Invariant Signatures of Architecture, Engineering, and Construction Objects to Support BIM Interoperability between Architectural Design and Structural Analysis

Jin Wu, S.M. ASCE[1]; Hezha Lutfalla Sadraddin, S.M. ASCE[2], Ran Ren, S.M. ASCE[3]: Jiansong Zhang, A.M. ASCE[4]; and Xiaoyun Shao, A.M. ASCE[5]

Abstract

There has been an increasing demand in building information modeling (BIM) for structural analysis. However, model exchange between architectural software and structural analysis software, which is an essential task in a construction project, is not fully interoperable yet. Various studies showed that there are information missing and information inconsistency problems during conversion of models between different software; and the lack of foundational methods enabling a seamless BIM interoperability between architectural design and structural analysis is evident. To address this gap and facilitate more use of BIM for structural analysis, the authors developed invariant signatures for architecture, engineering, and construction (AEC) objects and proposed a new data-driven method to use invariant signatures for solving practical problems in BIM applications. The invariant signatures and the data-driven method were tested in developing the interoperable BIM support tool for structural analysis through an experiment. Ten models were created and used in this experiment, including five models for training and five models for testing. Information validation and mapping algorithm was developed based on invariant signatures and training models, which was then evaluated in the testing models. Comparing with a manually created gold standard, results showed that the desired structural analysis software input were

[1] Automation and Intelligent Construction (AutoIC Lab), School of Construction Management Technology, Purdue University, 401 N. Grant Street, West Lafayette, IN 47907. E-mail: wu1275@purdue.edu.
[2] Department of Civil and Construction Engineering, Western Michigan University, Kalamazoo, Michigan, USA. E-mail: hezhalutfalla.sadraddin@wmich.edu.
[3] Automation and Intelligent Construction (AutoIC Lab), School of Construction Management Technology, Purdue University, 401 N. Grant Street, West Lafayette, IN 47907. E-mail: ren153@purdue.edu.
[4] Automation and Intelligent Construction (AutoIC Lab), School of Construction Management Technology, Purdue University, 401 N. Grant Street, West Lafayette, IN 47907 (corresponding author). E-mail: zhan3062@purdue.edu.
[5] Department of civil and Construction Engineering, Western Michigan University, Kalamazoo, Michigan, USA. E-mail: xiaoyun.shao@wmich.edu.

successfully generated using the algorithm. The invariant signatures of AEC object can therefore be expected to serve as the foundation of seamless BIM interoperability.

Author keywords: BIM interoperability; Structural analysis; Architectural design; Geometric signature; Material signature.

Introduction

There has been an increasing demand in building information modeling (BIM) for structural analysis. The comprehensive nature of BIM predicts its use in different applications in the architectural, engineering, and construction (AEC) domain, in an interoperable way. This naturally led to the expectation that an architectural model should directly support engineering analysis such as structural analysis. However, this is far from reality. As a McGraw-Hill Construction report shows, the value/difficulty ratio for engineering analysis using BIM is very low (McGraw-Hill Construction 2014). This ratio for structural analysis is even negative, which means a structural designer/engineer is better off creating a structural analysis model from scratch rather than reusing the BIM model generated from the corresponding architectural design process.

A structural analysis consists of a series of actions including structural geometry and support design, material mechanical properties definition, load calculation and assignment, system and component analysis and design to determine suitability when exposed to external loading effects. Because model exchange between architectural and structural analysis software, which is an essential task in a construction project, is not fully interoperable yet. Structural analysis is usually forced to include a pre-processing step which takes most of the time and effort to model the whole structure (i.e., geometry and support creation, material definition, and load assignment, etc.) during the structural analysis process. An interoperable model exchange would shorten this modeling time and reduce misunderstanding between architects and structural engineers. Example challenges faced by existing methods for interoperable BIM use for structural analysis include information missing, joint disconnection, element embedment/nesting (i.e., one element embedded completely inside another), and element overlapping (i.e., one member partially cuts into another) (Muller et al. 2017).

To facilitate an interoperable BIM use for structural analysis, the authors developed invariant signatures for AEC objects and proposed a new data-driven method to use invariant signatures for solving practical problems in BIM applications. This paper introduces the invariant signatures concept and its application method, as well as its implementation in the context of structural analysis where an algorithm was developed for converting architectural models to structural analysis inputs based on invariant signatures.

Background

BIM Interoperability and Knowledge Gaps for Structural Analysis

The AEC domain is meeting technical challenges in software interoperability and the amount of information or data types that need to be processed and communicated. Among the many phases and tasks in the AEC domain, architectural design and structural analysis are two important ones with the software and information interoperability between them being extensively researched. Table 1 shows a summary of example work on BIM interoperability between architectural and structural models.

Table 1. Example work on BIM interoperability between architectural and structural models

Issue	Literature
Information missing and information inconsistency	Sacks et al. (2010), Bank et al. (2010), Ramaji and Memari (2018), Ren et al. (2018)
Information missing and information inconsistency; Software function limitations	Hu et al. (2016), Aldegeily et al. (2018)
Software function limitations	Watson (2011), Yalcinkaya et al. (2014), Fleming (2016)

Model visualization limitations	Sanguinetti et al. (2012), Steel et al. (2012)
Geometric/spatial complications	Muller et al. (2017),
Other	Yang and Zhang (2006), Grilo and Jardim-Goncalves (2010), Nawari (2011), Solnosky et al. (2014), Shin (2017)

As shown in Table 1, there are several types of challenges in BIM interoperability between architectural design and structural analysis in the AEC domain. For example, information missing and information inconsistency (Sacks et al. 2010; Bank et al. 2010); software function limitations in information conversion (such as geometric information) (Watson 2011; Yalcinkaya et al. 2014; Fleming 2016); model visualization limitations in BIM interoperability (Sanguinetti et al. 2012; Steel et al. 2012); and the geometric/spatial complications (e.g., overlapping of structural parts in a model) that affect an efficient BIM interoperability (Muller et al. 2017).

The information-related challenges have been the focus of the research community. Various studies showed that between architectural design and structural analysis, there can be information missing and information inconsistency problems during the conversion of models between different software. For example, only geometric components of a model could be transmitted between Tekla structure (Trimble Company 2019) and Revit Architecture 2008, and cross-sectional properties were missing and must be set separately if there are many elements in a model (Sacks et. al 2010). Some preliminary tests showed that in SAP2000 (Computers and Structures 2019), the imported material information through an IFC file could not be loaded into a structural model to conduct structural analysis (Ren and Zhang 2019). Different types of boundary conditions in Revit (pinned, roller, and fixed) are treated as pinned when transferred to ETABS and SAFE (Computers and Structures 2019; SAFE Software 2019; Aldegeily et al. 2018). BIM interoperability workflows/pathways between architectural design and structural analysis have been investigated by few researchers. For example, Ramaji and Memari (2018) summarized three workflows between building information models (BIMs) and analytical structural models: (1) structural analysis models could be exported from BIM authoring tools directly; (2) BIMs could be imported into structural analysis tools to establish structural analysis model in reference to the architectural model. (3) BIMs could be transferred to analytical structural models by a third-party application. Aldegeily et al. (2018) summarized three types of paths for data transfer between architectural and structural models: (1) "direct link using native file, which is the direct link between software programs from the same provider"; (2) "direct link using API, which is the data transfer with a BIM platform through its APIs" and; (3) "indirect link, which is the indirect transfer of information through third-party software or methods/algorithms." However, there is a lack of foundational methods that enable a seamless BIM interoperability between architectural design and structural analysis.

Industry Foundation Classes

Industry Foundation Classes (IFC) is a set of building product models, which was first specified in 1996 (BuildingSMART 2017) and is now registered as an international standard ISO 16739. It is the most widely used non-proprietary exchange format for representing building information (Volk et al. 2014). In an IFC model, information of a project is represented as a set of IFC entities, such as "*IfcBeam,*" "*IfcColumn,*" "*IfcWall,*" and "*IfcSlab.*" Each entity includes a certain number of IFC attributes and additional IFC properties. IFC-based BIMs provide a uniform standard of information exchange for BIM applications and therefore a potential interoperability solution between different BIM software, such as Autodesk Revit (Autodesk 2019), Tekla Structure (Trimble Company 2019), and SAP2000. Many software can import an IFC file or export an IFC file for supporting lifecycle facility data management in BIM. However, the IFC exportation function does not cover all needed data for structural analysis. For example, they may not support the representation of constraints (e.g., boundary conditions), rules (e.g., structural code requirements), or parameters of complex entities (Thein 2010; Ren et al. 2018).

Invariant Signatures

The idea of invariant signatures for AEC objects originated from how objects were visually detected and how concepts were rigorously defined, in neural signatures and mathematical signatures, respectively. For visual detection of objects, the mystery of how the brain works is gradually unraveled with the development in biology, especially the perceptual system, i.e., how people recognize an object by patterns and features (Brandman and Peelen 2017; Johnson and Olshausen 2003). Different from neural signatures, mathematical signatures consist of rigid rules that uniquely define a concept. The corresponding signatures are supported by accepted axioms and proved theorems (Coolidge 1902). In addition, mathematical signatures shall cover all possible cases without major modifications (Daniyarova et al. 2012).

Both neural signatures and mathematical signatures used the idea of signatures of objects to detect and recognize objects. This idea of signatures is also widely used in other scenarios (Stow et al. 2012; Nelson and Sokkappa 2008). In the field of BIM technology, geometric information was widely used and needed in different tasks in the AEC domain, including architectural design, structural analysis, and others. It is an essential part and usually takes a large portion of BIM data (Zhang 2018). To capture the essence of geometric information for AEC objects in a systematic way, Wu and Zhang (2019a) proposed invariant geometric signatures with mathematical definitions, geometric theorems, and statistical measures to capture the key geometric information about an AEC object. They proposed 9 features (e.g., dimensional ratios, number of surfaces) to capture the geometric properties and used them to develop an algorithm that can classify 336 building objects with a higher than 85% accuracy. This invariant geometric signature has been successfully tested in supporting BIM object classification into common AEC objects including footings, slabs, walls, beams, and columns (Wu and Zhang 2018; 2019b). The research demonstrated in this paper further explores the usability of the invariant geometric signatures in supporting BIM interoperability between architectural design and structural analysis, by also expanding it into invariant signatures accounting for not only geometric information but material information. Such invariant signatures are expected to be adaptable and functional for multiple (if not all) application tasks in the AEC domain, including structural analysis.

