3D Object Classification using Geometric Features and Pairwise Relationships

Ling Ma*

School of Art, Design and Architecture, University of Huddersfield, Huddersfield, UK

&

Rafael Sacks, Uri Kattel, Tanya Bloch

Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa, Israel

Abstract: *Object classification is a key differentiator of Building Information Modeling (BIM) from 3D CAD. Incorrect object classification impedes the full exploitation of BIM models. Models prepared using domain-specific software cannot ensure correct object classification when transferred to other domains, and research on reconstruction of BIM models using spatial survey has not proved a full capability to classify objects. This research proposed an integrated approach to object classification that applied domain experts' knowledge of shape features and pairwise relationships of 3D objects to effectively classify objects using a tailored matching algorithm. Among its contributions: the algorithms implemented for shape and spatial feature identification could process various complex 3D geometry; the method devised for compilation of the knowledge base considered both rigor and confidence of the inference; the algorithm for matching provides mathematical measurement of the object classification results. The integrated approach has been applied to classify 3D bridge objects in two models: a model prepared using incorrect object types and a model manually reconstructed using point cloud data. All these objects were successfully classified.*

1 INTRODUCTION

Object classification is central to Building Information Modeling (BIM). Misclassification impedes the full exploitation of BIM models in applications such as building code-compliance checking, energy analysis and cost estimation, because none of these can be done if the objects' types are not defined.

Classification errors arise when BIM authoring tools do not have the full range of building objects needed for a design. Design tools are domain or discipline specific, and modelers sometimes improvise. For example, Belsky et al. (2016) noted a case where a precast joint filler strip was modeled as a structural column because no filler strip object was available to the modeler in the authoring tool.. As a result, the model would include one more column, which conveys wrong design intent.

Classification information may also be lost or missing when BIM models are exchanged. When the models are exported and loaded between different BIM authoring tools, they are often used as the reference geometry only due to the interoperability problem. The users have to spend much time on manually classifying the objects to be able to execute BIM-enabled applications like structural analysis and energy analysis.

Classification information may also be missing when the model is reconstructed from 3D spatial survey data. Recent advances in acquiring and processing point cloud data of existing buildings, using laser scanning (Kim et al., 2013; Bosche et al., 2015; Diaz-Vilarino et al., 2015) and/or video/photogrammetry (Arias et al., 2007; Fathi and Brilakis, 2016), hold the promise of compiling BIM models automatically or semi-automatically. Object identification in the spatial survey data includes two steps: 1) 3D reconstruction and 2) object classification. Many researchers have demonstrated the capability to create 3D solid geometry from point cloud data (PCD) (Walsh et al., 2013; Jung et al., 2016), yet none of them have proven a full capability to classify all the objects in a large-scale scenario.

This research aimed at developing an approach to facilitating classification of 3D solid objects in a model. The approach could be applied in two scenarios:

1) A BIM model has been prepared or exchanged, but the types of the objects are not correctly expressed.
2) A complete 3D solid model has been reconstructed using spatial survey data, but the types of the objects are not identified.

The approach includes three major technical components:

1) Representation of domain experts' knowledge of object classification using the features
2) Identification of shape and spatial features of 3D objects
3) Classification of objects based on mathematical computation

This research makes the following contributions:

1) The limitations of the forward chaining rule-based object classification approach were identified (section 2.4).
2) A matrix representation was devised to express domain experts' knowledge of object classification (section 3.1).
3) State-of-the-art algorithms were implemented for computing shape features and spatial relationships of 3D objects (section 3.2).
4) An algorithm was developed for classifying objects based on a tailored similarity computation (section 3.3).

The paper begins with a background review of several object classification approaches. The following sections illustrate the technical components of the new approach, presenting two applications. The conclusions section discusses the advantages and limitations.

2 BACKGROUND

The proposed approach is based on the observation that people can classify building objects by recognizing their shape features and their spatial relationships with other objects. It builds on the research on expert systems (Clancey, 1983), rule-checking systems for BIM (Pauwels et al., 2011), and semantic enrichment systems for BIM (Belsky, Sacks and Brilakis, 2016). For example, a beam is usually a horizontal object with a longitudinally extruded cross-section (shape feature) and is perpendicular to its supporting columns (spatial relationship). Domain experts apply such knowledge to recognize objects, and computers can be coded to implement reasoning systems using similar logical constructs to identify objects in spatial data.

Many researchers have used shape features (described in section 2.1 below) or spatial relationships (section **2.2**) to identify objects. Forward chaining rules (section **2.3**) have been applied to encapsulate domain experts' knowledge of object classification. However, rule-based inference lacked rigor and failed to leverage the inexact human knowledge (section **2.4**).

2.1 Object identification based on shape features

Some building objects can be recognized based on the features of their lower level geometry primitives (e.g., face, polyline, etc.). For example, columns could be identified based on the facts that they have long, vertical boundary edges and their surfaces have uniform textures (Zhu and Brilakis, 2010; Paal et al., 2015); bridge decks, house roofs, ceilings, walls and floors could be identified based on their intrinsically flat surfaces (Zhang et al., 2015; Valero et al., 2016); pipes could be identified based on their rough surfaces and curvature (Dimitrov and Golparvar-Fard, 2015; Czerniawski et al., 2016). However, none of these has proved the capability of identifying complex solid objects with a variety of shapes in a scene even though the solid geometries could be reconstructed.

2.2 Object identification based on spatial relationships

Wang et al. (2015) used topological relations (e.g., the roof is above the walls and adjacent to exterior walls) together with surface features (such as size, normal direction, with/without openings) to distinguish walls, doors, windows and roofs. Given the intrinsic characteristics of equipment in a chemical plant, Son et al. (2015) realized that connectivity information is crucial for classification of such objects. They established connectivity graphs of typical equipment based on a 3D CAD database, built a similar graph of actual objects of a model to be classified, and finally identified the object types based on graph matching. Similarly, Anand et al. (2013) built connection graphs for identification of office objects. Ruiz-Sarmiento et al. (2015) encoded more non-commutative pairwise relations in their algorithm for identifying a floor, a wall, a table, a chair and a computer screen in a small scene.

By matching the spatial features between an 'as-designed' BIM model and the 'as-is' PCD, Tang et al. (2016) and Kalasapudi et al. (2017) were able to identify the cylindrical building elements in the PCD that corresponded to the 'as-designed' BIM objects for object change detection and variation analysis. Similarly, based on the fact that the spatial relationships between structural elements in an unreinforced masonry wall frame do not change after undergoing earthquake damage, Zeibak-Shini et al. (2016) were able to identify the 'as-damaged' structural objects that corresponded to the 'as-designed' BIM objects. These three examples proved the feasibility of using pairwise relationships to recognize objects that are of interest, but they assumed the existence of a BIM model as a matching template. They were used in application scenarios different from this research.

