect that is stored in the objectified *DefectCause* relationship. Last, additional documents may provided for further information about a defect or damaged component. For this purpose, a *DocumentReference* is available.

Visual data primarily consist of images, e.g., photos or sketches. If these are rectified images, they may be used as textures depicted on the 3D geometry. For this purpose, the data model contains the class *Texturing*. A texture needs a defined area to be placed, this information is stored in the *TextureArea* class. The *Texturing* itself contains the URI of the image and is linked to the *DefectAnnotation*.

Geometry data are crucial for physical defects, for instance cracks or spalling. A defect annotation and a building product have one or more geometries. Multiple geometries occur if a defect has been registered repeatedly over time or if a component has an undamaged and damaged representation. To consider multiple geometries with the related context information, the geometry has a related *RepresentationContext*.

On the basis of the BIM model with the related geometric damage information, further semantic and visual data is added. Several studies proposed methods for automatic semantic enrichment. Although, this study relies on manual semantic enrichment, automatic enrichment may be used as well. One task is to add additional defects to cover further damage types, such as material changes, joint defects, or divergences from specification. Another task is to include the cause and effect relations between individual defects or group defects. To include external documents, additional photos, or textures, such data is added manually as well.

6 Implementation

The implementation is split up into image processing and data model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

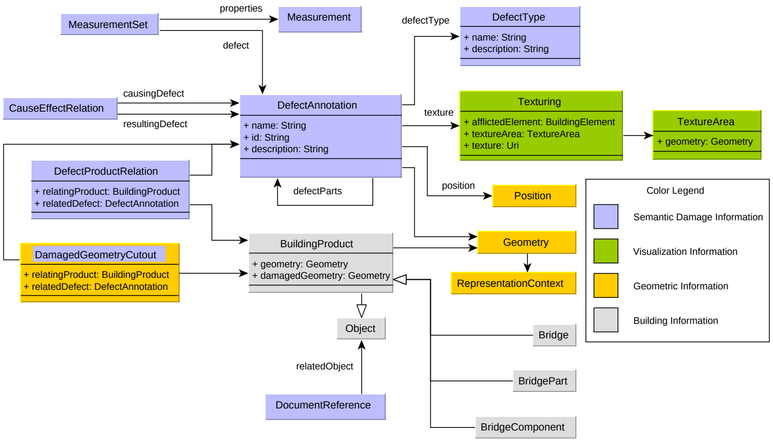


Figure 8: UML diagram of the data model. Semantic data is depicted in blue. Yellow elements refer to geometric data and green elements are related to visual data. The gray elements are related to the bridge.

6.1 Defect Geometry Generation and Alignment

The camera calibration, which consists of the intrinsic parameters and radial distortion coefficients, is calculated by processing a set of photos with checkerboard patterns via the OpenCV [33]. OpenSfM is used to generate the point clouds based on the defect photos [34]. OpenCV has the advantage of providing better quality of point clouds generated in comparison with other alternatives. Additionally, OpenSfm commands for 3D reconstruction can be run OpenSfM commands seamlessly in a background process. And backwards projection require the estimated pose of the camera that can be obtained through default methods and debug files. For damage recognition, the semantic CNN TernausNet16 has been retrained and used [35]. Aligning the defect and generating the defect geometry requires the point cloud of the undamaged building or object. This is achieved by modeling the structure or object in Revit [36], export this model into IFC, import the IFC into Blender [37] for triangulation and final generation of the dense point cloud. Next, the point cloud from the structure-from-motion algorithm and the point cloud generated based on the BIM model are aligned using CloudCompare [38]. Finally, the Python Gmsh library is used to calculate the defect geometry [39].

Convolution Neural Networks (CNN's) have been tested to provide superior performance in object recognition that has been demonstrated through the Kaggle challenges. Several work has proved this advantages in case of spalling detection and segmentation [40, 41, 27]. Hence, this study also relied on a CNN for spalling segmentation. Out of numerous CNN"s, the TernausNet16 has been chosen. This CNN is based on the the UNet including 16 layers of VGG encoder architecture [35]. Deviating from the original implementation of TernausNet16, a 5-fold cross-validation with 15 epochs per fold was used for the transfer learning process. This allowed the learning process to be performed even with a limited number of images. The dataset used for the transfer learning was

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

the Concrete Structure Spalling and Crack data base [40, 41]. Subsequently, a Jaccard index of 0.832 and validation losses of 0.173 could be measured during evaluation of the pixelwise segmentation.

6.2 As-Damaged BIM

As aforementioned in Section 2.2, this study aims to provide a framework with respect to the big open BIM concept. In compliance with this requirement, the IFC 4 standard is used for the implementation of this data model [42]. Four different entities may be used to represent the defect annotation: *IfcProxy*, *IfcVoidingFeature*, *IfcSurfaceFeature*, or *IfcAnnotation*. Proxies are generic elements that can represent every entity not included in the IFC so far, and hence, are suitable for any damage type. Voiding features lead to a subtraction of the defect geometry from geometry of the affected component. This circumstance is applicable for physical defects. Surface features represent changes on the surface of a component, such as corrosion. Annotations are designed to add further information to a component. In case of divergences from national bridge requirements or norms, these annotations may be used.

The defect product relation depends on the selection of the defect annotation. In case of using a voiding feature, *IfcRelVoidsElement* is the way to go. This relationship leads to cutting out the defect geometry from the component geometry. However, providing a context would not be possible in this case. Another possibility is to assign a defect to a component with *IfcRelAssociatesProduct*. A defect is part of a component as long as both exist; hence, an aggregation is suitable as well. For the defect cause, only the assignment is applicable.