Proposed Invariant Signature and Its Application Methodology

To adapt the invariant geometric signatures to its use in structural analysis, the authors extend the concept of invariant geometric signatures to a richer concept of invariant signatures. The authors define the invariant signature of an AEC object to be a set of intrinsic properties of the object that distinguish itself from others and that do not change with data schema, software implementation, modeling decisions, and/or language/culture contexts. Such invariant signatures should uniquely describe the object and provide all the information needed for various analyses. Different from the predefined category of IFC entities, e.g., *IfcShapeRepesentation*, *IfcIShapeProfile*, the invariant signatures capture the essence of an AEC object, which describes the scientific nature and does not depend on the schema or data structure used to represent the object. For example, the IFC schema contains 24 entities (e.g., *IfcProfileDef*) in *IfcProfileResource*, so that many irregular shapes can be represented differently, e.g., using boundary representation or extruding a 2-D plane. The invariant signatures can handle such complications by generating the same properties for the same objects represented differently, which separates the task of studying the IFC schema from the task of solving an AEC problem (e.g., object classification or structural analysis). In addition, the invariant signatures can be expanded with additional properties tailored for a specific purpose.

For example, AEC object classification and structural analysis can utilize different sets of properties. While object classification relies more on dimensional ratios, structural analysis relies more on geometric relations. For the task of structural analysis, the authors define the invariant signatures of AEC objects (Fig. 1) with the following two sets of properties: geometric properties (i.e., geometric signatures) and material properties (i.e., material signatures). The geometric properties cover both the shape and locational information of an object. The shape information is carried by the following nine properties as defined in the previous invariant geometric signature: "number of sub-components, number of faces, cross-sectional profile, extrusion direction, dimensional ratios, number of straight lines and curves, boundary line connection angles, lengths, and turn directions" and can be instantiated in the developing phase. The locational information is reflected by the x,y,z coordinates of the origin when placing an object as well as its orientation. The material information includes a set of six parameters as follows: material strengths, mass density, Poisson ratio, shear modulus, thermal expansion coefficient, and Young's modulus for structural analysis.

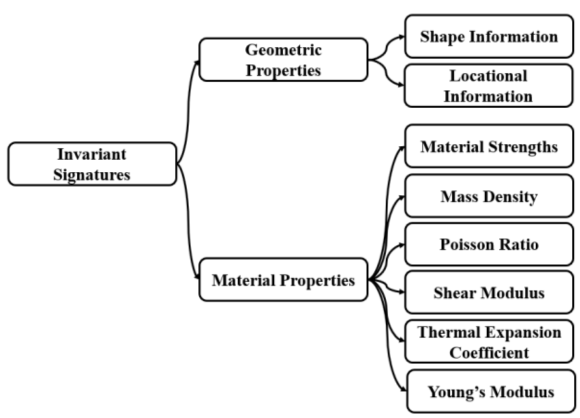


Fig. 1. Proposed invariant signature

To facilitate the practicality of the invariant signatures, the authors also propose a method for applying the invariant signatures to solving a practical problem, in a data-driven manner. The proposed method includes the following five steps: (1) problem definition, (2) data collection/creation, (3) preprocessing, (4) invariant signature application, and (5) evaluation (Fig. 2). The problem definition step defines and clarifies the problem to be addressed, by explicitly specifying the input and output of the expected analysis algorithms. The data collection/creation step gathers needed data used to develop solutions for the defined problem, through collecting data from existing resources, or creating data if existing data is not readily available. The gathered data is divided into a training dataset and a testing dataset, following a ratio that ranges from 80/20 to 50/50. The ratio cannot go below 50/50 because the data-driven nature of this method requires similar data in the training dataset to those in the testing dataset. The preprocessing step analyzes the data in both training and the testing dataset, to generate gold standards of the expected output from these data. The invariant signature application step leverages an iterative process that consists of feature extraction, algorithm development, and testing. The feature extraction sub-step extracts candidate features from the training data (i.e., data in the training dataset) based on the geometric and material properties defined in invariant signatures. The algorithm development sub-step develops two main sub-algorithms: information validation sub-algorithm, and information mapping sub-algorithm. The information validation sub-algorithm validates if a model contains all the required feature values for the problem of interest, while the information mapping sub-algorithm maps these feature values into targeted output. The testing sub-step compares the results generated by the developed algorithm with the gold standard in the training data: if there is any error, the method moves back to the feature extraction sub-step and starts another iteration; otherwise the method proceeds to the evaluation step, during which the developed algorithm is applied to the testing dataset and its processing results are compared with the gold standard. The comparison produces an accuracy measurement that evaluates the performance of the developed algorithm.

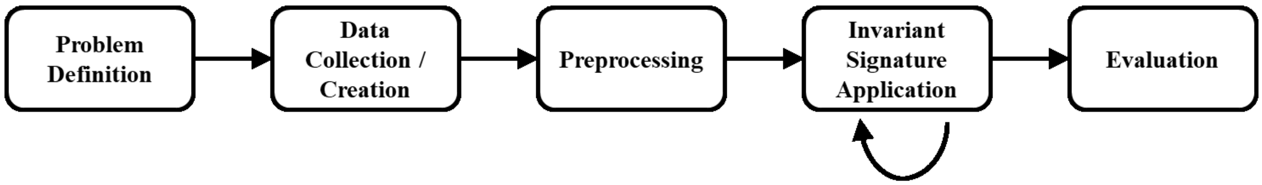


Fig. 2. Five-step method for applying the invariant signatures to solving a practical problem

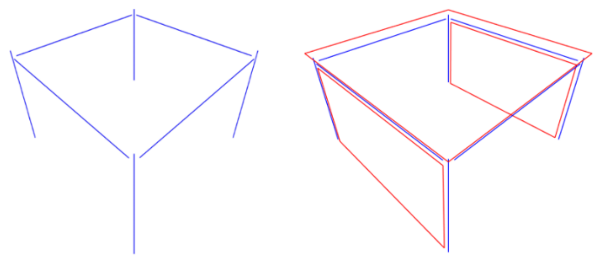
Experimental Testing, Results, and Discussion

To experimentally evaluate the methodology, the authors followed the five-step method in Fig. 2. With a clear definition of the problem in step 1, the authors collected and processed data based on which the authors used an iterative approach to develop the processing algorithm with sub-algorithms for information validation and information mapping. Finally, the authors evaluated the developed algorithm in the testing data and evaluatively analyzed the results.

Problem Definition

According to the knowledge gaps identified in the background section, the authors defined the problem to be enabling an automated and error-free (i.e., seamless) indirect (i.e., through third-party algorithm) information transfer from an architectural design model to a structural analysis model. The input to the expected processing algorithm will be an architectural model. The output from the expected processing algorithm will be a structural analysis model. IFC format was selected to represent the architectural modeling information, because it is the most widely-used international standard for AEC data.

To reveal IFC issues in structural analysis, SAP2000 structural analysis software (Computers and Structures 2019) was utilized to read two simple one-bay structures - one with four beams and four columns and another with four columns, four beams, two walls, and one slab. It was found that all connections of the two models were not connected including beam-column, wall-column, wall-beam, and slab-beam connections (Fig. 3). In Fig. 3, blue lines represent beams and columns and red squares represent the wall and slab. It is clear that none of the elements are connected; therefore, structural analysis software will not be able to analyze the model as is and it will be difficult to solve the connection issue for complicated models, i.e., modeling from scratch would take less effort than modifying such transferred model with missing information. In addition to the connection issue, IFC export does not cover the structural behavior of the members such as cross-sectional and material mechanical properties. Material mechanical properties will therefore need to be checked and added using the expected algorithm.

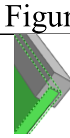
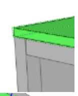
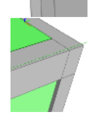
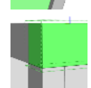
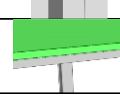


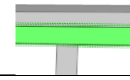
Beam-column frame   Beam-column-wall-slab frame

Fig. 3. IFC models for case study in SAP2000

In addition to the issues in connectivity and material properties, overlapping of elements occurred during the exportation to IFC. Different scenarios of nesting and overlapping may occur because there is no universal method among architects when creating their models. Therefore, in this study most common scenarios (Table 3) are tested to validate the capability of invariant signature in solving nesting and overlapping issues.

Table 3. Common nesting/overlapping scenarios

Scenarios	Description	Figures
Scenario 1	Wall overlapped with beams and columns	
Scenario 2	Slab nested in beams and columns	
Scenario 3	No overlapping or nesting	
Scenario 4	Beam nested in column	
Scenario 5	Continuous slab (require virtual cut)	

Scenario 6	Continuous beam (require virtual cut)	

Data Collection/Creation

Architectural software Autodesk Revit (Autodesk 2019) was utilized to develop three-dimensional models and then IFC coordination view 2.0 (CV 2.0) of the models was exported. Since IFC CV 2.0 is not directly applicable for the structural analysis software used in this study, mapping algorithm based on the invariant signatures was developed and utilized to extract necessary information and convert them to STAAD (Bentley Company 2019) input text format. The authors chose STAAD structural analysis software because it allows the use of nonproprietary format (i.e., textual data) as input.

To support the development and testing of the processing algorithms based on invariant signatures, the authors created five models for developing (i.e., training) and four models for testing the algorithms. Similar structural models with different dimensions and element overlapping scenarios are used for the training and the testing models.

In the training data, the models consist of AEC objects for reinforced concrete beams, columns, slabs, and walls. Each AEC object will be used to generate one set of invariant signatures including both geometric and material signatures to represent that object (e.g., a beam can be represented by 7 geometric signatures and 9 material signatures specified in Table 6 and Table 7). The five training models are shown in Fig. 4 and their cross-sectional dimensions are summarized in Table 4. The training models include: (1) model 1 has only beams and columns, (2) model 2 consists of beams, columns, walls and slab in which slab sits on the beams and columns, (3) model 3 has the same configuration as model 2 except for the overlapping of walls with beam and columns, (4) model 4 has the same dimensions as models 2 and 3 without element overlapping (i.e., slab and walls are located at the face of beams and columns), and (5) model 5 is a one-bay by two-bay buildings with different cross-sectional dimensions.