2.3 Forward chaining rules for knowledge representation

Logic rule-based inference systems (Adeli, 2003) have been used for machine control (Jardon et al., 2012) and knowledge-based scheduling (Kao and Adeli, 2002). Inference rules encapsulate human knowledge to automate the computing process for these purposes. Similarly, rule-based inferencing can also combine the shape features and spatial relationships of domain specific 3D objects to interpret a domain expert's object recognition knowledge. Belsky, Sacks and Brilakis (2016) used a customized rule-based semantic enrichment engine, named SeeBIM, to parse an IFC building model file, compute the geometric features and topological relations of the 3D objects, and infer the types of the objects based on user-defined rules (e.g., if object 1 and 2 are two adjacent precast concrete walls, and object 3 is located within the clearance between object 1 and 2, then object 3 is a sealing strip). The end user could leverage domain experts' knowledge of object classification, which was encapsulated in forward chaining inference rules.

2.4 Limitations of rule-based inference approach

Forward chaining rules can become very complex and error-prone for object classification of a complex structure, because they may be interdependent (identification of some objects depends on the prior identification of other related objects) and conflict conditions among them can result in infinite loops in the computation. Consider the following two rules:

1) IF an object has features 'a' and 'b' THEN it is X
2) IF an object has features 'b' and 'c' THEN it is Y

As a result, if, in an actual model, one object has all the three features, i.e., 'a', 'b' and 'c', then it would be alternatively recognized, first as type X and then as type Y, in an infinite loop. This occurs because the first rule does not strictly specify whether objects of type X have or do not have feature 'c', i.e., some objects of type X may have feature 'c', but some may not. Similarly, according to rule 2, some objects of type Y have feature 'a', but some do not. Applications in manufacturing have shown that the major problem of rule-based methods is to ensure the validity of the rules that are compiled heuristically (Babic et al., 2008).

Sacks et al. (2017) proposed a more rigorous approach to compiling object classification rules using a tailored string comparison algorithm. The algorithm would automatically discard the two ambiguous rules above. But this did not take into account the fact that human knowledge is often imprecise or ambiguous. Rule-based inference will fail to classify objects that violate any of the assumptions concerning their features. For example, the inference will not be able to give any result if one object only has feature 'a' because both conditions are not satisfied. However, if this object has to be assigned a type, it is more likely to be type X than type Y, because all the objects of type X have feature 'a', while only some objects of type Y may have it. In conclusion, the rule-based approach excludes the possibility of inference based on imprecise knowledge (Steffens, 2006).

As a result, a new approach is required. It should provide a rigorous way to encapsulate domain specific knowledge of object classification, and it should allow inference of object classification based on inexact human knowledge.

3 A NEW APPROACH TO OBJECT CLASSIFICATION

Figure 1 shows the three steps of the proposed approach:

Step 1: Interpret the domain experts' knowledge of object classification using a computer-readable data structure (knowledge base). The knowledge base encapsulates knowledge about objects' shape features and their spatial context.

Step 2: Compute object shape features and spatial relationships using state-of-the-art algorithms.

Step 3: Automate the object classification by matching the features of the models and the features used in the knowledge base. This is the major difference from the 'if-then' rule-based approach.

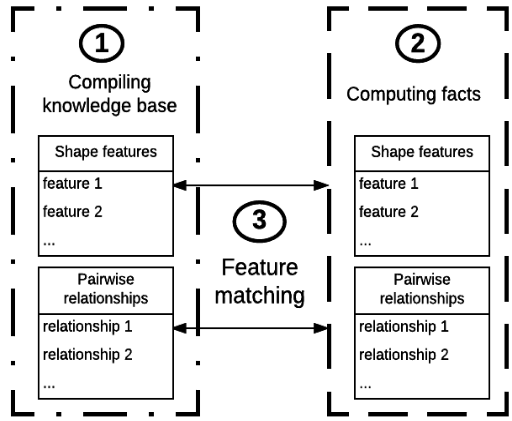


Figure 1 Overview of the approach

3.1 Compilation of object classification knowledge matrices

The implicit object classification knowledge was made explicit in the form of matrices. Figure 2 shows the workflow. Section 3.1.1 explains steps 1, 2 and 3 for compiling the knowledge matrices. Section 3.1.2 explains step 4, how to evaluate the comprehensiveness of the object classification knowledge.

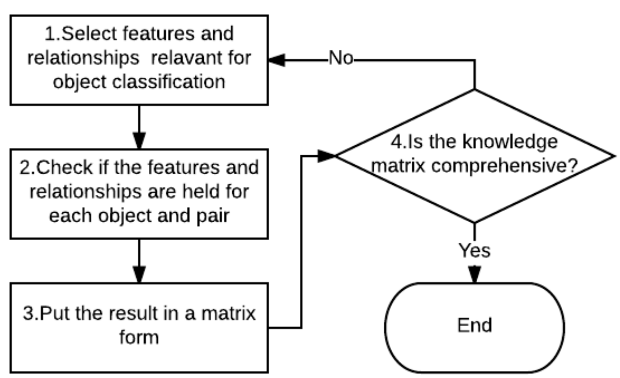


Figure 2 Compiling a knowledge matrix

3.1.1 Compilation of knowledge matrices

Objects in different domains could be classified based on different domain specific knowledge. In order to 'teach' a computer to recognize the objects, such knowledge should be interpreted in a machine-readable format. A matrix representation was proposed for this purpose.

Table 1 is a synthetic example of a knowledge matrix of objects' singular shape features. The column headers are singular shape features that could be chosen from section 3.2.1 or others. The row headers are object types for classifications. Domain experts are asked to express their knowledge by filling value in the cells. '1' signifies that this type of object always has this feature; '-1' signifies that it never has this feature; '0' signifies that it may or may not have this feature.

Table 1

An example of shape feature knowledge matrix

	Feature1	Feature2	Feature3
T1	1	1	1
T2	1	1	0
T3	0	-1	-1
T4	-1	1	1

Table 2 is a synthetic example of a knowledge matrix of objects' pairwise relationships. The column headers are pairwise relationships that could be chosen from section 3.2.2 or others. The row headers are pairs of object types. Again, domain experts are asked to express their knowledge by filling values in the cells. '1' signifies that two objects of these two types always have this relationship; '-1' signifies that they never have this relationship; '0' signifies that they may or may not have. Note that two objects of the same type are also evaluated in the matrix, e.g., the first and the last row of Table 2. In addition, the same pairs of object types appear twice in the table (e.g., [T1, T2] and [T2, T1]) because some relationships are not commutative (e.g., where they are in contact with one another, objects of the first type are always above objects of the second type) and the sequence of the objects in the pair matters. In addition, the relationships also apply to two objects of the same types, so pairs like [T1,T2] and [T2,T2] exist in the table.

Table 2

An example of pairwise relationship knowledge matrix

	Relation1	Relation2	Relation3	Relation4
T1, T1	1	0	0	-1
T1, T2	-1	-1	0	1
T1, T3	0	1	0	-1
T1, T4	1	0	1	1
T2, T1	1	0	0	-1
T2, T2	1	-1	0	0
T2, T3	-1	0	1	-1
T2, T4	-1	-1	0	0
T3, T1	1	0	1	-1
T3, T2	-1	1	1	1
T3, T3	-1	1	0	1
T3, T4	1	1	-1	1
T4, T1	1	0	0	1
T4, T2	0	1	0	1
T4, T3	0	-1	0	-1
T4, T4	-1	0	0	0

Given a specific scene, the number of object types are limited. For example, an AASHTO girder bridge mainly includes girders, columns, shear keys, abutment, and bearings. However, the knowledge matrices may vary when the domain experts choose different features and relationships for these objects. Therefore, how to evaluate the comprehensiveness of the object classification knowledge is a problem. We deal with this in the following section.