IFC offers additional classifications via *IfcTypeObject*. Type objects are linked to the related instances via *IfcRelDefinesByType*. *IfcDocumentReference* provides the functionality to include external documents or additional photos. Properties and property sets are used to incorporate measurement data.

Geometries are stored as *IfcGeometricRepresentationItems*. Several subclasses of this class provide possibilities to model different geometries, for example, surface models, solid models, or Constructive Solid Geometries (CSG). Any geometry has a relation to a representation that again has a relation to an *IfcRepresentationContext*.

For texturing a 3D model or a part of it, the texture is provided by *IfcSurfaceTexture*. Based on the used subclass, a blob, image, or pixel texture is provided. By using a styled item and a surface style, the texture is connected to the geometric representation item. Besides the image for the texture, the texture mapping is required. The texture mapping defines which vertex of the geometry corresponds to which coordinate in the texture. This may be done via an algorithm or explicitly. IFC provides the class *IfcTextureCoordinate* to provide this mapping information.

7 Case Study

The case study aims to validate the proposed frame work. For this purpose, the "Thüringer Landesamt für Bau und Verkehr" [Thuringian Department for Construc-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

tion and Transport] provided us data of a bridge with severe defects, such as extensive spallings at the beams and corroded reinforcement. A side view of the entire bridge is shown by Figure 9. Details about the location or overall photos cannot be provided because of data privacy issues. First, photos are acquired and utilized to generate the BIM model with spalling and geometric damage information. Furthermore, additional defects from existing reports are added. The resulting model is used for subsequent planning of non destructive testing, FEA, and the assessment.

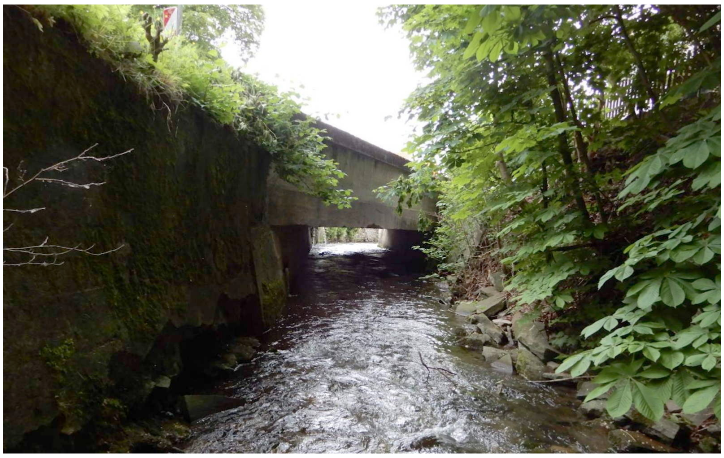


Figure 9: Bridge that has been used for the investigation. Name and location of the bridge are confidential.

7.1 Generation of BIM with Physical Defects

The photos for automatic processing have been taken manually with a Sony Alpha 7 III camera and a 28-70 mm lens. Figure 10 left shows some example photos that have been used for the damage segmentation. The spalling investigated is on a primary beam of the bridge. It shows several exposed and corroded reinforcement. 26 images with a resolution of 4000 x 6000 Pixels of this spalling have been taken. These images were used segmenting the defect as shown in Figure 10 right. In parallel, the images are used to generate the point cloud of the damaged beam. A screenshot of the point cloud generated is shown in Figure 10 right.

Figure 11 illustrates the generation of the point cloud with the defect based on the segmented defect and the point cloud of the component. Yellow points in the damaged point cloud highlight the points that are marked as damaged.

Subsequently, the point cloud with the defect information is used for generating the vertices of the defect geometry. Damaged points are used to triangulate a mesh, which is used as profile for extruding the final geometry. Figure 12 shows the point cloud with the defect as input and the resultant defect geometry.

A BIM model of the bridge has been created by using existing plans, measurements on site and Google Maps. Based on the BIM model, a synthetic point cloud of the as-built

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

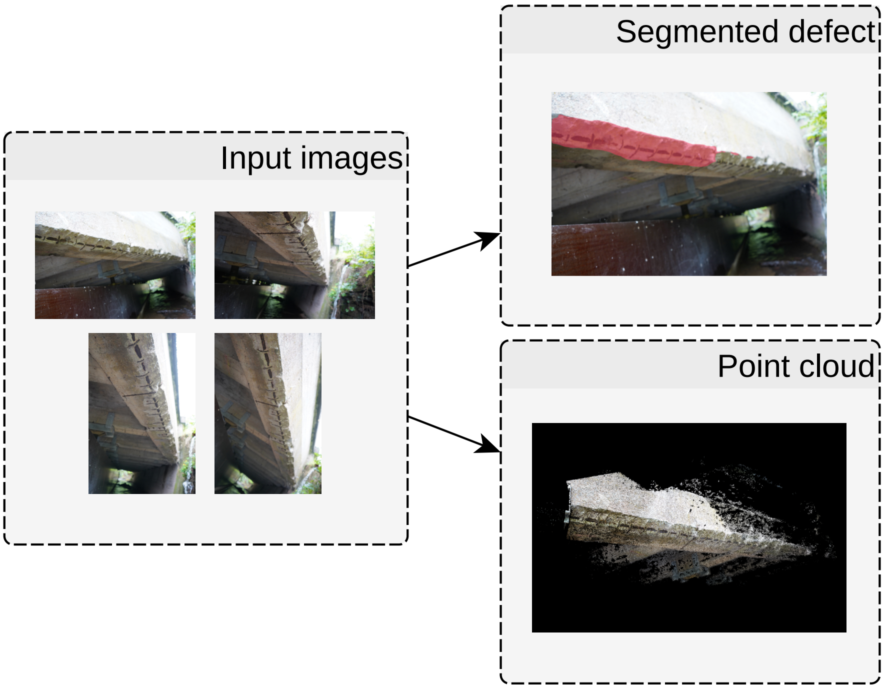


Figure 10: Point cloud reconstruction and back-projection of segmented damage based on photos.