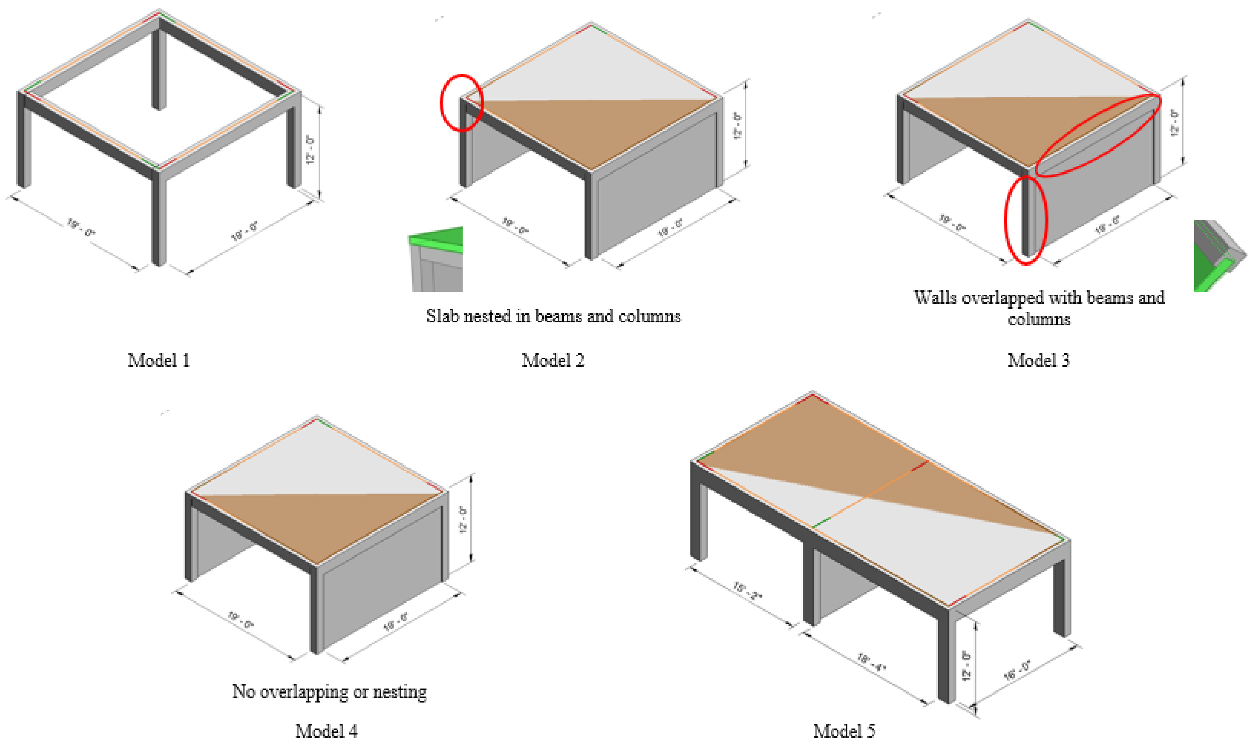


Fig. 4. Visualizations of the training models

Table 4. Cross-sectional dimensions of training models

Models	Beams, m (ft)	Columns, m (ft)	Shear walls, m (ft)	Slabs, m (ft)
1	0.46 (1.5) x 0.31 (1)	0.31 (1) x 0.31 (1)	n/a	n/a
2	0.46 (1.5) x 0.31 (1)	0.31 (1) x 0.31 (1)	0.15 (0.5)	0.15 (0.5)
3	0.46 (1.5) x 0.31 (1)	0.31 (1) x 0.31 (1)	0.15 (0.5)	0.15 (0.5)
4	0.46 (1.5) x 0.31 (1)	0.31 (1) x 0.31 (1)	0.15 (0.5)	0.15 (0.5)
5	0.61 (2) x 0.31 (1)	0.46 (1.5) x 0.31 (1)	0.20 (0.67)	0.20 (0.67)

Similar to the training models, the first four testing models are reinforced concrete one-story buildings including 2 models of a one-bay building, 1 model of a two-bay building, and 1 model of a three-bay building that has different structural dimensions and number of elements. Fig. 5. shows the visualizations of these four testing models. The cross-sectional dimensions of these models are summarized in Table 5. The fifth testing model is based on a real project and will be introduced in the evaluation section.

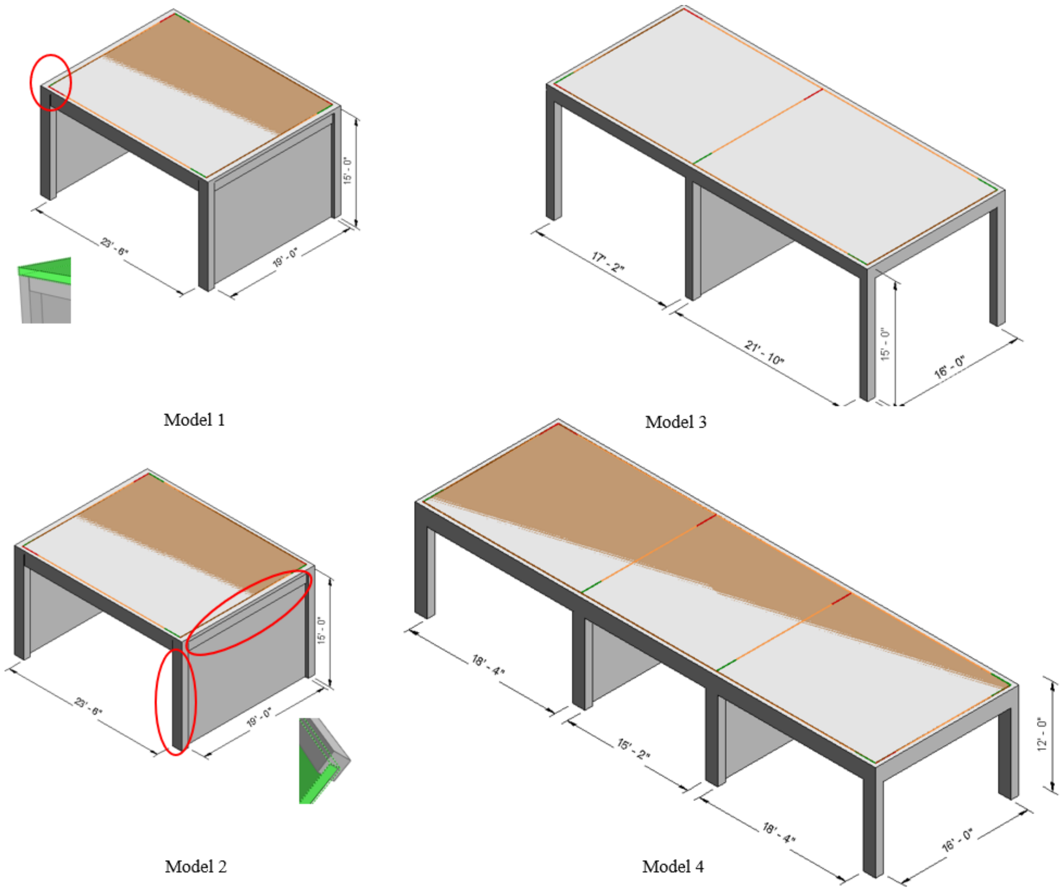


Fig. 5. Visualizations of testing models

Table 5. Cross-sectional dimensions of testing models

Models	Beams, m (ft)	Columns, m (ft)	Shear walls, m (ft)	Slabs, m (ft)
1	0.61 (2) x 0.31 (1)	0.46 (1.5) x 0.31 (1)	0.20 (0.67)	0.20 (0.67)
2	0.61 (2) x 0.31 (1)	0.46 (1.5) x 0.31 (1)	0.20 (0.67)	0.20 (0.67)
3	0.46 (1.5) x 0.31 (1)	0.31 (1) x 0.31 (1)	0.15 (0.5)	0.15 (0.5)
4	0.61 (2) x 0.31 (1)	0.46 (1.5) x 0.31 (1)	0.20 (0.67)	0.20 (0.67)

Preprocessing

For all the models, their corresponding STAAD input files for structural analysis were manually created to form the gold standard. In these STAAD input files (i.e., the gold standard), there are different sections such as header, joint coordinates, member incidences, material properties, and member property, which provide the information corresponding to the geometric and material signatures of the building models. Fig. 6 shows an example STAAD input file. The header section includes general project information such as project name, date, and unit of measurement. The joint coordinates section defines coordinates of joints, that are used to depict connections between structural elements. For example, the first two coordinates in Fig. 6 (i.e., 1st 0 0 0 and 2nd 0 12 0) represent the starting and ending points of one column. Number 1 means the first joint whose x, y, and z coordinates are 0, 0, and 0, respectively. The member incidences section defines structural members based on coordinate labels. For example, 1 1 2 means structural element member 1 is connected by joints 1 and 2. The material section depicts material property values of the selected material for the building model. Finally, the member property section defines the cross-sectional properties of structural members. For example, “1 3 8 10 PRIS YD 1 ZD 1” means members 1, 3, 8, and 10 have a prismatic shape for their cross-sectional dimensions with a dimension of 1 foot in both y- and z-directions. The developed algorithm needs to output similar STAAD input files, with correct values according to the architectural models.

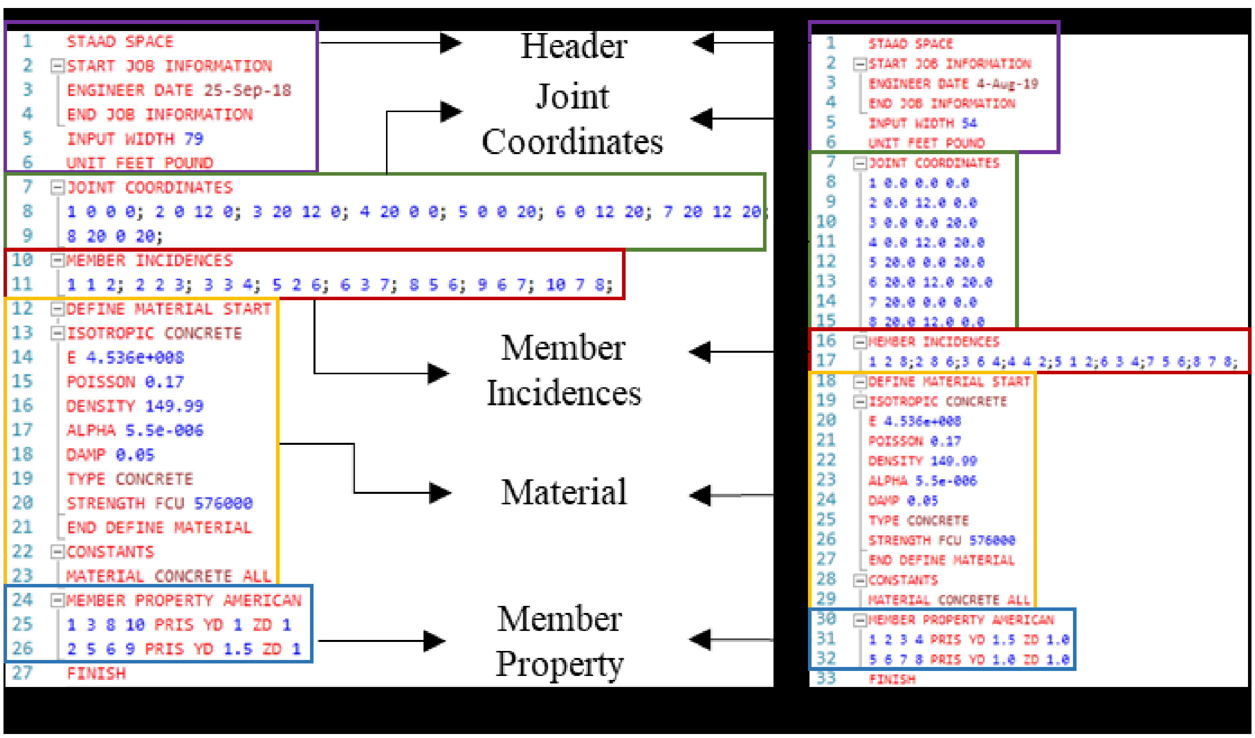


Fig. 6. STAAD input: a. Gold standard. b. Results of developed algorithm

Invariant Signature Application

To accomplish the goal of generating STAAD input files, the authors conducted the feature extraction, developed the processing algorithm that consists of information validation and mapping sub-algorithms, and recursively tested and modified the algorithm for structural analysis application.