3.1.2 Evaluation of knowledge matrices

The classification knowledge is comprehensive when the matrices have enough information to distinguish each object type from the others. Each row in Table 1 is referred to as a feature vector $\mathbf{V}_i$ of object Type i (Ti in the table). $\mathbf{V}_i$ is a three-dimensional vector because three features are considered. For example, the feature vectors of object Type1 and Type2 are shown in Figure 3.

Value '0' is ambiguous because it includes the conditions represented by either '-1' or '1'. Two types of objects can be clearly distinguished only when the non-zero coordinates of their feature vectors are not entirely the same. Namely, the projection of $\mathbf{V}_i$ on to $\mathbf{V}_j$ is not the same as $\mathbf{V}_j$ and the projection of $\mathbf{V}_j$ on to $\mathbf{V}_i$ is not the same as $\mathbf{V}_i$. As is shown in Figure 3, the projection of $\mathbf{V}_2$ on to $\mathbf{V}_1$ does not equal to $\mathbf{V}_1$, but the projection of $\mathbf{V}_1$ on to $\mathbf{V}_2$ equals to $\mathbf{V}_2$, so the object Type1 and Type2 are not clearly distinguished in the knowledge matrix.

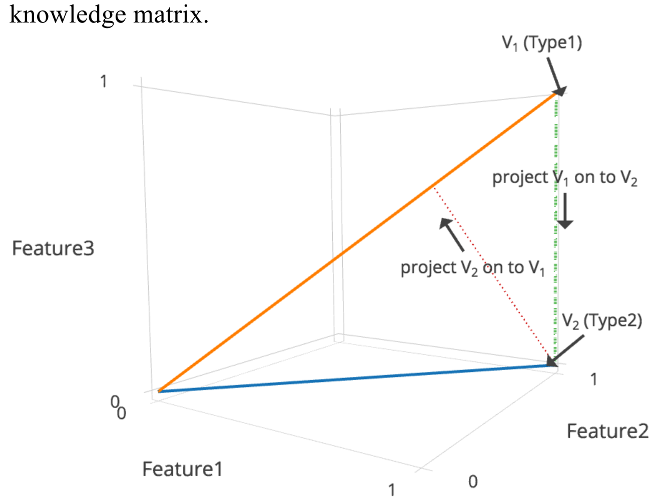


Figure 3 Feature vectors of Type1 and Type2 in Table 1

The projection of $\mathbf{V}_i$ on to $\mathbf{V}_j$ is computed using Eq. (1), and we only need to evaluate if C_{ij} equals to 1 or C_{ji} equals to 1.

$$\mathbf{P}_{\mathbf{V}_i\mathbf{V}_j} = \frac{\mathbf{V}_i \cdot \mathbf{V}_j}{\mathbf{V}_j \cdot \mathbf{V}_j}\mathbf{V}_j = C_{\mathbf{V}_i\mathbf{V}_j}\mathbf{V}_j \tag{1}$$

Based on the data in Table 1, $C_{\mathbf{V}_i\mathbf{V}_j}$ of all the feature vectors are computed and shown in Table 3. $\mathbf{V}_3$ is unique because $C_{\mathbf{V}_3\mathbf{V}_j} \neq 1$ and $C_{\mathbf{V}_i\mathbf{V}_3} \neq 1$ where $i \neq 3$ and $j \neq 3$. Namely, object Type3 is unique and its features are different from all others'. By the same token, object Type4 is also unique. But object Type1 and Type2 are not. The comprehensiveness of the knowledge matrix corresponds to the percentage of unique object types in the whole set. If the percentage is less than 100, the domain experts should identify more features that can make all the object types unique, and these may be either singular or pairwise features.

Table 3

$C_{\mathbf{V}_i\mathbf{V}_j}$ of all the feature vectors in Table 1

	$\mathbf{V}_1$	$\mathbf{V}_2$	$\mathbf{V}_3$	$\mathbf{V}_4$
$\mathbf{V}_1$	1.00	0.67	-0.67	0.33
$\mathbf{V}_2$	1.00	1.00	-0.50	0.00
$\mathbf{V}_3$	-1.00	-0.50	1.00	-1.00
$\mathbf{V}_4$	0.33	0.00	-0.67	1.00

The matrix of pairwise relationships in Table 2 could also be used to evaluate the uniqueness of object types. Each row in Table 2 is referred to as a vector $\mathbf{R}(\mathrm{i},\mathrm{j})$ representing all the relationships between a type i object and a type j object. $\mathbf{R}(\mathrm{i},\mathrm{j})$ is a four-dimensional vector because four relationships are considered. Similar to the shape feature matrix and Eq. (1), the uniqueness of the object pairs' relationships can also be evaluated based on $C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})}$, which is computed as follows:

$$C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})} = \frac{\mathbf{R}(\mathrm{i},\mathrm{j}) \cdot \mathbf{R}(\mathrm{m},\mathrm{n})}{\mathbf{R}(\mathrm{m},\mathrm{n}) \cdot \mathbf{R}(\mathrm{m},\mathrm{n})} \tag{2}$$

Based on the data in Table 2, $C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})}$ of all the combinations are computed and shown in Table 4. The relationships implied by $\mathbf{R}(\mathrm{i},\mathrm{j})$ and the relationships implied by $\mathbf{R}(\mathrm{m},\mathrm{n})$ are different when $C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})} \neq 1$ and $C_{\mathbf{R}(\mathrm{m,n})\,\mathbf{R}(\mathrm{i,j})} \neq 1$ ($i \neq m$ or $j \neq n$). According to Table 4, the relationships between a Type1 object and a Type2 object are unique because the cells in the row with header (T1,T3) and the cells in the column with header (T1,T3) have only one value of 1.00. This implies that if this set of relationships are found to hold for any pair of objects, which means that the identity of both objects can be determined. By the same token, the relationships implied by $\mathbf{R}(1{,}3)$, $\mathbf{R}(1{,}4)$, $\mathbf{R}(2{,}2)$ and $\mathbf{R}(2{,}3)$ are unique, thereby, all the objects could be identified since each of them appears in at least one of the relationship vectors.

In conclusion, the knowledge matrices are new representations for interpreting object classification knowledge. They can be rigorously evaluated using the proposed algorithm. Pairwise relationships are more likely to be able to identify an object than the singular shape features, for the following two reasons:

1) For t different object types, $C_{\mathbf{V}_i\mathbf{V}_j}$ is a t dimensional vector, while $C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})}$ is a t^2 dimensional vector, which is more likely to be unique.
2) For t different object types, each type is involved in $2t-1$ possible pairs. If any one of these pairs is unique, the object could be identified.

3.2 Computation of shape features and spatial contexts

The following two subsections outline the singular shape features and the pairwise spatial relationships used for object classification in this approach.

