

## Automated BIM Object Classification to Support BIM Interoperability

Jin WU<sup>1</sup> and Jiansong ZHANG<sup>2</sup>

<sup>1</sup> School of Construction Management Technology, Purdue University, West Lafayette, IN 47907; PH (301) 275-1272; email: wu1275@purdue.edu

<sup>2</sup> School of Construction Management Technology, Purdue University, West Lafayette, IN 47907; PH (765) 494-1574; FAX (765) 496-2246; email: jiansong-zhang@purdue.edu

### ABSTRACT

The ISO-registered industry foundation classes (IFC) data standard can facilitate building information modeling (BIM) interoperability by allowing a “one-to-many” software information flow pattern in the architecture, engineering, and construction (AEC) industry instead of the more complex “many-to-many” information flow pattern. However, even a full adoption of IFC cannot guarantee a seamless BIM interoperability, because of the flexibility allowed in adopting/using the IFC standard. Object classification in BIM is an important part of BIM interoperability. As part of an initial effort to pursue seamless BIM interoperability, the authors propose to use intrinsic geometric representations of IFC objects and geometric theorems to support the automated BIM object classification. A six-step method was proposed to develop automated IFC object classification algorithms using a data-driven approach. The method was preliminarily tested on classifying IFC objects into a cone frustum shape. The developed algorithm was successfully tested on three unseen bridge data. This shows an early promise of the proposed method to support error-free BIM object classification.

### INTRODUCTION

Building information modeling (BIM) is intended to support all phases and all disciplines of a building and construction project. Therefore, interoperability between different BIM software used in different phases and by different disciplines in a construction project is critical for the success of BIM. Industry foundation classes (IFC) is an ISO-registered data standard for building and construction industry data – ISO 16739. It is neutral and open. A wide adoption of the IFC standard can simplify the current “many-to-many” complex information flow pattern between BIM software into a “one-to-many” information flow pattern (Muller et al. 2017) in the architecture, engineering, and construction (AEC) industry. A broad use of the IFC standard can also reduce the current barriers to the transfer of information between different BIM applications, and alleviate/eliminate the information missing and inconsistency problems. However, even if the IFC standard is 100% adopted by all existing BIM software platforms, BIM interoperability still cannot be guaranteed to be seamless. This is largely because the IFC standard allows flexibility and can be used in a variety of ways. E.g., a wall might be modeled using `IfcWallStandardCase`, `IfcSlab`, or `IfcBeam` (Steel et al. 2012). Therefore, the object information directly read off from IFC entities may not be reliable. Nevertheless, such object information is required in a lot of BIM use scenarios such as automated cost estimation, where any object needs to be assigned

to a division (i.e., category, such as openings and furnishings) defined in the Construction Specifications Institute (CSI) classification systems such as UniFormat, MasterFormat, and OmniClass (Afsari and Eastman 2016). While object classification has been extensively researched on in natural language processing (e.g., named entity recognition) and computer vision domains (e.g., object recognition), similar efforts made in the civil engineering domain is relatively limited. Between the as-built model and as-design model, object classification studies that focused on as-design model are even fewer. It can be because this research gap is not quite obvious, as people tend to consider as-design models to be clear and well defined. Some automated BIM object classification efforts were reported by Belsky et al. (2015) and Ma et al. (2017), where expert knowledge were encoded into computable rules and heavily relied on for the classification decisions. Inconsistent results were observed using these methods, which may be partially attributed to the potential subjectivity of expert knowledge. In this paper, the authors propose a science-based and empirical data-driven method to develop automated object classification algorithms for IFC-based BIM models. The algorithm will rely on intrinsic properties of building elements and scientific theories (e.g., geometric theorems) rather than expert knowledge. The ultimate goal is to achieve an error-free object classification of IFC object in BIM models, to support seamless BIM interoperability in various AEC applications.

## **BACKGROUND**

### **Object Classification**

Object classification is detecting and recognizing an object automatically using a computer, based on the features and properties of the object. There are two main types of methods for object classification – machine learning (ML)-based methods and rule-based methods. A ML-based method utilizes ML algorithms for object classification (e.g., Weiss et al. 2010), whereas a rule-based method utilizes manually-coded rules (e.g., Li et al. 2001). Rule-based methods require more human effort for rule development, but tend to show high processing performance (Dragut and Blaschke 2006). Object classification is an important topic in many research areas such as computer vision (Hu et al. 2004), where the methods/algorithms for classifying static and moving objects into different categories (e.g., buildings, road signs, pedestrian) have been in rapid development.