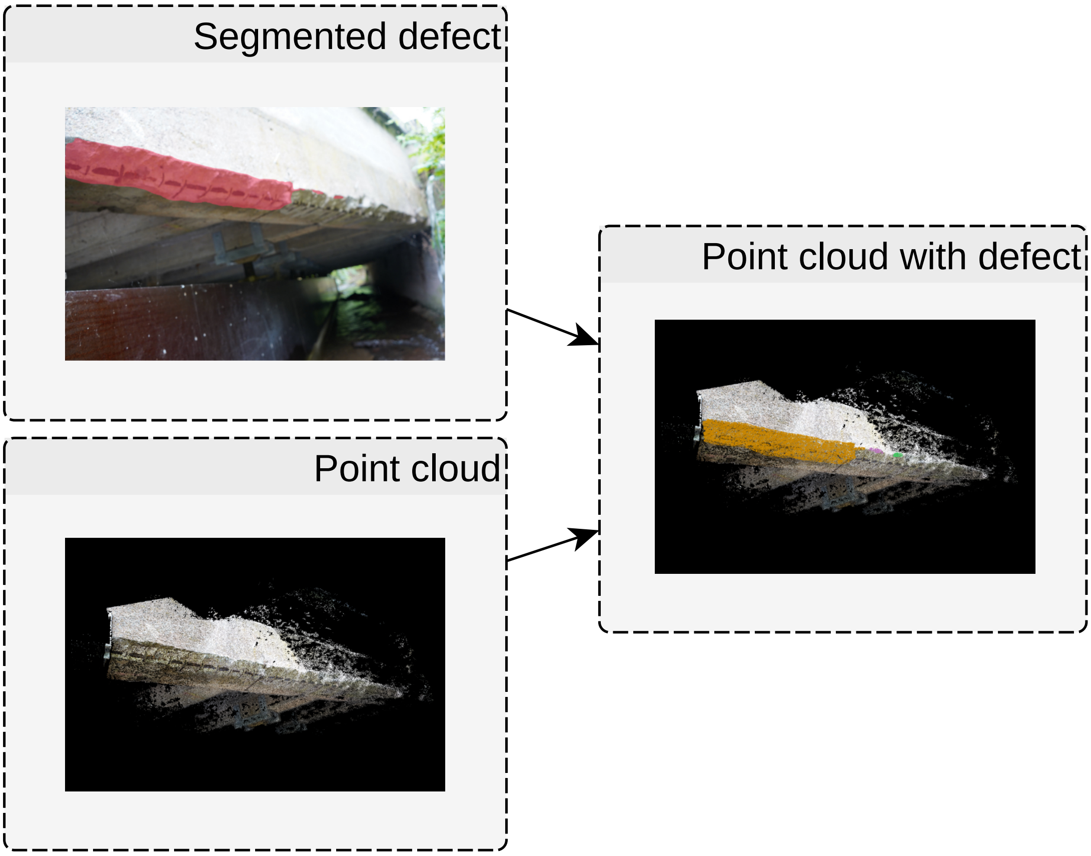


Figure 11: Generation of the damaged point cloud.

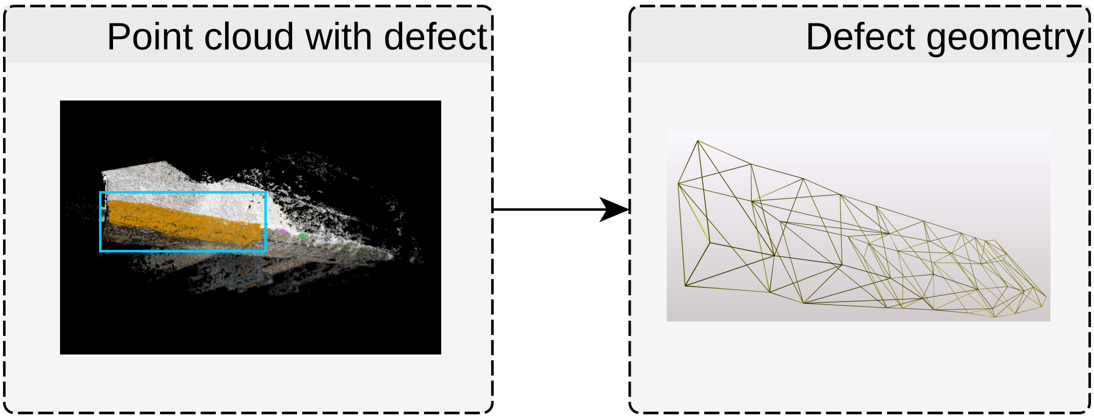


Figure 12: Generation of the defect geometry. The used points are bordered in blue.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

bridge model is generated by OpenSfM. The mesh of the segmented defect is generated by triangulating the vertices of the back-projected pixels in the segmented image into 3D world coordinates using its estimated depth map, the available intrinsic parameters of the camera, and its estimated pose calculated during the 3D reconstruction process via OpenSfM. The 3D reconstructed point cloud by SfM is downsampled and registered to the synthetic point cloud of the BIM model with Open3D [43] for the initial alignment and GoICP [32] for the refined global registration. Figure 13 shows the BIM model of the bridge and the point cloud.