Feature Extraction.

The shape properties include the built-in values extracted from the IFC object files, such as the length and width values from *IfcRectangleProfileDef*, and generated values such as the bounding box values, which were extracted from all possible shapes. In IFC, the locational information is represented by an axis that includes an origin and two perpendicular vectors representing the x, z axes. The y-axis is the normalized cross product of the x and z axes, which can be derived from the given axes. An object may have multiple axes convolved with each other, i.e., an axis within another axis, for several layers. The authors proposed to convert all axes into a final axis, to ease the analysis of the locational information. For material properties, the authors generated and analyzed material parameters (e.g., Mass Density and Young’s Modulus) based on different material types (i.e., steel, concrete and wood), and developed an input validation module that can check material information and leverage it for further analysis.

The authors used a data-driven approach to develop processing sub-algorithms to check and extract the geometric and material properties that include all distinguishing geometric and material information of an AEC object. Table 6 shows all the needed properties related to the geometric signatures, and Table 7 shows all the needed properties related to the material signatures.

Table 6. Properties of Geometric Signatures.

Signature Name	Signature Type	Value Type	Feature Description
X-dim	Shape	Numerical	X dimension of bounding box
Y-dim	Shape	Numerical	Y dimension of bounding box
Z-dim (Depth)	Shape	Numerical	Extruded Depth
Extruded_direction	Shape	3D Vector	Extruded directions (Z-dim)
Origin	Locational	3D Vector	The origin of the placement
x-axis	Locational	3D Vector	Vector of x-axis
z-axis	Locational	3D Vector	Vector of z-axis

Table 7. Properties of Material Signatures.

Material Property	IFC Entity
Density	*IfcMassDensityMeasure*
Young's Modulus	*IfcModulusOfElasticityMeasure*
Shear Modulus	*IfcShearModulusMeasure*
Poisson's Ratio	*IfcPositiveRatioMeasure = IfcRatioMeasure*
Thermal Expansion Coefficient	*IfcThermalExpansionCoefficientMeasure*
Ultimate Stress	*IfcPressureMeasure*
Yield Stress	*IfcPressureMeasure*
Compressive Strength	*IfcPressureMeasure*

Iterative Algorithm Development

The authors used an iterative approach to develop algorithms to process an IFC model input. The overall flow is shown in Fig. 7. The iterative algorithm development starts with the development of the sub-algorithms for extracting invariant signatures from all training models. Then the extracted features are verified by comparing to the original models. The extraction sub-algorithms are modified until all information is correctly extracted and verified. After verifying all the features of invariant signatures, a validating process is conducted to check if any required information is missing. In this validating process, all missing information based on geometric and material signatures are solicited until all required information is acquired. After obtaining all the required inputs, the information mapping sub-algorithms are iteratively developed until they can successfully generate the correct results, which is based on the comparison with the predefined gold standards. In the end, all the sub-algorithms are compiled together into one processing algorithm.

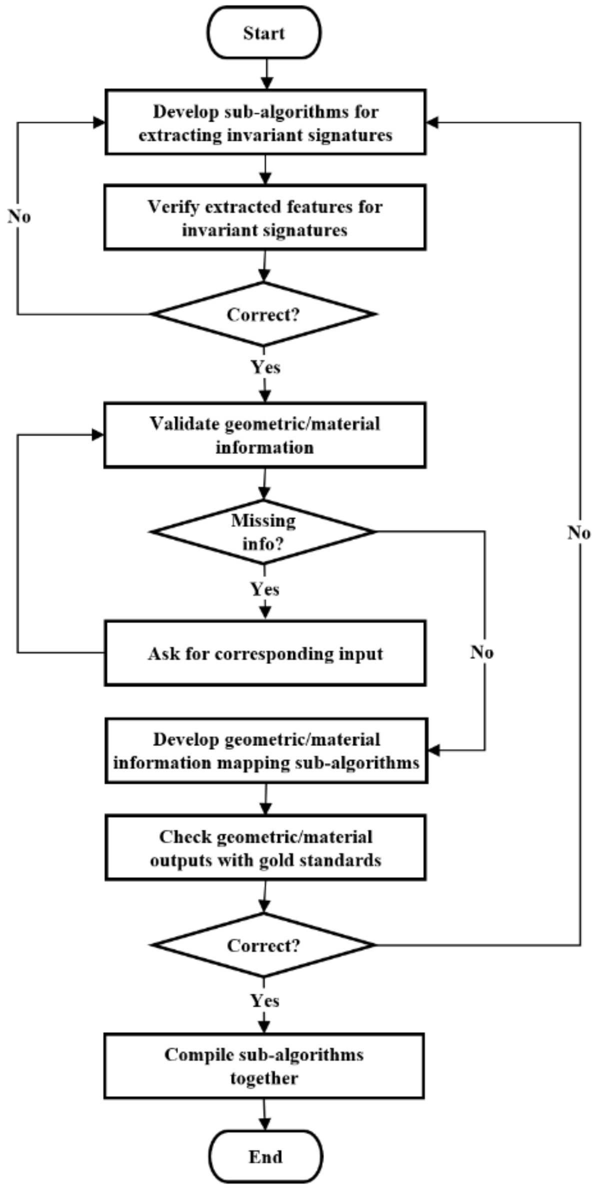


Fig. 7. Iterative Algorithm Development

Develop Sub-algorithms for Extracting Invariant Signatures & Verify Extracted Features for Invariant Signatures. Started with the sub-algorithm development for extracting invariant signatures, an iterative development approach was followed until the extracted information are verified to be correct. For geometric information, both shape and locational information were extracted. All shapes in the training data were represented by either *IfcExtrudedAreaSolid* or *IfcFacetedBrep*. For *IfcExtrudedAreaSolid,* objects were represented by extending a 2D plane.

Regular objects can be represented using built-in 2D rectangular shapes. In this case, *X-dim* and *Y-dim* can be extracted easily because the bounding box of a rectangular cuboid is the cuboid itself. *Z-dim* (depth) and *extruded_direction* were extracted directly from the *IfcExtrudedAreaSolid.* Although all shapes in the data were represented by a rectangular cuboid shape, the actual implementation of the object instances may not be regular because of the overlapping of objects (Table 3), where the boundary representation (*IfcFacetedBrep*) was used. The dimensions were calculated by taking the differences between the maximum and minimum values in the *X, Y,* and *Z* dimensions. *Extruded_direction* was set vertical, i.e., using a vector of <0, 0, 1>.

For locational information, by observing the data and documentation, the authors developed the sub-algorithm to calculate a resulting absolute placement from all relative placement (*IfcAxis2Placement3D*) instances associated with a given object (Fig. 8). The placement (both relative and absolute) of an object is represented by two orientation vectors (*x-vector* and *z-vector*) and a placement origin (point o). A vector or a point is represented with three values *x, y, z*, representing three dimensions. In IFC the placement of an object is usually defined in reference to the placement of another object, which forms relative placement. An absolute placement (i.e., in reference to the world coordinate) may represent the same information as multiple layers of relative placements. Using the algorithm in Fig. 8, the authors were able to calculate an absolute placement *a* [(i.e., represented as new origin and axes (x-axis and z-axis)] for a series of multiple layers of relative placements by iteratively performing the merging until all placements in the list merge into one.

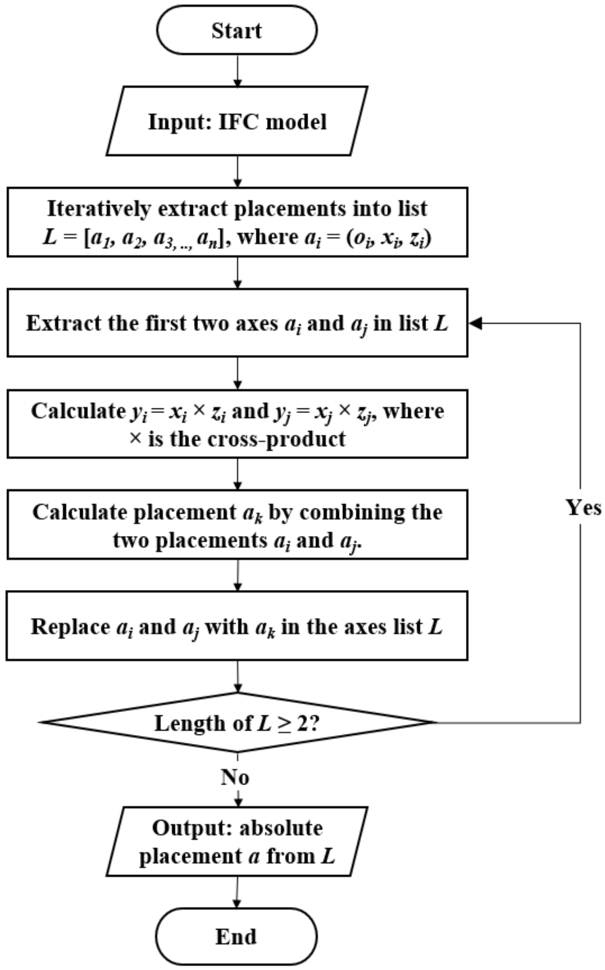


Fig. 8. Calculating an absolute placement that combine all relative placements

Material information was analyzed and generated based on the proposed invariant signatures' concepts for different types of materials under different analysis scenarios. For example, for concrete material, six required parameters are checked in the algorithm as follows: *MassDensity, PoissonRatio, ShearModulus, ThermalExpansionCoefficient, CompressiveStrength,* and *YoungsModulus*. During the process of analyzing the training data, all the material information was found missing, it was therefore solicited to be input manually by the information validation sub-algorithm (see details in the next section). The manual material information input was further used in the information mapping.

Validate Geometric/Material Information. After feature verification, the authors validated the extracted information to check if additional input were needed.