Table 4

$C_{\mathbf{R}(\mathrm{i,j})\,\mathbf{R}(\mathrm{m,n})}$ of all the feature vectors in Table 2

	T1,T1	T1,T2	T1,T3	T1,T4	T2,T1	T2,T2	T2,T3	T2,T4	T3,T1	T3,T2	T3,T3	T3,T4	T4,T1	T4,T2	T4,T3	T4,T4
T1,T1	1.00	-1.00	0.50	0.00	1.00	0.50	0.00	-0.50	1.00	-1.00	-1.00	0.00	0.00	-0.50	0.50	-0.50
T1,T2	-0.67	1.00	-0.67	0.00	-0.67	0.00	0.00	0.67	-0.67	0.33	0.33	-0.33	0.00	0.00	0.00	0.33
T1,T3	0.50	-1.00	1.00	-0.50	0.50	-0.50	0.50	-0.50	0.50	0.00	0.00	0.00	-0.50	0.00	0.00	0.00
T1,T4	0.00	0.00	-0.33	1.00	0.00	0.33	-0.33	-0.33	0.33	0.33	0.00	0.33	0.67	0.33	-0.33	-0.33
T2,T1	1.00	-1.00	0.50	0.00	1.00	0.50	0.00	-0.50	1.00	-1.00	-1.00	0.00	0.00	-0.50	0.50	-0.50
T2,T2	0.50	0.00	-0.50	0.50	0.50	1.00	-0.50	0.00	0.50	-1.00	-1.00	0.00	0.50	-0.50	0.50	-0.50
T2,T3	0.00	0.00	0.33	-0.33	0.00	-0.33	1.00	0.33	0.33	0.33	0.00	-1.00	-0.67	-0.33	0.33	0.33
T2,T4	-0.50	1.00	-0.50	-0.50	-0.50	0.00	0.50	1.00	-0.50	0.00	0.00	-1.00	-0.50	-0.50	0.50	0.50
T3,T1	0.67	-0.67	0.33	0.33	0.67	0.33	0.33	-0.33	1.00	-0.33	-0.67	-0.33	0.00	-0.33	0.33	-0.33
T3,T2	-0.50	0.25	0.00	0.25	-0.50	-0.50	0.25	0.00	-0.25	1.00	0.75	0.00	0.00	0.50	-0.50	0.25
T3,T3	-0.67	0.33	0.00	0.00	-0.67	-0.67	0.00	0.00	-0.67	1.00	1.00	0.33	0.00	0.67	-0.67	0.33
T3,T4	0.00	-0.25	0.00	0.25	0.00	0.00	-0.75	-0.50	-0.25	0.00	0.25	1.00	0.50	0.50	-0.50	-0.25
T4,T1	0.00	0.00	-0.50	1.00	0.00	0.50	-1.00	-0.50	0.00	0.00	0.00	1.00	1.00	0.50	-0.50	-0.50
T4,T2	-0.50	0.00	0.00	0.50	-0.50	-0.50	-0.50	-0.50	-0.50	1.00	1.00	1.00	0.50	1.00	-1.00	0.00
T4,T3	0.50	0.00	0.00	-0.50	0.50	0.50	0.50	0.50	0.50	-1.00	-1.00	-1.00	-0.50	-1.00	1.00	0.00
T4,T4	-1.00	1.00	0.00	-1.00	-1.00	-1.00	1.00	1.00	-1.00	1.00	1.00	-1.00	-1.00	0.00	0.00	1.00

3.2.1 Singular shape features

The singular shape features used include extent, orientation, volume, centroid, and center of mass:

1) The extents and orientation of a solid object were approximated by the parameters of its bounding box. Axis-aligned bounding boxes (AABB), which were used in SeeBIM (Belsky, Sacks and Brilakis, 2016), exaggerate the extent of non-axis aligned shapes. In contrast, this work used tight-fitting bounding boxes (TFBB), generated by an optimized caliper algorithm (Jylanki, 2015), for such approximation. A TFBB consists of three axes and three lengths corresponding to them. The longest axis was assumed to be the extrusion direction of a longitudinal object.
2) The volume of a solid object was computed using voxelization. The solid was voxelized (Wang and Kaufman, 1993) into a discrete 3D binary image; each voxel was labeled with '1' or '0' to signify whether it was inside or outside the object, so that the volume of the object was approximated by the sum of all the voxel values. Most building elements have basic geometry or are combinations of basic primitives, so the voxelization is sufficiently accurate and efficient. For irregular shapes, one may consider using a tetrahedron based algorithm (Zhang and Chen, 2001) instead.
3) The centroid of a solid object, represented as a triangular mesh, was calculated as a weighted average of its face centroids, with the weights being the face areas. In addition, the center of mass was the same as its centroid provided that all the objects in the models were of homogeneous material and/or had uniform-density, which was assumed in this work. This algorithm has been implemented in many software packages, such as Blender (Hess, 2007).

3.2.2 Pairwise relationships

Pairwise relationships include proximity measures (e.g., touching and overlapping), orthogonality and parallelism conditions, and comparison of shape features:

1) Proximity checking was performed in two phases: broad phase and narrow phase (Mirtich, 1997). In the broad phase, the potentially overlapped object pairs were selected by pruning away pairs whose AABBs were disjoint. In the narrow phase, the exact contact points of the selected potentially overlapped object pairs were computed using ray casting test (Daum and Borrmann, 2014) and bounding volume hierarchy based collision tests (Pan et al., 2012). This checking method worked on non-convex shapes, which was also a notable enhancement of SeeBIM. Two objects were considered to be in contact if the distance between their closest faces was smaller than 5mm.
2) Orthogonality checking was performed on pairs of longitudinal objects. Objects were perpendicular to each other when the dot product of their extrusion direction, derived from TFBB, approximately equaled zero; they were parallel to each other when the product approximately equaled one. The tolerance of angles incorporated in the orthogonality and parallelism checking was 3º.
3) Comparisons of shape features provided another set of pairwise relationships. For example, the centroid of an object is higher than that of the other; the volume of an object is greater than that of the other, etc.

3.3 Object classification based on similarity measures

In many cases, the knowledge matrices may be inexact. For example, a girder is an I beam in many AASHTO girder bridges, and this could be a singular shape feature used in a matrix like Table 1 for identifying most girders. However, tee-girders also exist in some bridges, and this may be neglected by the expert preparing the matrices. In this case even though a tee-girder does not fully match the features of an I girder, a good object classifier should still be able to infer that it is a girder in general because the two types of girders are similar in many other features and relationships with other objects. A similarity based object classification algorithm was proposed for this purpose.

Figure 4 shows the procedures of the algorithm. Section 3.3.1 explains step 1 and step 2 for compiling the fact matrix; section 3.3.2 explains step 3 and step 4 for computing the similarity values; section 3.3.3 explains step 5 for selecting the best candidate as the classification result based on the similarity value.