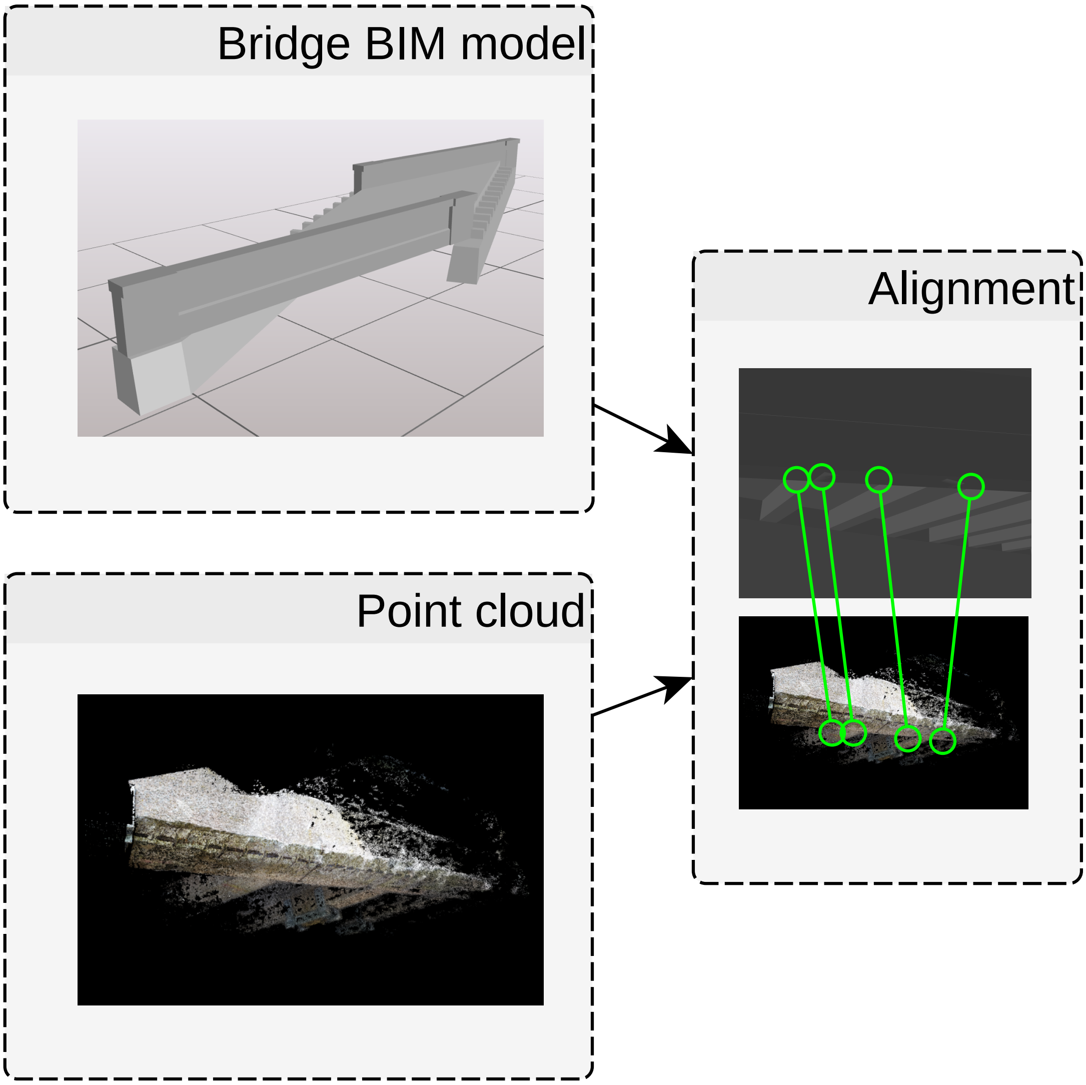


Figure 13: Alignment of the BIM model and the point cloid.

Next, the damage geometry, the bridge BIM model and the alignment are combined to build the as-damaged BIM model of the bridge. Figure 14 shows the BIM model on top, the geometry of the defect on the left and the alignment on the right. The damage geometry is used as geometry for an *IfcVoidingFeature* that is subtracted from the affected component. Figure 15 (a) depicts the cut plane of the cut view in Figure 15 (b). The spalling is subtracted from the primary beam and exposes the reinforcement as depicted by Figure 15 (c).