For geometric information, all needed feature values were successfully extracted. So, there was no need for additional information beyond those extracted from the IFC model file. However, the material information was missing in the information process flow from the architectural model to the IFC model file, the authors developed a customized algorithm to check material information and solicit user input for each parameter (e.g., mass density and Young's modulus) with pre-defined unit information. In this validation step, all the existing information and the user input information was stored for further mapping process, when all the material information (i.e., material types, existing and user input information) would be mapped directly to STAAD input file. For example, the sub-algorithm required manual input numerical number for a specific parameter (e.g., compressive strength) by prompting the following sentence in the system: "Please input Compressive Strength of the material (Unit:Psi)". The collected value would then be mapped into STAAD input as the following: "STRENGTH FCU XX", where XX represents the number collected. Based on the proposed invariant signatures' concept, the authors developed invariant material signatures for three main types of materials – steel, concrete, and wood. Table 7 lists the detailed properties in the invariant material signatures and their corresponding entities in an IFC file. Detailed material properties in the IFC files are defined by different IFC entities.

Develop Geometric/Material Information Mapping Sub-algorithms. After validating the required information in invariant signatures, the authors developed information mapping sub-algorithms that convert the invariant signatures to corresponding STAAD input files for each model. Fig. 9 illustrates the developed sub-algorithm for geometric information mapping.

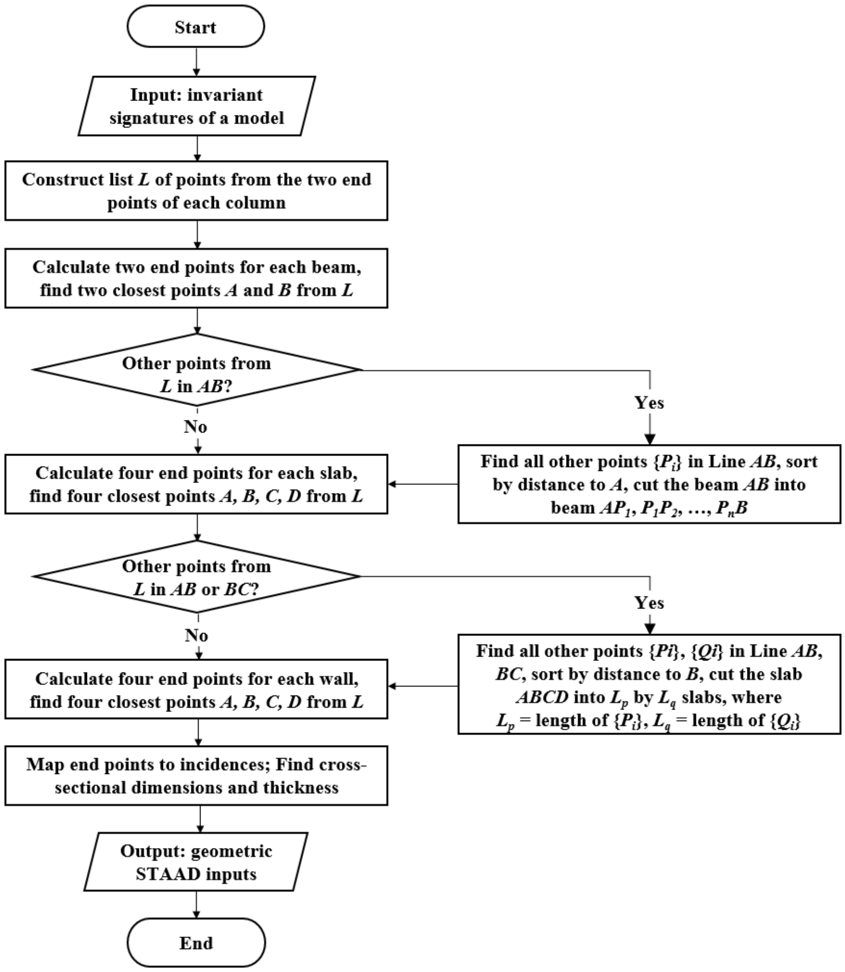


Fig. 9. Geometric information mapping sub-algorithm

The generation of the STAAD inputs started from processing all the columns. Based on the required format of the STAAD input file that a column object is represented by two points (Fig. 10a), which was defined as the centers of the top and bottom surfaces as the two ends of a straight-line segment. The length (YD) and width (ZD) information are generated by *X-dim* and *Y-dim* from the invariant signatures.

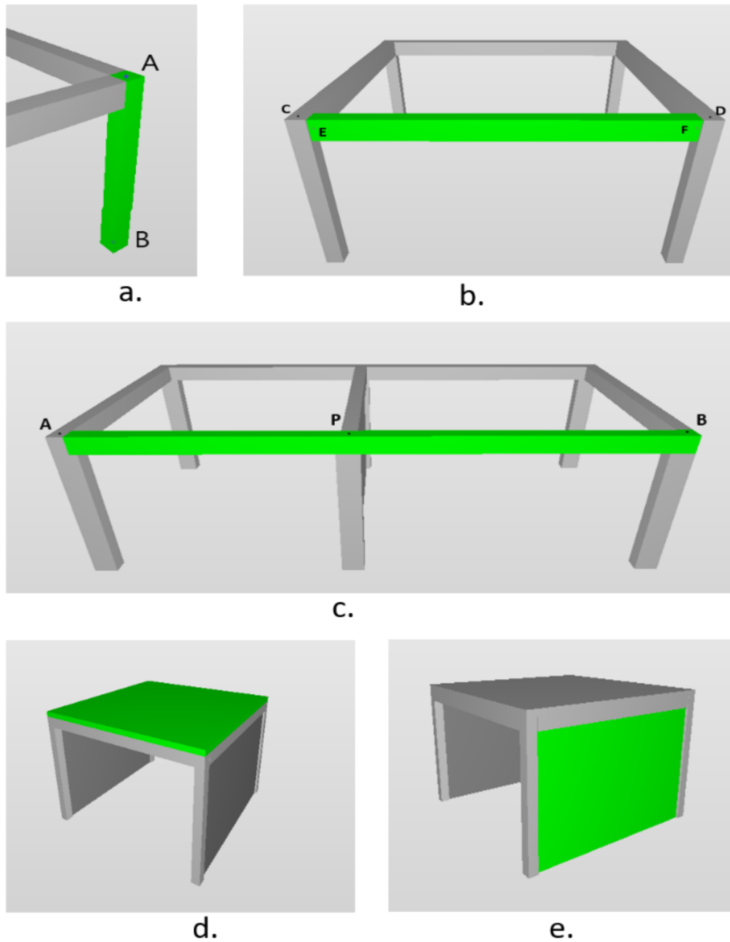


Fig. 10. Column, beam, slab, and wall examples

To determine the coordinates of the two end points of a column (e.g., point *A* and *B* in Fig. 10a) from invariant signatures, the authors observed that *B* is the origin, and *A* is calculated using Equation (1).

$$(x, y, z) = (\sigma_x, \sigma_y, \sigma_z) + (d_x, d_y, d_z) * d \qquad (1)$$

Where *(x, y, z)* is the coordinate of the end point (e.g., point *A* in Fig. 10a), (o_x, o_y, o_z) is the coordinate of the origin (e.g., point *B* in Figure 10a), (d_x, d_y, d_z) is the normalized extruded direction, and *d* is the extruded depth. The length and width are generated from invariant signatures to be *X-dim* and *Y-dim* or *Y-dim* and *X-dim*, based on the axis property. All end points of the columns form a list *L* that stores all the necessary points for defining all the objects.

For processing the beam instances, coordinates of the two end points were calculated from invariant signatures. Instead of directly using the coordinates of the two end points of the beam, the coordinates of the closest points of the connecting columns were used. Fig. 10b shows an example that instead of using points *E* and *F*, points *C* and *D* were used to determine the coordinates of the beam's two end points, which were the end points of the two connecting columns. The coordination of the two end points (*E* and *F*) was first calculated using Equation (1). Then the closest point of the connecting column is iteratively sought among all the column end points list *L* until the shortest Euclidian distance to point *E* or *F* is obtained. Similar to the column's dimensional definition, the length and width of a beam were generated based on *X-dim*, and *Y-dim* from invariant signatures. For the building model with multiple bays, a beam spanning two bays shall be represented by two beam members for the structural analysis purpose (Fig. 10c). This scenario was captured by checking if the line of the two end points of such a beam goes through another point in the *L* list. If another point *P* is in the line of *AB*, then this beam shall be split into two beams as *AP* and *PB* (Fig. 10c).

For mapping slab information, the STAAD input file requires four end points and a numeric value for its thickness. Fig. 10d shows an example of a slab. The coordinates of the end points were calculated using Equation (2), where *origin.x*, *origin.y*, and *origin.z* represent the three coordinate values of the origin. Similar to beams, slabs may be divided into multiple slabs if the edge of a slab passes through a point in the *L* list.

$$\text{Four points} = \left(origin.x \pm \frac{X\_dim}{2}, origin.y \pm \frac{Y\_dim}{2}, origin.z\right) \qquad (2)$$

Mapping wall information was more straightforward compared to slabs, as a wall does not need to be divided into multiple walls in the multi-bay building model. The coordinates of the four end points of a wall (Fig. 10 (e)) are calculated by Equations (3) and (4).

$$\text{Point 1 and 2} = (origin.x \pm max(X\_dim, Y\_dim)\,, origin.y, origin.z) \quad (3)$$

$$\text{Point 3 and 4} = (origin.x \pm max(X\_dim, Y\_dim)\,, origin.y, origin.z + extruded\_depth) \quad (4)$$

Based on the invariant signatures, the authors were able to identify the four points used to define a slab/wall. Then the coordinates of the columns closest to these four points were utilized to map the information of the end points of the slab/wall. The thickness was subsequently determined as the *Z-dim* for a slab object, and the minimum of *X-dim* or *Y-dim* for a wall object.

The developed material information mapping sub-algorithm includes three main steps: Step 1- check material type, during which the material name would be checked and picked up based on the "Name" attribute of the *IfcMaterial* instance in an IFC model. Step 2 - check material information or solicit user input. In this step, the customized algorithm checked the existence of the required material parameters. If they were not found in the IFC model (i.e., the name and nominal value attributes of the *IfcPropertySignleValue* instance did not exist), the algorithm solicited users to manually input numerical values of the material properties based on the predefined unit of each parameter. If the numerical values of the material properties were found in the IFC file, they were used directly in Step 3 for mapping. Step 3 - map information to the STAAD input file. In this step, the corresponding material information in the STAAD input file would be generated based on the IFC model and the user inputs. The algorithm was able to generate correct STAAD input files for all the four training models. Fig. 11 shows partially the flow of the material processing sub-algorithm.