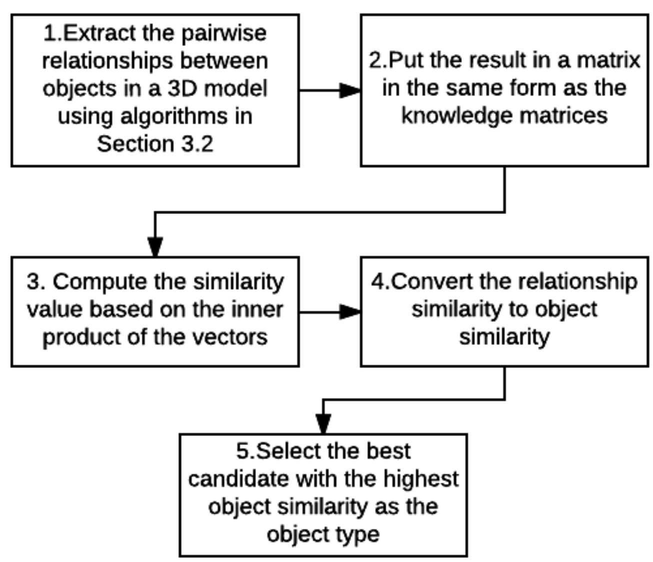


Figure 4 Compiling a knowledge matrix

3.3.1 Compilation of the fact matrix

Given a 3D model, the pairwise relationships of objects can be computed using the algorithms in Section 3.2. The results can be organized in the same matrix format as Table

2. For example, a model of three objects, i.e, O1, O2, and O3, could have pairwise relationships shown in Table 5.

Table 5

Computed pairwise relationships of three objects

Object pair	Relation1	Relation2	Relation3	Relation4
O1,O2	1	1	1	-1
O1,O3	-1	-1	1	1
O2,O1	-1	1	1	-1
O2,O3	1	1	1	1
O3,O1	1	1	1	-1
O3,O2	1	-1	1	-1

In Table 2, T1 represents an object type, and [T1, T1] represents two objects of the same type. However, O1 is a specific object in the 3D model, so pairs like [O1,O1] or [O2,O2] do not exist in Table 5. In addition, in a specific 3D model, a relationship between two objects either holds ('1') or does not hold ('-1'). The value '0', namely 'maybe', does not occur. The value in the table is named as fact matrix.

3.3.2 Computation of the similarity value

Each row in Table 5 is a four-dimensional vector $\mathbf{R}^*(\mathrm{i,j})$. $\mathbf{R}^*(1,2)$ represents all the relationships between object 1 and object 2. On the other hand, the $\mathbf{R}(1,2)$ in Table 2 represents all the relationships between an object of type T1 and an object of type T2. If the two vectors are the same then O1 is very likely to be an object of type T1 and O2 is very likely to be an object of type T2. The similarity of two vectors could be measured by the angle between them, which can be computed by the inner product of the two vectors. It is computed using the following equation, where $\mathrm{i,j} \in [1,2,3]$, $\mathrm{i} \neq \mathrm{j}$ and $\mathrm{m,n} \in [1,2,3,4]$. Table 6 shows all the similarity values computed using data in Table 5 and Table 2.

$$\mathbf{S}_{(\mathrm{i,j})(\mathrm{m,n})} = \frac{\mathbf{R}^*(\mathrm{i,j}) \cdot \mathbf{R}(\mathrm{m,n})}{|\mathbf{R}^*(\mathrm{i,j})||\mathbf{R}(\mathrm{m,n})|} \qquad (3)$$

All the shaded cells in Table 6 imply the similarity between O1 and T1, because they appear in the same position of the two pairs (one is the row header and the other one is the column header). Obviously, if the number of objects is u and the number of object types is v, then the number of shaded cells for any of such match is $2v(u-1)$. In this example, the number of shaded cells is $2\times4\times(3-1) = 16$.

So by summing up the value in all the cells and dividing the sum by 16, we get the normalized similarity between an object and an object type, as is shown in Table 7.

Table 7

The similarity between an object and an object type derived from Table 6

	T1	T2	T3	T4
O1	0.24	-0.12	0.03	-0.14
O2	0.35	-0.14	0.02	-0.11
O3	0.00	0.28	-0.22	0.03

3.3.3 Selection of the best candidate as object type

Due to the existence of '0', i.e. the uncertainty in the logical condition, in Table 2, the value in Table 7 could almost never be 1. In each row, the cell with the greatest value implies that its column header is the best candidate as the object's type. As a result, O1 is most likely to be of type T1, O2 is most likely to be T1, and O3 is most likely to be T2. In some situation, the similarity value of the best candidate and the second-best candidate are very close (the user can give a threshold for such differentiation), which then requires either incorporating new pairwise relationships to distinguish them or asking experienced users to classify the object.

In conclusion, the knowledge matrix and the fact matrix are compared in Euclidian space based on the angles between vectors. The matching is based on the pairwise relationships but the result can be used to derive the similarity between an object and an object type following an identified pattern in the matching result. The similarity value is used to select the best candidate as the final object classification.

4 DEMONSTRATION OF THE APPROACH IN TWO SCENARIOS

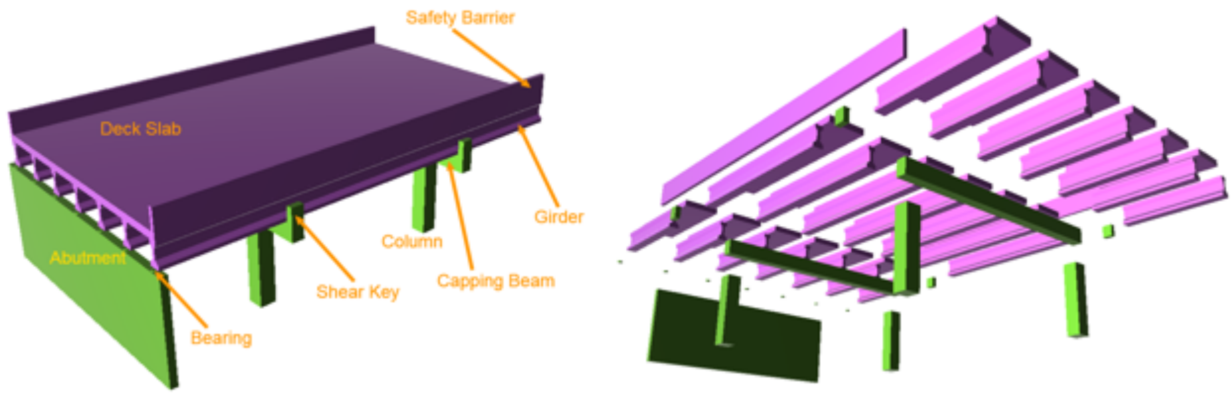
The benefits of using BIM in asset management is becoming clearer. This drives the need to incorporate semantically rich bridge information models in bridge management systems (BMS).