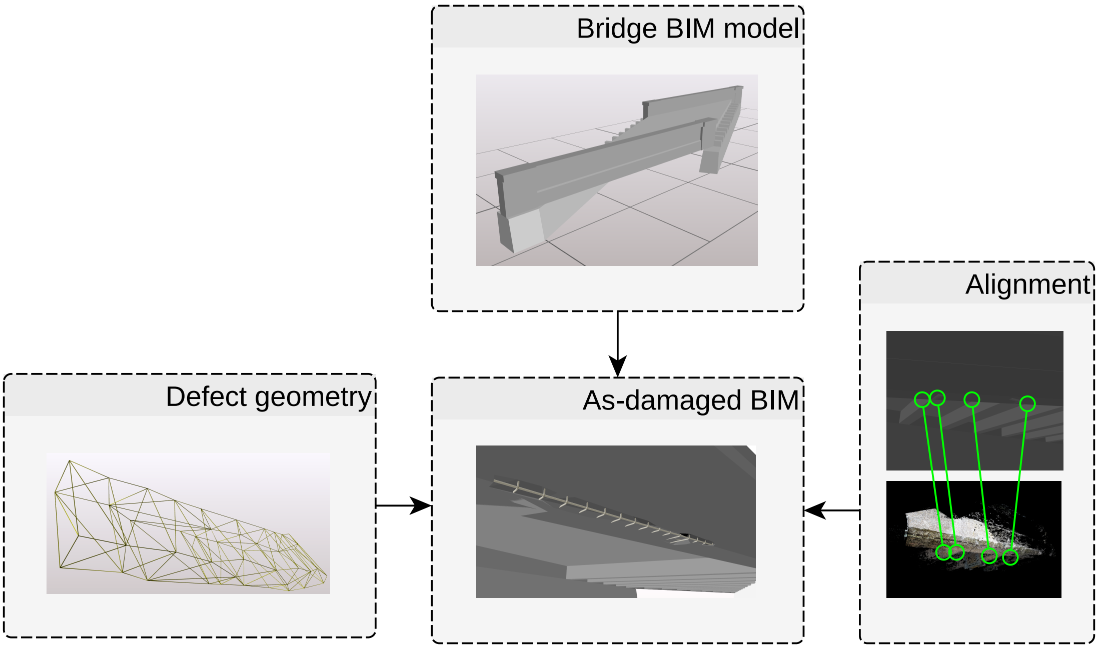


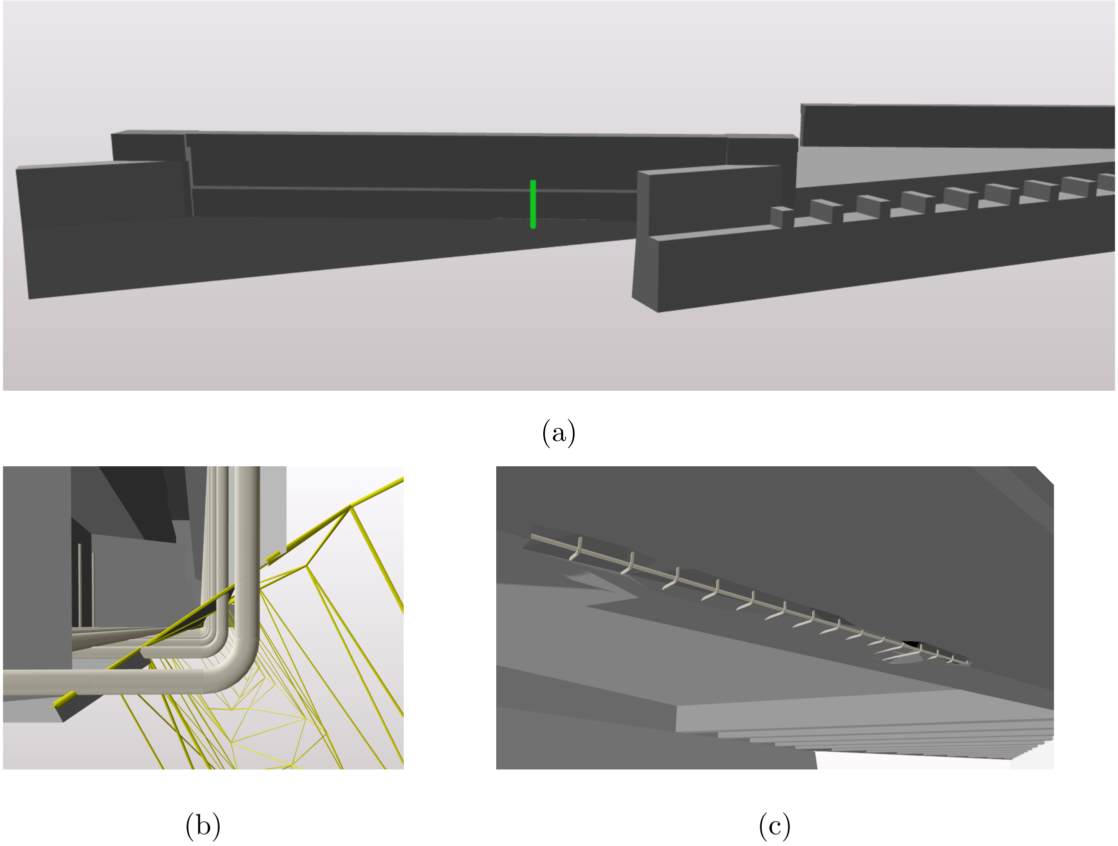
Figure 14: Generation of the as-damaged BIM model.

7.2 Adding Report Data

Analyzing the latest inspection report of the bridge, further 22 additional defects are identified. First, physical defects, such as cracks and spallings, may be integrated in the as-damaged model in the same way. Second, moisture penetration at the superstructure have to be included in the model. Moisture penetration cannot be represented by a subtraction. Instead, a simple proxy with a geometry for the extent and localization or photos may be used. Third, the report contains notes about rock pockets, plant covering, divergences from specifications, and missing elements. These defects may be included in the model using textual descriptions and photos as shown by Artus and Koch [29].