In the AEC domain, models can be classified into two categories according to the way data were generated: as-designed model where people created the data, and as-built model where people collected the data. As-designed models are often created using a BIM authoring tool such as Autodesk Revit, Bentley Microstation, and Graphisoft ArchiCAD. As-built models are often created using sensors to detect and record physical objects. One of the most widely used sensing technique in the AEC domain is the Light Detection and Ranging (LIDAR) technique. It uses pulsed laser light and the reflection pulses to measure distances from any point in the space and derive the coordinates of all such points (NOS 2017). It was first used by the National Center for Atmospheric Research to measure clouds, and was widely used now in the AEC domain for capturing an as-built model (Rabatel et al. 2008). As-built models and As-designed models have different representation properties. For example, as-designed

models often use structured data formats such as IFC whereas as-built models often use unstructured data formats such as PCD (i.e., a format for point cloud data from LiDAR).

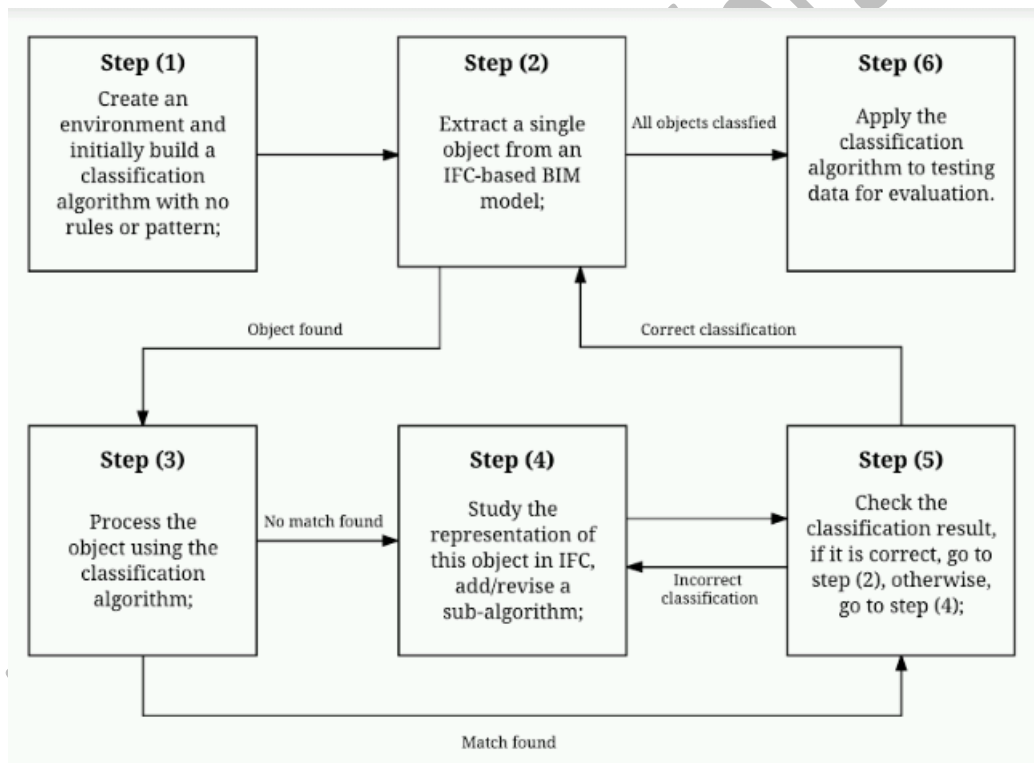
### **BIM Object Classification**

BIM is expected to facilitate information exchange between different stakeholders in the same AEC project, by representing both physical and functional characteristics of a project throughout all the phases of its lifecycle (GSA 2007). Industry foundation classes (IFC), as the ISO registered data standard for building and construction industry data, is designed to support information modeling and exchange in all phases of a building and construction project throughout its lifecycle: planning, design, construction, operation & maintenance, demolition or remodeling, to support the idea of BIM. Almost all major BIM software vendors claimed compatibility of their products with the IFC standard (Steel et al. 2012). IFC has been heavily studied in research and is representing the main focus area for solving BIM interoperability problems (Poirier et al. 2014). However, even with the clearly defined entities and attributes in an IFC schema, data can still be represented in different ways because of the following three main reasons: (1) flexibility is enabled by allowing more than one way of representing an object, for example, the geometry of a box-shaped column can be represented using either a faceted boundary representation (Brep) or a swept solid representation; (2) more flexibility is allowed by customized representation of any information using IFC property set; and (3) modeling choices can affect information representation using IFC, which can originate from either the inherent choices prescribed in a software program (e.g., using Brep to represent a beam) or the arbitrary decisions of a modeler who is using a BIM authoring software to create models (e.g., use ifcSlab to represent a wall). This variation in data representation using IFC goes against the interoperability goal of BIM. It clearly affects BIM applications such as a cost estimation tool. In cost estimation, there are two main steps: quantity take-off of building objects according to cost items, and pricing of the cost items based on unit price data. An important premise of these two steps is a correct identification of building objects, such as walls, doors, and slabs, and assigning them to the correct category in the corresponding classification system such as UniFormat, MasterFormat, and OmniClass. To automate such identification and assignment, automated object classification is needed. While there were few research works addressing such automated BIM object classification (Belsky et al. 2015; Ma et al. 2017), error-free classification to support seamless BIM interoperability is beyond the state of the art. The names of the objects defined in the IFC data may include errors caused by misuse. For example, an IfcSlab may be misused to represent a wall because their shapes could be similar. While the names of IFC entities can be misleading in this case of entity misuse, the geometric shapes information are reliable and it is almost always accurate for a successful building representation. To address the research gap of BIM object classification, a drastically different approach was initiated (Mandava and Zhang 2016; Akanbi and Zhang 2017) to investigate the potential of using solid scientific theories and intrinsic properties of BIM objects to support a seamless BIM interoperability. As part of this initial effort, the authors propose to use intrinsic geometric representations