For new built bridge projects, efforts have been made to use BIM in bridge design. However, many bridge objects do

Table 6

$\mathbf{S}_{(\mathrm{i,j})(\mathrm{m,n})}$ computed using Eq. 3 and data in Table 5 and Table 2

obj2	T1,T1	T1,T2	T1,T3	T1,T4	T2,T1	T2,T2	T2,T3	T2,T4	T3,T1	T3,T2	T3,T3	T3,T4	T4,T1	T4,T2	T4,T3	T4,T4
O1,O2	0.71	-0.87	0.71	0.29	0.71	0.00	0.29	-0.71	0.87	0.00	-0.29	0.00	0.00	0.00	0.00	-0.50
O1,O3	-0.71	0.87	-0.71	0.29	-0.71	0.00	0.29	0.71	-0.29	0.50	0.29	-0.50	0.00	0.00	0.00	0.50
O2,O1	0.00	-0.29	0.71	-0.29	0.00	-0.71	0.87	0.00	0.29	0.50	0.29	-0.50	-0.71	0.00	0.00	0.50
O2,O3	0.00	-0.29	0.00	0.87	0.00	0.00	-0.29	-0.71	0.29	0.50	0.29	0.50	0.71	0.71	-0.71	-0.50
O3,O1	0.71	-0.87	0.71	0.29	0.71	0.00	0.29	-0.71	0.87	0.00	-0.29	0.00	0.00	0.00	0.00	-0.50
O3,O2	0.71	-0.29	0.00	0.29	0.71	0.71	0.29	0.00	0.87	-0.50	-0.87	-0.50	0.00	-0.71	0.71	-0.50



(a) (b)

Figure 5 a) overview and b) exploded view of the synthetic bridge model

not exist in BIM software object libraries (such as bearings, shear keys, and abutments). The designers must substitute available types of objects to represent them (e.g., using a wall object to represent an abutment). The object types need to be corrected before the model can be used in a BMS. Section 4.1 demonstrates the process of correcting the classification of the objects in a bridge model using the proposed approach.

For existing bridges, laser scanning is an efficient way to collect the spatial data of the site. The data can be used for compiling accurate 3D models of the bridges. After the 3D reconstruction, it is crucial to classify the generated 3D objects correctly. Section 4.2 demonstrates an application of the proposed approach to classify the objects in a 3D bridge model created based on laser scanned point cloud data.

4.1 Correct the types of misclassified objects

The bridge consists of 59 objects of eight types, but the model received had just three types of objects, as is shown in Table 8 and Figure 5. Before the bridge model could be used in the BMS, these objects need to be correctly classified. This was done using the proposed approach.

Table 8
Objects in the bridge model

Symbol	Bridge objects	Number	Object type in the model
A	Abutment	1	Beam
B	Bearing	7	Column
C	Column	4	Column
D	Deck Slab	18	Slab
E	Girder	21	Beam
F	Safety Barrier	2	Beam
G	Shear Key	4	Column
H	Capping Beam	2	Beam

Compiling the knowledge matrix for this type of girder bridge required three hours work with a bridge engineer. The team defined 18 logical conditions in three different categories:

1) Logical conditions of the spatial relationships:
   I. Are the two objects in contact?
   II. Are the two objects parallel?
2) Comparison of the shape features:
   III. Is object 1's centroid higher than that of object 2?
   IV. Is object 1's bottom face lower than object 2's?
   V. Is object 1 longer than object 2?
   VI. Is object 1's volume greater than that of object 2?
   VII. Is object 1 completely above object 2?
   VIII. Is object 1 closer to the road axis than object 2?
   IX. Is object 1 closer to the transverse axis?
   X. Do the two objects overlap along the vertical axis?
3) Logical conditions of the shape features where each object in a pair is evaluated independently:
   XI. Is object 1 longitudinal?
   XII. Is object 2 longitudinal?
   XIII. Is object 1 transverse?
   XIV. Is object 2 transverse?
   XV. Is object 1 vertical?
   XVI. Is object 2 vertical?
   XVII. Is object 1 convex?
   XVIII. Is object 2 convex?

Table 9 shows the logical conditions of pairwise relationships between girders and all the bridge object types. The logical conditions of the same pairwise relationships between every pair of objects in the model were computed. The 59 objects were involved in $59 \times 58 = 3{,}422$ pairs, so the whole result was a $3{,}422 \times 18$ matrix.

Based on the two matrices, the similarities between each of the 59 objects and each of the eight object types were computed using Eq. 3. Table 10 shows part of the result. Object 1 and Object 2 are most likely to be girders (E,

Table 9
Girder (E in Table 8) related pairwise relations

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII
E, E	0	1	-1	-1	0	0	-1	0	0	-1	1	1	-1	-1	-1	-1	-1	-1
E, A	-1	-1	1	-1	-1	-1	1	-1	1	-1	1	-1	-1	1	-1	-1	-1	1
E, B	0	-1	1	-1	1	1	1	0	1	-1	1	-1	-1	-1	-1	1	-1	1
E, C	-1	-1	1	-1	1	0	1	0	0	-1	1	-1	-1	-1	-1	1	-1	1
E, D	1	1	-1	1	0	-1	-1	0	0	-1	1	1	-1	-1	-1	-1	-1	1
E, F	-1	1	-1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1
E, G	-1	-1	1	-1	1	1	-1	1	0	1	1	-1	-1	-1	-1	1	-1	1
E, H	1	-1	1	-1	-1	-1	1	-1	0	-1	1	-1	-1	1	-1	-1	-1	1
A, E	-1	-1	-1	1	1	1	-1	0	-1	-1	-1	1	1	-1	-1	-1	1	-1
B, E	0	-1	-1	1	-1	-1	-1	0	-1	-1	-1	1	-1	-1	1	-1	1	-1
C, E	-1	-1	-1	1	-1	0	-1	0	0	-1	-1	1	-1	-1	1	-1	1	-1
D, E	1	1	1	-1	0	1	1	0	0	-1	1	1	-1	-1	-1	-1	1	-1
F, E	-1	1	1	-1	1	1	1	-1	0	-1	1	1	-1	-1	-1	-1	1	-1
G, E	-1	-1	-1	-1	-1	-1	-1	-1	0	1	-1	1	-1	-1	1	-1	1	-1
H, E	1	-1	-1	1	1	1	-1	0	0	-1	-1	1	1	-1	-1	-1	1	-1

Table 10
Matching result of the first bridge example computed using Eq. 3

Object	A	B	C	D	E	F	G	H
1	-0.07	0.02	0	0.24	0.35	0.27	0.04	-0.06
2	-0.08	-0.09	-0.01	0.24	0.34	0.32	-0.02	-0.02
				…				
59	0	-0.02	0	0.44	0.25	0.55	0.06	0.07

according to Table 8)), while Object 59 is most likely to be a safety barrier (F, according to Table 8). For each object, the object type with highest similarity value was the object classification result. All the objects were correctly identified using this approach.

4.2 Classifying objects reconstructed from point clouds

Laser scanning is an effective method to collect spatial data for creating 'as-is' bridge models. Once a 3D geometric model has been reconstructed from laser scanned PCD, which may be partly automated, the objects must be correctly classified. A reconstructed 3D model of a concrete girder highway bridge, on Route 79 in Haifa, Israel, is used to illustrate application of the approach for object classification.

The 3D model, which was manually prepared using Revit by tracing the PCD, is shown in **Figure 6**. All the objects were modelled using available standard objects in Revit. The model contained 331 bridge objects of nine different types, as shown in **Table 11**.