The report contains several numerical information, such as crack widths, the size of areas, and volume. Several defects in the report could be related to each other with a cause-effect relationship, for example, moisture penetration led to corrosion, which in turn led to spalling. Two defects are marked for noncompliance with existing norms. These defects have also a reference to the related norm or guideline. Last, some information, such as "water passes through the expansion joint at the support", may be stored

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1



(a)

(b)

(c)

Figure 15: Cut plane of the view marked in green (a). Cut view of the defect (b) and plan view of the defect (c)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

as text.

7.2.1 Lessons Learned

On the one hand, dense point clouds, which have been used for the geometry generation, include detailed information. On the other Hand, dense point clouds require huge amount of memory and increases the processing time. The density of the point cloud has to be defined based on the requirements of subsequent processing steps.

To provide comprehensive damage models, the BIM model, which is the basis of the DIM, has to be complete. This requires incorporating small or invisible objects, such as the reinforcement, drainage and bearings. The model used for the case study, such as misses drainage.

Modeling and visualizing missing components in the as-damaged model need further investigation. A missing component could be modeled by just removing the component from the model; however, this would include that an expected model is necessary for comparison. Another approach could be a flag at the component indicating that the component is missing. Using property sets in the IFC file may be a possible implementation for that. However, this does not provide a standard approach to model missing components.

The inspection report contains two defects that are related to joints, e.g., the joint tape is cracked. Modeling joints has gotten less attention until now. Not only, joints are parts of bridges and other buildings, but also joints are non-material objects and occur in huge amounts, which led to the practice to omit modeling joints. Future research has to clarify how damaged joints may be included in DIMs.

Some defects affect the location of the structure in relation to the environment or affect the surrounding itself, for instance settling, tilting, or subsidence of the slope. A proper visualization of these defects require a model of the surrounding ground. Different from that is the distortion of the superstructure of a bridge. This changes the geometry of numerous elements. This type of defect could also not be handled.

8 Summary

To ensure safety, traffic safety, and durability of bridges, inspections, analyses, and simulations are required. Hitherto, inspections and subsequent processes relied on paper based inspection and data exchange. The proposed framework outlines a complete digitized data acquisition and exchange of building and damage information. Based on an existing BIM model in the form of an IFC file and photos from the damaged structure, a BIM model with geometric damage data is generated. The model is implemented using an open standard, and hence, also processes, which require damage data and have not been respected yet, can make use of the model and included data. Some examples for subsequent processes are planning of non-destructive surveys, FEA, and bridge assessments.

The main contributions of this study are:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

- Development and implementation of an image based process for:
  - Defect geometry calculation
  - Defect alignment to BIM models
- Development of an inspection framework based on open file formats
- Providing a case study to test practical usability of the data acquisitions and storage

9 Discussion and Outlook

With novel technologies, numerous photos may be taken for bridge inspections. This study developed an open framework to acquire bridge data and facilitate inspection data for planning, analysis, and assessment. In the first step, the defect geometry is generated. A prior study with a small example used a point cloud of a single component to calculate and align defect geometry [44]. The present use case contained more photos of a bigger structure. This led to a denser point cloud with too many vertices for further processing, thus, splitting up this point cloud was necessary. Generally speaking, the density of the point cloud is an important factor in processing and storing damage data.

The proposed open framework is able to handle geometric, semantic, and graphical data of defects. Furthermore, the DIM can accommodate various damage types effectively, like spalling, corrosion, divergences from specifications, and moisture penetration. Problematic defects are plant covering, missing components, and rock pockets. Due to missing concepts for modeling joints, the framework is not able to visualize damaged joints properly. Especially, joint defects are important in case of inspections [45]. Modeling joints and joint defects should be addressed in future research.

With the information of planned reinforcement and existing defects, such as spalling, cracks, and moisture penetration, most of the planning of ultrasonic or impact-echo surveys can be done in office. However, survey results are included as reports only until now. The aim for future work is to incorporate the results of non-destructive testing in machine-readable formats.

Importing the geometry of the damaged component into structural analysis software helps the engineer to define structural models either using the 3D volume model as a draft for the structural framework or mesh the 3D model for volumetric FEA. In case of meshing the geometry, additional handwork of the engineer is necessary to resolve problems in the element meshes, for example, singularities.

Finally, the visualization for assessment provides the engineer with available semantic, geometric, and graphical data. The engineer may observe component groups or the entire structure to gain an overview of problematic defects or components. However, decisions about the condition rating of the bridge lies within the responsibility of the engineer. This final decision requires deep knowledge of the existing norms and guidelines and experience in impacts of defects. The Framework aims to digitizes the workflow of data acquisition and exchange but leaves decisions to the engineer.

10 Acknowledgements

We thank the “Thüringer Landesamt für Bau und Verkehr” [Thuringian Department for Construction and Transport] to provide us with data and helped us with their practical expertise.

References

[1] Rainer Schach et al. “Lebenszykluskosten von Brücken”. In: *Bauingenieur* 81 (2006), pp. 7–14.

[2] Lars-Marten Nagel et al. *Wettlauf gegen den Verfall*. Ed. by Die Welt. 2016. URL: `http://www.welt.de/politik/interaktiv/bruecken/deutschlands-bruecken-wettlauf-gegen-den-verfall.html`.