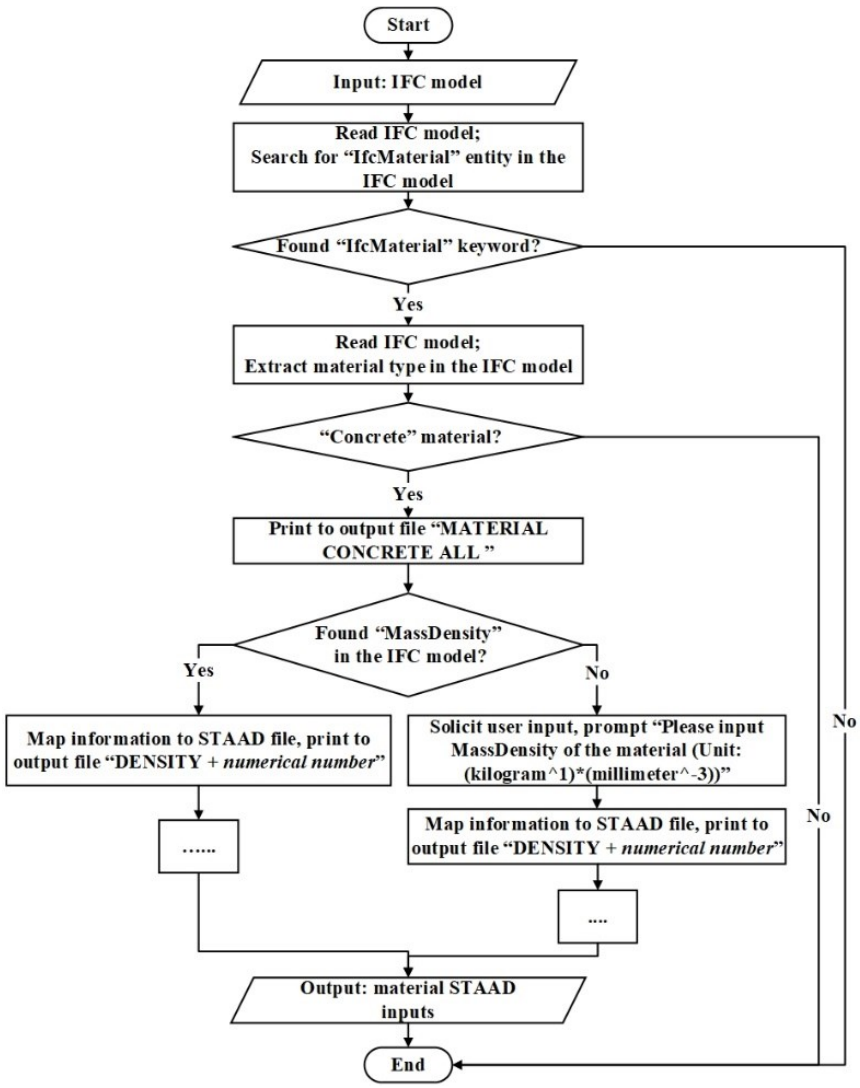


Fig. 11. Material processing sub-algorithm flow diagram

Check Geometric/Material Outputs with Gold Standards. To check the processing outputs of all training models, the outputs were compared to the gold standard (i.e., the STAAD input files of the models). Sample STAAD inputs for the one bay beam-column frame developed based on invariant signatures and based on the gold standard are shown in Fig. 6 (a) and (b), respectively. Both input files gave exactly the same structural model in element connectivity, cross-sectional dimensions and material properties. To obtain a quantitative measure of the algorithm’s performance, the authors implemented metrics in describing the accuracy of each section of header, joint coordinates, member (column, beam, slab, wall) incidences, member property, material properties, and overall (Table 8). In the comparison of the joint coordinates with the gold standard, the order of the coordinates does not matter. However, they must contain the same set of coordinates. For member incidences, the comparison focused on if every object was represented with the correct ending points (incidences). Even though the data from the gold standard and from the algorithm output may appear to be different, they were considered to match if the difference was solely due to different sequences of the same incidences. In effect, the incidences define the joints connectivity based on which the length of beams, height of columns, length and width of slabs, and length and height of walls can be checked by taking the distances of the corresponding points. For beams and columns, cross-sectional dimensions were checked by YD, and ZD of MEMBER PROPERTY AMERICAN (line 24 in Fig. 6 (a) and line 30 in Fig. 6(b)). For slabs and walls, the thickness was checked. The data evaluation results using the proposed metrics are summarized in Table 8, in which both the geometric and material information were generated correctly by the developed algorithm, with a 100% accuracy.

Table 8. Algorithm evaluation results on training data (correct number of instances/total number of instances)

Section	Subsection	1	2	3	4	5
Header		6/6	6/6	6/6	6/6	6/6
Joint Coordinates		8/8	8/8	8/8	8/8	12/12
Member Incidences	Column	4/4	4/4	4/4	4/4	12/12
	Beam	4/4	4/4	4/4	4/4	12/12
	Slab	NA	1/1	1/1	1/1	2/2
	Wall	NA	2/2	2/2	2/2	1/1
Member Property	Column Dimensions	8/8	8/8	8/8	8/8	6/6
	Beam Dimensions	8/8	8/8	8/8	8/8	6/6
	Slab Thickness	NA	1/1	1/1	1/1	2/2
	Wall Thickness	NA	2/2	2/2	2/2	1/1
Material Information		8/8	8/8	8/8	8/8	8/8
Total		*46/46*	*52/52*	*52/52*	*52/52*	*78/78*

The developed algorithm was also designed to overcome the overlapping issues. The tested scenarios are: i. beams sitting on columns, ii. columns passing through beams, iii. walls cutting beams and columns, iv. slab and walls at the face of beams and columns, v. continuous slab that requires virtual cut, and vi, continuous beam that requires virtual cut. According to the results in Table 8, the developed algorithm successfully solved the overlapping issue in the training models.

Compile Sub-Algorithms Together. After checking the correctness, all the developed sub-algorithms were combined into one algorithm that can take an IFC file as an input and output a valid STAAD file to be imported by the software for structural analysis.

Evaluation

The algorithm developed based on the training data was then tested on the four models in the testing dataset which had different dimensions of structural configurations and members with the training models as explained previously. Similar to the training models, results of all the testing models using the developed algorithm were compared to the gold standard STAAD input files and evaluated in accuracy. According to the evaluation results (Table 9), a 100% accuracy was achieved demonstrating that invariant signature was capable of supporting the extraction of information required for conducting structural analysis including geometry, element connectivity, cross-sectional dimensions, and material mechanical properties from an IFC CV 2.0 model file created by the architect software.

Table 9. Algorithm evaluation results on testing data (correct number of instances/total number of instances)

Section	Subsection	1	2	3	4
Header		6/6	6/6	6/6	6/6
Joint Coordinates		8/8	8/8	8/8	8/8
Member Incidences	Column	4/4	4/4	6/6	8/8
	Beam	4/4	4/4	6/6	8/8
	Slab	1/1	1/1	2/2	3/3
	Wall	2/2	2/2	1/1	2/2
Member Property	Column Dimensions	8/8	8/8	12/12	16/16
	Beam Dimensions	8/8	8/8	12/12	16/16
	Slab Thickness	1/1	1/1	2/2	3/3
	Wall Thickness	2/2	2/2	1/1	2/2
Material Information		8/8	8/8	8/8	8/8
Total		*52/52*	*52/52*	*78/78*	*88/88*

The developed algorithm was further evaluated by importing the STAAD input files of the five training and the four testing models, which consisted of the three major structural properties (i.e., element connections, model geometry including frame and cross-sectional dimensions, and material properties), into the software. The resulting structural models visualized by STAAD of the four testing models are shown in Fig. 12 as examples to demonstrate the model dimensional information were successfully generated and correctly mapped using the algorithm.

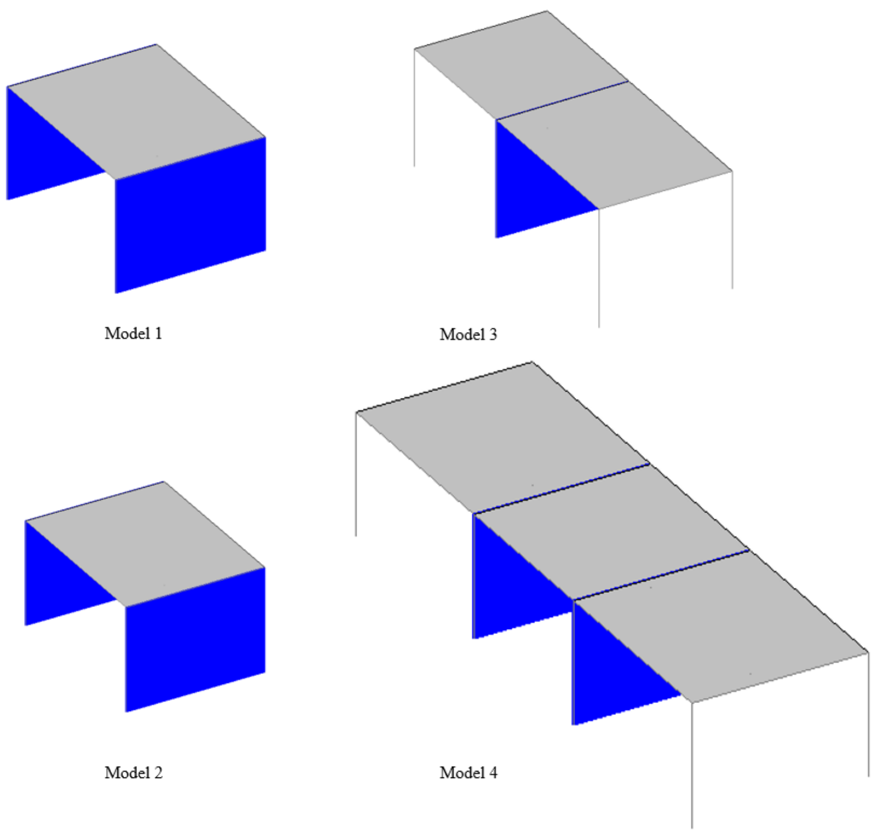


Fig. 12. STAAD visualization of the processed testing models

In addition, the processes of converting architectural model to structural model using the existing method and the proposed invariant signature-based method are compared and illustrated in Fig. 13. As the current version of STAAD does not support both IFC input and output yet, other software such as SAP2000 is needed to convert the IFC CV file generated based on the architectural model into an intermediate CIS/2 file that can be imported by STAAD. It can be observed from Fig. 13(a) that elements that are not connected and plate elements (i.e., wall and slab) are missing in the STAAD model generated using this method. Also, this method does not support correct conversion of the cross-sectional properties and material information. In comparison, the information conversion process using the proposed invariant signatures does not require third party software (e.g., SAP2000) to convert the IFC file to the STAAD input file. Both geometric and material information were correctly extracted and mapped as demonstrated above.