Table 11
Objects in the Route 79 bridge model

Correct concrete girder bridge object types	Number of objects	Object types used in Revit
Primary girder	30	Beam
Capping beam	2	Beam
Transverse beam	99	Beam
Precast deck panel	162	Slab
Plinth on capping beam	20	Column
Shear key on capping beam	4	Column
Shear key on abutment	4	Column
Abutment	2	Wall
Column	8	Column

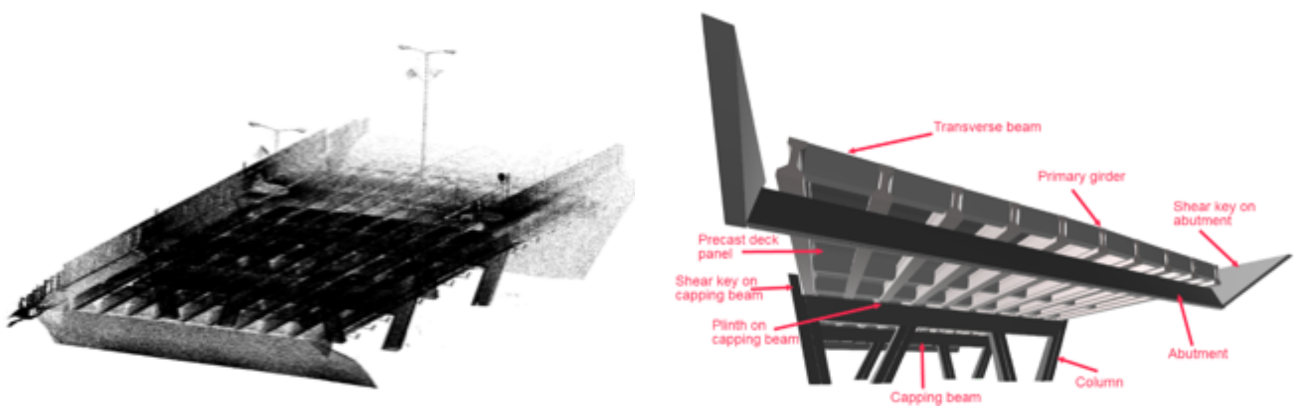
Knowledge matrices compiled by different users (domain experts) could be different, even if the matrices are for identifying the same type of structure, because people recognize objects in different ways. The team required two hours to compile an 81×19 knowledge matrix, independently from the first bridge example, for the bridge of this kind. The matrix evaluated 19 different logical conditions:

1. Is object 1 in contact with object 2's side face?
2. Is object 1 in contact with object 2's front / back face?
3. Is object 1 in contact with object 2's bottom face?
4. Is object 1 in contact with object 2's top face?
5. Are the two objects in contact?
6. Are the two objects in parallel?
7. Is object 1's centroid higher than object 2's?
8. Is object 1's extrusion longer than object 2's?
9. Is object 1's volume greater than object 2's?
10. Is object 1 wider than object 2?
11. Is object 1 taller than object 2?
12. Is object 1 vertically extruded?
13. Is object 1's extrusion direction parallel to the road axis?
14. Is object 1's extrusion direction parallel to the skew angle of the bridge supports?
15. Is object 1 horizontal?
16. Is object 2 vertically extruded?
17. Is object 2's extrusion direction parallel to the road axis?
18. Is object 2's extrusion direction parallel to the skew angle of the bridge supports?
19. Is object 2 horizontal?

Then, a 109,230×19 fact matrix ($331\times(331-1)=109{,}230$) was derived by computing the actual pairwise conditions of the objects in the bridge model. Based on the two matrices, the similarities between each of the 331 objects and each of the nine object types were computed using Eq. 3. Table 12 shows part of the result. For each object, the object type with highest similarity value was the object classification result. All the objects were correctly identified using this approach.

5 DISSCUSSION

With the two tailored knowledge matrices, the presented approach was able to classify all the 3D objects in the two example bridges with 100% accuracy. The examples were used to prove the feasibility of the approach to expressing human knowledge using the proposed matrix format; they also showed the flexibility of the approach to classifying the same object types using different knowledge matrices, which is very practical because different people may have different



(a) (b)

Figure 6 a) point cloud and b) 3D model of the concrete girder bridge

Table 12

Part of matching result of the second bridge model

	Candidate1	Candidate2	Candidate3
Object 1	Girder (0.23)	Shear key on abutment (0.18)	Capping beam (0.16)
Object 2	Plinth (0.38)	Shear key on capping beam (0.29)	Column (0.21)
Object 3	Column (0.31)	Abutment (0.27)	Shear key on abutment (0.23)
		…	

perceptions concerning the features and relationships to be used for recognizing the same object types.

The object types in the two bridges are slightly different. For example, there are no plinths in the first bridge, while safety barriers were not modeled in the second bridge. Therefore, the knowledge matrices of the two examples were different. One knowledge matrix can be reused for automated object classification in models that have the same list of object types. Naturally, other domains will have different lists of objects, and although the specific knowledge for their classification will also be different, they will nevertheless have features and relationships that can be used, so that the approach is applicable.

To compile a knowledge matrix applicable for a variety of structures would require careful and comprehensive evaluation of all object types that could possibly be included. However, the goal of this research was not to propose a versatile knowledge matrix that can be used for all kinds of structures, but to devise an effective knowledge representation and a mathematical algorithm for object classification using the formalized knowledge.

The implementation of the presented approach had a restriction: the knowledge matrices could only be built on the features and relationships of which the logical conditions can be computed, so that the matching between the knowledge and the reality can be performed. Features and relationships that are additional to the ones in section 3.2 could make the approach more versatile if they could be computed. For example, identifying walls with windows may require evaluation of the solids inclusion conditions. An additional caveat for the scenario of 'scan-to-BIM' is that the approach assumes the existence of a complete 3D model. However, current research has not yet achieved full capability to reconstruct accurate and complete 3D models, due to the problems like occlusion in the scans.

6 CONCLUSION

Lack of object classification information impedes the full exploitation of BIM. Given the difficulties encountered by rule-processing systems in processing rule sets derived from human experts directly, this research proposed an alternative approach to compiling expert knowledge, to processing it, and thus for effective object classification. The approach has been implemented in concrete girder bridges. However, the meaning of the work and its results is broader than demonstrating effectiveness of the approach for this domain.

The approach has the following innovations:

1) It encapsulates domain experts' knowledge of the relationships between objects' types and their shape features and pairwise relations in a matrix form.
2) It incorporates state-of-the-art algorithms for processing the geometry of BIM models.
3) It provides an algorithm to evaluate the comprehensiveness of object classification knowledge.
4) It provides a feature-based matching algorithm based on mathematical computation.

In ongoing research, the approach presented could be enhanced in the following ways:

1) The knowledge matrices have incorporated singular shape features and pairwise relationships. Three-way relationships can also be adopted in a similar way, thus leveraging more information to distinguish objects.
2) Compilation of knowledge matrices requires domain specific skills and experience. We are considering a hybrid approach that is less dependent on human expert knowledge, and more on machine-learning. This might alleviate the need for explicit and rigorous statement of the object relationship matrices.

At present, however, machine-learning for 3D object classification is seriously hampered by the lack of a population of BIM models. In the domain of concrete highway bridges, for example, a very large sample of fully detailed and properly classified BIM models would be needed for training a machine learning algorithm, and no such populations exist. Given that most new projects are built with the aid of BIM models, it may become possible in the future to apply machine learning for automatically establishing the implicit knowledge bases required for object classification.

7 REFERENCE

Adeli, H. (2003), *Expert Systems in Construction and Structural Engineering*, CRC Press.

Anand, A., Koppula, H. S., Joachims, T. & Saxena, A. (2013), Contextually Guided Semantic Labeling and Search for Three-Dimensional Point Clouds, *International Journal of Robotics Research*, **32**(1), 19-34.