[3] Rafael Sacks et al. *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*. Third edition. Hoboken, New Jersey: Wiley, 2018. ISBN: 9781119287544.

[4] André Borrmann et al., eds. *Building Information Modeling: Technology Foundations and Industry Practice*. 2nd ed. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-92861-6. DOI: `10.1007/978-3-319-92862-3`. URL: `https://doi.org/10.1007/978-3-319-92862-3`.

[5] Rafael Sacks et al. “SeeBridge as next generation bridge inspection: Overview, Information Delivery Manual and Model View Definition”. In: *Automation in Construction* 90 (2018), pp. 134–145. ISSN: 09265805. DOI: `10.1016/j.autcon.2018.02.033`.

[6] Eric Lebegue. *IFC-Bridge & IFC for Roads*. München, 8.10.2013. URL: `http://iug.buildingsmart.org/resources/itm-and-iug-meetings-2013-munich/infra-room/ifc-bridge-ifc-for-roads`.

[7] Rebekka Volk, Julian Stengel, and Frank Schultmann. “Building Information Modeling (BIM) for existing buildings — Literature review and future needs”. In: *Automation in Construction* 38 (2014), pp. 109–127. ISSN: 09265805. DOI: `10.1016/j.autcon.2013.10.023`.

[8] L. Barazzetti et al. “BIM from Laser Scans... not just for Buildings: Nurbs-Based Parametric Modeling of a Medieval Bridge”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-5 (2016), pp. 51–56. ISSN: 2194-9050. DOI: `10.5194/isprs-annals-III-5-51-2016`.

[9] Christopher Kropp, Christian Koch, and Markus König. “Interior construction state recognition with 4D BIM registered image sequences”. In: *Automation in Construction* 86 (2018), pp. 11–32. ISSN: 09265805. DOI: `10.1016/j.autcon.2017.10.027`.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

[10] George Hearn. *Bridge Inspection Practices*. 2007. DOI: 10.13140/RG.2.1.5179.5604. URL: https://www.researchgate.net/publication/298792603_Bridge_Inspection_Practices.

[11] Philipp Hüthwohl et al. "Integrating RC Bridge Defect Information into BIM Models". In: *Journal of Computing in Civil Engineering* 32 (2018), pp. 1–14. DOI: 10.1061/(ASCE)CP.1943-5487.0000744.

[12] Al-Hakam Hamdan and J. Raimar Scherer. "Modular Concatenation of Reference Damage Patterns". In: *EWork and EBusiness in Architecture, Engineering and Construction*. Ed. by Jan Karlshoj and Raimar Scherer. Milton: Chapman and Hall/CRC, 2018. ISBN: 9781138584136. URL: https://www.researchgate.net/publication/328074732_Modular_Concatenation_of_Reference_Damage_Patterns.

[13] F. Tanaka et al. "Bridge Information Modeling based on IFC for supporting maintenance management of existing bridges". In: *ICCCBE 2018 Conference Proceedings*. Ed. by Kristo Mela, Sami Pajunen, and Ville Raasakka. 2018, pp. 778–785. ISBN: 978-951-758-632-0.

[14] Stephanie German, Ioannis Brilakis, and Reginald DesRoches. "Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments". In: *Advanced Engineering Informatics* 26.4 (2012), pp. 846–858. ISSN: 14740346. DOI: 10.1016/j.aei.2012.06.005.

[15] Stephanie German Paal, Ioannis Brilakis, and Reginald DesRoches. "Automated measurement of concrete spalling through reinforcement detection". In: (2014).

[16] Thikra Dawood, Zhenhua Zhu, and Tarek Zayed. "Machine vision-based model for spalling detection and quantification in subway networks". In: *Automation in Construction* 81 (2017), pp. 149–160. ISSN: 09265805. DOI: 10.1016/j.autcon.2017.06.008.

[17] Hangbin Wu et al. "Concrete Spalling Detection for Metro Tunnel from Point Cloud Based on Roughness Descriptor". In: *Journal of Sensors* 2019 (2019), pp. 1–12. ISSN: 1687-725X. DOI: 10.1155/2019/8574750.

[18] Nhat-Duc Hoang, Quoc-Lam Nguyen, and Xuan-Linh Tran. "Automatic Detection of Concrete Spalling Using Piecewise Linear Stochastic Gradient Descent Logistic Regression and Image Texture Analysis". In: *Complexity* 2019 (2019), pp. 1–14. ISSN: 1076-2787. DOI: 10.1155/2019/5910625.

[19] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017. DOI: 10.1109/iccv.2017.322.

[20] P. Borin and F. Cavazzini. "Condition Assessment of RC Bridges. Integrating Machine Learning, Photogrammetry and BIM". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W15 (2019), pp. 201–208. DOI: 10.5194/isprs-archives-XLII-2-W15-201-2019. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W15/201/2019/.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

[21] Silvana Bruno, Mariella de Fino, and Fabio Fatiguso. "Historic Building Information Modelling: Performance assessment for diagnosis-aided information modelling and management". In: *Automation in Construction* 86 (2018), pp. 256–276. ISSN: 09265805. DOI: 10.1016/j.autcon.2017.11.009.

[22] Matthew M. Torok, Mani Golparvar-Fard, and Kevin B. Kochersberger. "Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment". In: *Journal of Computing in Civil Engineering* 28.5 (2014), A4014004. ISSN: 0887-3801. DOI: 10.1061/(ASCE)CP.1943-5487.0000334.