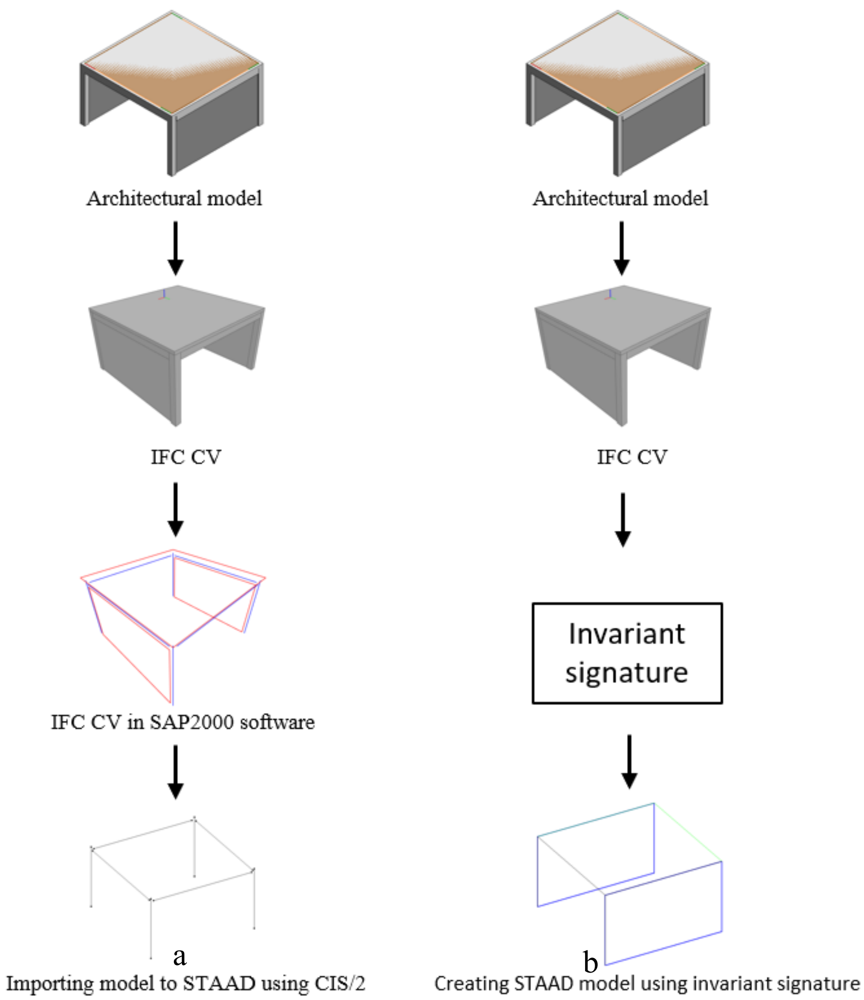


Fig. 13. Converting architectural model to analytical structural model processes: a. existing method for converting architectural model to STAAD model. b. Converting architectural model to STAAD using invariant signatures

In addition to the four simple testing models, the authors also tested their algorithms on the model of a real building project. This is a two-story penthouse RC commercial building. The building is

31.70 meters long and 21.60 meters wide and it was constructed for a medium size grocery store. Due to the topography of the land, the first story was used as storage (semi-basement) and the second story was the main store. Fig. 14 shows the visualization of the architectural design and Fig. 15 shows the structural view. The authors applied their developed algorithm to this building and were able to generate the results shown in Table 10.

Table 10. Algorithm evaluation results on a real project (number of instances)

Section	Subsection	Generated	Correctly Generated	Total	Precision	Recall
Header		6	6	6	100%	100%
Joint Coordinates		64	64	64	100%	100%
Member Incidences	Column	44	44	44	100%	100%
	Beam	65	60	68	88.2%	92.3%
	Slab	25	23	30	83.3%	92.0%
	Wall	NA	NA	NA		
Member Property	Column Dimensions	72	72	80	90%	100%
	Beam Dimensions	128	128	128	100%	100%
	Slab Thickness	25	25	25	100%	100%
	Wall Thickness	NA	NA	NA		
Material Information		8	8/8	8/8	100%	100%
Total		*437*	*430*	*453*	*94.9%*	*98.4%*

The results showed a 94.9% precision and 98.4% recall. The authors conducted an error analysis on the incorrect instances. The causes of the errors include: a). There were 4 beams that use surface model (similar to Brep but different in the IFC representation) in the real model, where there were no such representations in the training data. So, the invariant signatures were not correctly generated. This was fixed by adding the processing of such new representations. b). The algorithm generated 5 repeated beams existing in the model, which was caused by erroneous exporting into IFC. However, this was detected and fixed by checking if the beam index already existed. c). Two slabs were not generated correctly because of their stair openings (Fig. 16), i.e., the algorithm generated the larger size of slabs because the algorithm was not trained to solve openings based on training data. However, this can be addressed easily by finding the corresponding slab and cut the slab with the opening to find the correct dimension. d). Column dimensions were not correct for the eight cylindrical columns. This can be easily fixed with the addition of radius on the dimensions. e). Four beams and five slabs were not successfully processed as they exceeded the main frame boundaries (Fig. 17). This error occurred because there were no similar situations in the training data. To fix this, the algorithms can be extended to include the addition of the objects that exceed the boundaries. For example, the algorithm can check if the ending point exceeds the boundary of the columns. If so, new end points will be added and the same algorithm will be able to generate the desired results based on the new points.

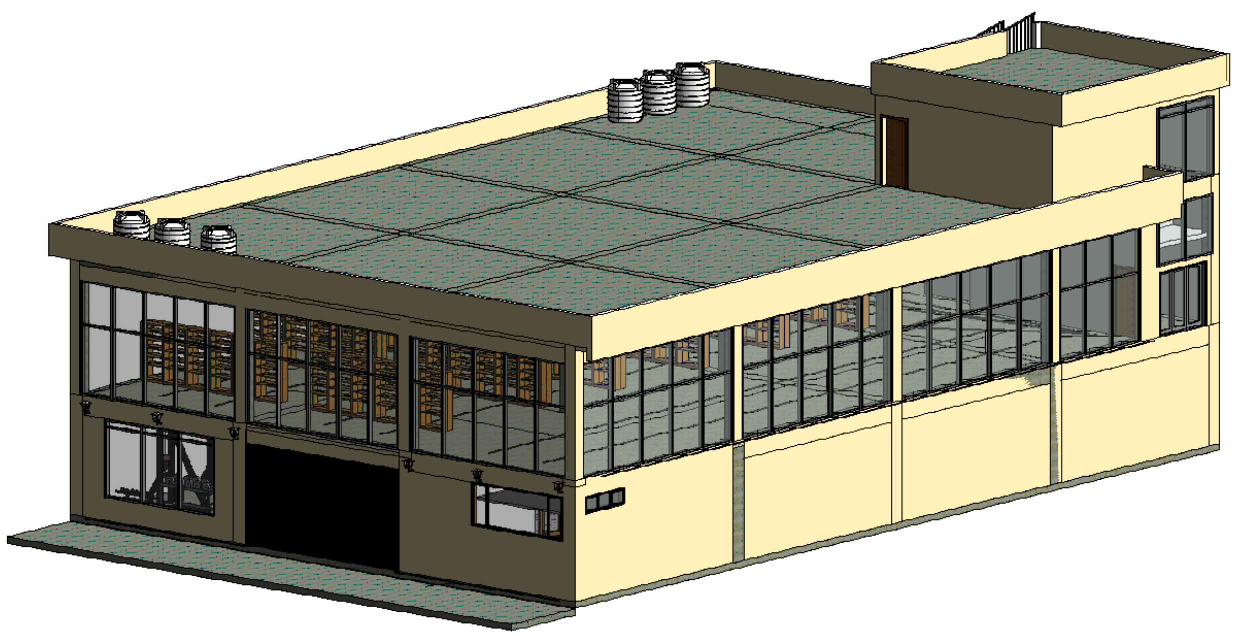


Fig. 14. Architectural view of a real project for testing

Fig. 15. Visualization of the structural view of the real project for testing

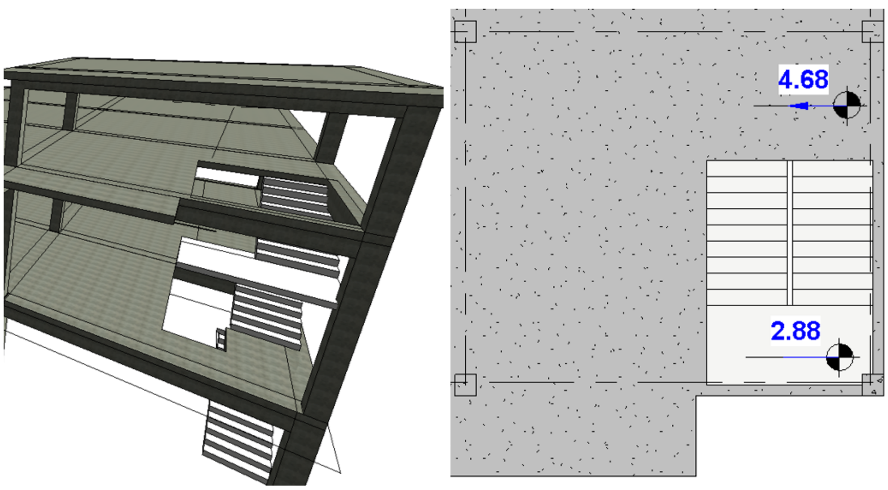


Fig. 16. The incorrectly generated slabs because of the stair opening

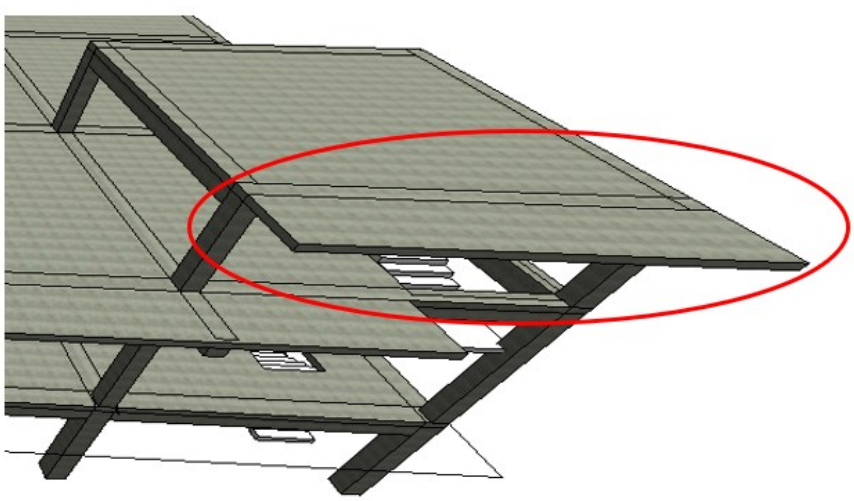


Fig. 17. An example of slabs that goes beyond the column boundary

In summary, the algorithms generated a 94.9% precision and 98.4% recall in the resulting information for a real project model of which the object representation was quite different from the ones in training data, which showed the robustness of the developed algorithm and the method. It also showed that invariant signatures can capture the scientific nature of an AEC objects. Invariant signatures can be adopted in structure analysis with high accuracy.