Arias, P., Caamano, J. C., Lorenzo, H. & Armesto, J. (2007), 3d Modeling and Section Properties of Ancient Irregular Timber Structures by Means of Digital Photogrammetry, *Computer-Aided Civil and Infrastructure Engineering*, **22**(8), 597-611.

Babic, B., Nesic, N. & Miljkovic, Z. (2008), A Review of Automated Feature Recognition with Rule-Based Pattern Recognition, *Computers in Industry*, **59**(4), 321-337.

Belsky, M., Sacks, R. & Brilakis, I. (2016), Semantic Enrichment for Building Information Modeling, *Computer-Aided Civil and Infrastructure Engineering*, **31**(4), 261-274.

Bosche, F., Ahmed, M., Turkan, Y., Haas, C. T. & Haas, R. (2015), The Value of Integrating Scan-to-Bim and Scan-Vs-Bim Techniques for Construction Monitoring Using Laser Scanning and Bim: The

Case of Cylindrical Mep Components, *Automation in Construction*, **49**, 201-213.

Clancey, W. J. (1983), The Epistemology of a Rule-Based Expert System—a Framework for Explanation, *Artificial intelligence*, **20**(3), 215-251.

Czerniawski, T., Nahangi, M., Haas, C. & Walbridge, S. (2016), Pipe Spool Recognition in Cluttered Point Clouds Using a Curvature-Based Shape Descriptor, *Automation in Construction*, **71**, 346-358.

Daum, S. & Borrmann, A. (2014), Processing of Topological Bim Queries Using Boundary Representation Based Methods, *Advanced Engineering Informatics*, **28**(4), 272-286.

Diaz-Vilarino, L., Conde, B., Laguela, S. & Lorenzo, H. (2015), Automatic Detection and Segmentation of Columns in as-Built Buildings from Point Clouds, *Remote Sensing*, **7**(11), 15651-15667.

Dimitrov, A. & Golparvar-Fard, M. (2015), Segmentation of Building Point Cloud Models Including Detailed Architectural/Structural Features and Mep Systems, *Automation in Construction*, **51**, 32-45.

Fathi, H. & Brilakis, I. (2016), Multistep Explicit Stereo Camera Calibration Approach to Improve Euclidean Accuracy of Large-Scale 3d Reconstruction, *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, **30**(1), 11.

Hess, R. (2007), *The Essential Blender: Guide to 3d Creation with the Open Source Suite Blender*, No Starch Press.

Jardon, A., Victores, J. G., Martinez, S. & Balaguer, C. (2012), Experience Acquisition Simulator for Operating Microtuneling Boring Machines, *Automation in Construction*, **23**, 33-46.

Jung, J., Hong, S., Yoon, S., Kim, J. & Heo, J. (2016), Automated 3d Wireframe Modeling of Indoor Structures from Point Clouds Using Constrained Least-Squares Adjustment for as-Built Bim, *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, **30**(4), 15.

Jylanki, J. (2015), An Exact Algorithm for Finding Minimum Oriented Bounding Boxes.

Kalasapudi, V. S., Tang, P. & Turkan, Y. (2017), Computationally Efficient Change Analysis of Piece-Wise Cylindrical Building Elements for Proactive Project Control, *Automation in Construction*, **81**, 300-312.

Kao, W. M. & Adeli, H. (2002), Multitasking Object-Oriented Blackboard Model for Design of Large Space Structures, *Engineering Intelligent Systems for Electrical Engineering and Communications*, **10**(1), 3-8.

Kim, C., Son, H. & Kim, C. (2013), Automated Construction Progress Measurement Using a 4d Building Information Model and 3d Data, *Automation in Construction*, **31**, 75-82.

Mirtich, B. (1997), Efficient Algorithms for Two-Phase Collision Detection, *Practical motion planning in robotics: current approaches and future directions*, 203-223.

Paal, S. G., Jeon, J. S., Brilakis, I. & DesRoches, R. (2015), Automated Damage Index Estimation of Reinforced Concrete Columns for Post-Earthquake Evaluations, *Journal of Structural Engineering*, **141**(9), 13.

Pan, J., Chitta, S. & Manocha, D. (2012), Fcl: A General Purpose Library for Collision and Proximity Queries, *2012 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 3859-3866.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R. & Van Campenhout, J. (2011), A Semantic Rule Checking Environment for Building Performance Checking, *Automation in Construction*, **20**(5), 506-518.

Ruiz-Sarmiento, J. R., Galindo, C. & Gonzalez-Jimenez, J. (2015), Exploiting Semantic Knowledge for Robot Object Recognition, *Knowledge-Based Systems*, **86**, 131-142.

Sacks, R., Ma, L., Yosef, R., Borrmann, A., Daum, S. & Kattel, U. (2017), Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry, *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, **31**(6), 04017062.

Son, H., Kim, C. & Kim, C. (2015), 3d Reconstruction of as-Built Industrial Instrumentation Models from Laser-Scan Data and a 3d Cad Database Based on Prior Knowledge, *Automation in Construction*, **49**, 193-200.

Steffens, T. (2006), *Enhancing Similarity Measures with Imperfect Rule-Based Background Knowledge*, IOS Press.

Tang, P., Chen, G. & Shen, Z. (2016), A Spatial-Context-Based Approach for Automated Spatial Change Analysis of Piece-Wise Linear Building Elements, *Computer-Aided Civil and Infrastructure Engineering*, **31**(1), 65-80.

Valero, E., Adan, A. & Bosche, F. (2016), Semantic 3d Reconstruction of Furnished Interiors Using Laser Scanning and Rfid Technology, *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, **30**(4), 19.

Walsh, S. B., Borello, D. J., Guldur, B. & Hajjar, J. F. (2013), Data Processing of Point Clouds for Object Detection for Structural Engineering Applications, *Computer-Aided Civil and Infrastructure Engineering*, **28**(7), 495-508.

Wang, C., Cho, Y. K. & Kim, C. (2015), Automatic Bim Component Extraction from Point Clouds of

Existing Buildings for Sustainability Applications, *Automation in Construction*, **56**, 1-13.

Wang, S. W. & Kaufman, A. E. (1993), Volume Sampled Voxelization of Geometric Primitives, *Visualization, 1993. Visualization'93, Proceedings., IEEE Conference on*, IEEE, pp. 78-84.

Zeibak-Shini, R., Sacks, R., Ma, L. & Filin, S. (2016), Towards Generation of as-Damaged Bim Models Using Laser-Scanning and as-Built Bim: First Estimate of as-Damaged Locations of Reinforced Concrete Frame Members in Masonry Infill Structures, *Advanced Engineering Informatics*, **30**(3), 312-326.

Zhang, C. & Chen, T. (2001), Efficient Feature Extraction for 2d/3d Objects in Mesh Representation, *International Conference on Image Processing*, IEEE, pp. 935-938.

Zhang, G., Vela, P. A., Karasev, P. & Brilakis, I. (2015), A Sparsity-Inducing Optimization-Based Algorithm for Planar Patches Extraction from Noisy Point-Cloud Data, *Computer-Aided Civil and Infrastructure Engineering*, **30**(2), 85-102.

Zhu, Z. H. & Brilakis, I. (2010), Concrete Column Recognition in Images and Videos, *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, **24**(6), 478-487.