[23] Brodie Chan et al. "Defining a conceptual framework for the integration of modelling and advanced imaging for improving the reliability and efficiency of bridge assessments". In: *Journal of Civil Structural Health Monitoring* 6.4 (2016), pp. 703–714. ISSN: 2190-5452. DOI: 10.1007/s13349-016-0191-6.

[24] Brendan Michael McGuire. "Using building information modeling to track and assess the structural condition of bridges". Master thesis. Colorado: Colorado State University, 2014.

[25] Brendan McGuire et al. "Bridge Information Modeling for Inspection and Evaluation". In: *Journal of Bridge Engineering* 21.4 (2016), p. 04015076. ISSN: 1084-0702. DOI: 10.1061/(ASCE)BE.1943-5592.0000850.

[26] Al-Hakam Hamdan et al. "A semantic modeling approach for the automated detection and interpretation of structural damage". In: *Automation in Construction* 128 (2021), p. 103739. ISSN: 09265805. DOI: 10.1016/j.autcon.2021.103739.

[27] Dušan Isailović et al. "Bridge damage: Detection, IFC-based semantic enrichment and visualization". In: *Automation in Construction* 112 (2020), p. 103088. ISSN: 09265805. DOI: 10.1016/j.autcon.2020.103088.

[28] Richard E. Lindenberg and Jonathan C. McGormley. "Visualizing Bridge Inspection with 2D+1 Software". In: *Computing in Civil Engineering*. Ed. by Ioannis Brilakis, SangHyun Lee, and Burcin Becerik-Gerber. Reston, VA: American Society of Civil Engineers, 2013, pp. 857–864. ISBN: 978-0-7844-1302-9. DOI: 10.1061/9780784413029.107.

[29] Mathias Artus and Christian Koch. "Modeling Geometry and Semantics of Physical Damages using IFC". In: *EG-ICE 2020 Workshop on Intelligent Computing in Engineering*. Ed. by Lucian Constantin Ungureanu and Timo Hartmann. 2020, pp. 144–153. ISBN: 978-3-7983-3156-3.

[30] Mathias Artus and Christian Koch. "Modeling Physical Damages Using the Industry Foundation Classes – A Software Evaluation". In: *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*. Ed. by Eduardo Toledo Santos and Sergio Scheer. Vol. 98. Lecture Notes in Civil Engineering. Cham: Springer International Publishing, 2021, pp. 507–518. ISBN: 978-3-030-51294-1. DOI: 10.1007/978-3-030-51295-8{\textunderscore}36.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

[31] Jiaolong Yang, Hongdong Li, and Yunde Jia. "Go-ICP: Solving 3D Registration Efficiently and Globally Optimally". In: *2013 IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 1457–1464. ISBN: 978-1-4799-2840-8. DOI: `10.1109/ICCV.2013.184`.

[32] Jiaolong Yang et al. "Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration". In: *IEEE transactions on pattern analysis and machine intelligence* 38.11 (2016), pp. 2241–2254. DOI: `10.1109/TPAMI.2015.2513405`.

[33] Gary R. Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. 1. ed., [Nachdr.] Software that sees. Beijing: O'Reilly, 2011. ISBN: 9780596516130.

[34] Mapillary AB. *OpenSfM*. 2014. URL: `https://opensfm.org`.

[35] Alexey A. Shvets et al. "Automatic Instrument Segmentation in Robot-Assisted Surgery using Deep Learning". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 624–628. ISBN: 978-1-5386-6805-4. DOI: `10.1109/ICMLA.2018.00100`.

[36] Autodesk. *Revit*. Ed. by Autodesk. 2019. URL: `https://www.autodesk.de/products/revit/overview`.

[37] Blender Foundation. *Blender*. 1995. URL: `https://www.blender.org/`.

[38] Daniel Girardeau-Montaut. *CloudCompare*. 2015. URL: `http://www.cloudcompare.org/`.

[39] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. ISSN: 00295981. DOI: `10.1002/nme.2579`.

[40] Liang Yang et al. "A robotic system towards concrete structure spalling and crack database". In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 1276–1281. ISBN: 978-1-5386-3742-5. DOI: `10.1109/ROBIO.2017.8324593`.

[41] Liang Yang et al. "Semantic Metric 3D Reconstruction for Concrete Inspection". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018, pp. 1624–16248. ISBN: 978-1-5386-6100-0. DOI: `10.1109/CVPRW.2018.00204`.

[42] buildingSMART International Ltd. *IFC4 Add2: Addendum 2 [official]*. 2016. URL: `https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/`.

[43] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. *Open3D: A Modern Library for 3D Data Processing*. URL: `http://arxiv.org/pdf/1801.09847v1`.

[44] Mathias Artus, Mohamed Said Helmy Alabassy, and Christian Koch. "IFC based Framework for Generating, Modeling and Visualizing Spalling Defect Geometries". In: *EG-ICE 2021 Workshop on Intelligent Computing in Engineering*. Ed. by Jimmy Abualdenien et al. Universitätsverlag der TU Berlin, 2021.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 January 2022 doi:10.20944/preprints202201.0094.v1

[45] Mathias Artus and Christian Koch. "State of the art in damage information modeling for RC bridges – A literature review". In: *Advanced Engineering Informatics* 46 (2020), p. 101171. ISSN: 14740346. DOI: 10.1016/j.aei.2020.101171.