Discussion

In this paper, the authors implemented the invariant signatures using IFC models. IFC models can vary based on BIM software and schemas. Such variations can be addressed by standard frameworks and tools such as model view definitions (MVDs), which provide a construct for IFC implementation and a standard schema for defining required/optional information in an IFC model. Although there are existing MVD checking tools to help check information of an IFC model, there are limitations in their capabilities. For example, the attribute values of entities could not be directly checked by IfcDoc, the most commonly used MVD checking tool. In contrast, the research reported in this paper was able to and focused on such detailed information (e.g., attribute values of IFC entity instances) to implement the invariant signatures. So the state of the art MVD checking tool (i.e., IfcDoc) is not sufficient to support our proposed method in this paper. In addition, IFC is a comprehensive international standard targeting at the life cycle information support for AEC project. While there are more focused data standard for structural analysis support such as CIS/2, such standard was designed to transfer data between structural software, which is different from our research objective of addressing interoperability between architectural and structural software. Moreover, the authors' proposed method based on invariant signatures can be used for developing and exchanging information between BIM applications for different tasks.

Conclusions

The Model exchange between architectural software and structural analysis software is not fully interoperable yet. There is a lack of foundational methods that enable a seamless BIM interoperability between architectural design and structural analysis. To address this gap and facilitate future use of BIM for structural analysis, the authors developed invariant signatures for architecture, engineering, and construction (AEC) objects and proposed a new data-driven method to use invariant signatures for solving practical problems in BIM applications. The invariant signatures include the inherent geometric and material properties of AEC objects. An experiment was conducted to develop an information validation and mapping algorithm based on five training models, which was then tested on four testing models. Results showed that the developed algorithm successfully generated structural analysis input files for all the four testing models. Comparing to an existing model conversion workflow, the proposed method achieved better accuracy. The developed algorithm was shown to achieve perfect accuracy on simple testing models and high accuracy in real project models. The algorithms following the method can be improved to 100% accuracy based on more training cases. Because there are an enumerable number of patterns for building objects, the proposed method can be used to accumulatively grow to eventually include all patterns to generate a robust algorithm for supporting structural analysis on all buildings. Invariant signatures were therefore demonstrated to support BIM interoperability between architectural design and structural analysis. In addition, the developed algorithm demonstrated the potential for it to be used in more complex AEC models.

Contributions to the Body of Knowledge

This research contributes to the body of knowledge in three main ways. First, on a theoretical level, it offers an invariant signature theory with initial geometric and material signatures that can be used to capture the essence of architectural, engineering, and construction objects to be used in various engineering analyses and model exchange tasks. This theory goes beyond the state of the art of AEC information representation based on data schema, and enables robust AEC information representation ignorant to software implementation, modeling decisions, and/or language/culture contexts. Second, on a practical level, this research advances the empirical knowledge in the area of BIM interoperability by offering a new data-driven method to use invariant signatures for solving practical problems in BIM applications. This method supports the many-to-one paradigm of BIM interoperability and brings seamless and universal BIM interoperability one step closer to reality. Third, on an empirical level, this theoretical and empirical knowledge in the area of BIM interoperability can support an improved model exchange between different AEC processes such as between architectural design and structural analysis, between architectural design and cost analysis, and between architectural design and energy consumption analysis.

Limitations and Future Work

The authors acknowledge two main limitations of this study as follows. First, the invariant signatures in this research were only tested on regular-shaped buildings and members. For example, it was not tested for irregular shapes. Second, the experiment only used one structural analysis software. Therefore, in their future research, the authors plan to expand algorithm development to cover more complicated structures such as high-rise buildings, incorporate curved and irregular-shaped building elements into consideration, and test the invariant signatures on more software platforms.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grants No. 1745374 and 1745378. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

References

Aldegeily, M., Zhang, J., Hu, Y., and Shao, X. (2018). "From architectural design to structural analysis: a data-driven approach to study building information modeling (BIM) interoperability." *Proc., 54th ASC Annual Intl. Conf.*, ASC, Fort Collins, CO, 537-545.

Autodesk Revit Software. (2019). Autodesk <https://www.autodesk.com/products/revit/overview> (Nov. 2019).

Bank, L. C., McCarthy, M., Thompson, B. P., and Menassa, C. C. (2010). "Integrating BIM with system dynamics as a decision-making framework for sustainable building design and operation." *Proc., the First Intl. Conf. on Sustainable Urbanization* (ICSU), Hong Kong Polytechnic University, Hong Kong, China.

Brandman, T. and Peelen, M. (2017). "Interaction between Scene and Object Processing Revealed by Human fMRI and MEG Decoding." *J. Neuroscience*., 37(32), 7700-7710.

BuildingSMART. (2017). "IFC2× Edition 3 Technical Corrigendum 1." < http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/> (Aug. 28, 2017).

Coolidge, J.L. (1902) “Euclid's elements.” *Bulletin of the American Mathematical Society*, 8(5), 216-220.

Daniyarova, E., Myasnikov, A., and Remeslennikov, V. (2012). “Algebraic geometry over algebraic structures. V. The case of arbitrary signature.” *Algebra and Logic*, 51(1), 28-40.

ETABS. (2019). Computers and Structures. Inc. < https://www.csiamerica.com/categories/etabs> (Nov. 2019).

Fleming, W. S. (2016). BIM modelling for structural analysis. Master thesis: Master of Technology: Structural Engineering, Faculty of Civil and Environmental Engineering, Poznan University.

Grilo, A., and Jardim-Goncalves, R. (2010). “Value proposition on interoperability of BIM and collaborative working environments.” *Autom. in Constr.*, *19*(5), 522-530.

Hu Z. Z., Hu X. Y., Wang H. W., and Kassemb M. (2016). “Improving interoperability between architectural and structural design models: An industry foundation classes-based approach with web-based tools”. *Autom. in Constr.,* 66 (2016), 29–42.

Johnson, J. and Olshausen, B. (2003). “Timecourse of neural signatures of object recognition.” J. *Vision*, 3(7), 499-512.

Nawari, N. (2011). “Standardization of structural BIM.” *KProc. Comput. in Civ. Eng.* ASCE, Reston, VA.

Nelson, K. and Sokkappa, P. (2008). “A Statistical Model for Generating a Population of Unclassified Objects and Radiation Signatures Spanning Nuclear Threats.” *International Nuclear Information System*, 40(17).

McGraw-Hill Construction. (2014). "The business value of BIM for construction in major global markets: How contractors around the world are driving innovation with Building Information Modeling." Bedford, MA.

Muller, M. F., Garbers, A., Esmanioto, F., Huber, N., Loures, E. R., and Canciglieri, O. (2017). Data interoperability assessment though IFC for BIM in structural design–a five-year gap analysis. *J. Civ. Eng. and Mgmt.* 23(7), 943–954.

Ramaji, I. J., and Memari, A. M. (2018). "Interpretation of structural analytical models from the coordination view in building information models". *Autom. Constr.*, 90(June 2018), 117-133.

Ren, R., Zhang, J., and Dib, H. N. (2018). BIM Interoperability for Structural Analysis. In *Construction Research Congress 2018: Construction Information Technology*.

Ren, R., and Zhang, J. (2019). "Model Information Checking to Support Interoperable BIM Usage in Structural Analysis." *Inter. Conf. on Comput. in Civ. Eng. 2019*. ASCE, Reston, VA.

Sacks, R., Kaner, I., Eastman, C. M., and Jeong, Y. S. (2010). "The Rosewood experiment—Building information modeling and interoperability for architectural precast facades." *Autom. in Constr.*, 19(4), 419-432.

SAFE. (2019). SAFE Software. < https://www.safe.com/> (Nov. 2019).

Sanguinetti, P., Abdelmohsen, S., Lee, J., Lee, J., Sheward, H., and Eastman, C. (2012). "General system architecture for BIM: An integrated approach for design and analysis." *Adv. Eng. Info.*, 26(2), 317-333.

SAP2000. (2019). Computers and Structures. Inc. <https://www.csiamerica.com/products/sap2000> (Nov. 2019).

Solnosky, R. L., Memari, A. M., and Ramaji, I. J. (2014). "Structural BIM processes for modular multi-story buildings in design and construction." *Proc., 2nd Res. Bldg. Design & Constr. Conf.,* Penn State University, University Park, PA, 201-215.

Steel, J., Drogemuller, R., and Toth, B. (2012). "Model interoperability in building information modelling." *Software & Systems Modeling*, 11(1), 99-109.

Stow, D.A., Toure, S.I., Lippitt, C.D., Lippitt, C. L., and Lee, C. (2012). "Frequency distribution signatures and classification of within-object pixels." *Inter. J. Applied Earth Observations and Geoinformation*, 15, 49-56.

Shin, T. S. (2017). Building information modeling (BIM) collaboration from the structural engineering perspective. International Journal of Steel Structures, 17(1), 205-214.

STAAD (2019). Bentley Company. < https://www.bentley.com/en/products/brands/STAAD> (Nov. 2019).

Thein, V. (2010). "IFC Integrating building information modeling". Bentley Systems, Incorporated.

Tekla Structures. (2019). Trimble Company. <https://www.tekla.com/us/products/tekla-structures> (Nov. 2019).

Volk, R., Stengel, J., and Schultmann, F. (2014). "Building information modeling (BIM) for existing buildings — literature review and future needs." *Autom. in Constr.*, 38(2014), 109–127.

Watson, A. (2011). Digital buildings–Challenges and opportunities. *Advanced engineering informatics*, *25*(4), 573-581.

Wu, J., and Zhang, J. (2018). "Automated BIM object classification to support BIM interoperability." *Proceedings of the Construction Research Congress*, ASCE, Reston, VA, Wu, J., and Zhang, J. (2019a). “Introducing Geometric Signatures of Architecture, Engineering, and Construction Objects and a New BIM Dataset.” *Proceedings* of *the 2019 ASCE International Conference on Computing in Civil Engineering,* ASCE, Reston, VA.

Wu, J., and Zhang, J. (2019b). “New automated BIM object classification method to support BIM interoperability.” *Journal of Computing in Civil Engineering*, 33(5), 04019033.

Yalcinkaya, M., and Singh, V. (2014). Building information modeling (BIM) for facilities management–literature review and future needs. In *IFIP International Conference on Product Lifecycle Management* Springer, Berlin, Heidelberg.

Yang, Q. Z., and Zhang, Y. (2006). “Semantic interoperability in building design: Methods and tools.” *Computer-Aided Design*, 38(10), 1099-1112.

Zhang, J. (2018). “Towards systematic understanding of geometric representations in BIM standard: an empirical data-driven approach.” *Proc., ASCE Construction Research Congress*, ASCE, Reston, VA, 96-105.