of IFC objects and geometric theorems to support the automated BIM object classification.

## PROPOSED METHOD

The authors propose the following 6-step method to automatically classify BIM objects in IFC (Figure 1): (1) create an environment (e.g., database) for extracting objects from IFC-based BIM models and initially build a classification algorithm with no rules or patterns; (2) extract a single object from an IFC-based BIM model (i.e., training model); (3) process the object using the classification algorithm, if there is no match with any pattern, go to step (4), otherwise, go to step (5); (4) study the representation of this object in IFC, add/revise a sub-algorithm using pattern-based rules for identifying this type of object into the classification algorithm, and go to step (5); (5) check the classification result, if it is correct, go to step (2), otherwise, go to step (4); (6) when all objects in the training model is processed, apply the classification algorithm to testing data for evaluation.



**Figure 1. The proposed 6-step Method**

This method is designed for classifying any IFC object automatically into predefined categories. A rule-based sub-algorithm will be developed for each predefined category, to identify an object in that category, such as windows, walls, and slabs in the case of building object categories, and cubes, cylinders, and cones in the case of geometric shape categories. As a premise of the sub-algorithm development, properties of these IFC categories are studied and formalized into features to help construct the rule pattern for use in the sub-algorithm. Out of the rich information presented in an IFC model, geometric representations usually take a significant portion.

Therefore, geometric properties will be used primarily in the rules of sub-algorithms. Other information can be leveraged in addition as needed. In an IFC model, geometric properties are represented in two ways: (1) through the inherent geometric relationships between the Cartesian points used to represent a shape; and (2) through the IFC data structure in representing the geometric information.

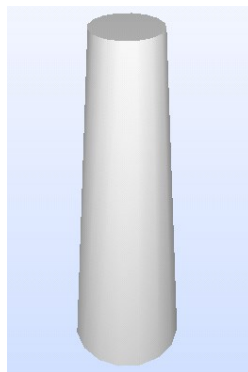
The ultimate algorithm developed using the proposed method will contain many sub-algorithms with each sub-algorithm designated for one known category. This algorithm will process an input object data by applying existing pattern-based rules of different categories to find a category match. Therefore, the success of the algorithm relies on the success in finding a category match by its sub-algorithms. To achieve an accurate classification, the sub-algorithm will check a combination of properties of the object based on geometric theorems. If the object matches all the required properties of a certain category, then it is classified into that category. Each sub-algorithm is designed to accept the data that belongs to the corresponding category, and reject the data that does not belong to that category.

## EXPERIMENTAL RESULTS

To test the proposed method, a small experiment was conducted on cone frustum-shaped piers of simple bridge models in IFC, for classifying objects into this shape category. The detailed steps are described as follows:

(1) *Create an environment (e.g., database) for extracting objects from IFC-based BIM models and initially build a classification algorithm with no rules or patterns.* In this step, an empty database was created to accommodate future IFC objects; and a classification algorithm was initialized in Java programming language with no known type of shapes.

(2) *Extract a single object from an IFC-based BIM model (i.e., training model).* In this step, the simple bridge model created by Mandava and Zhang (2016) was used as the training model. A cone frustum-shaped pier in the simple bridge model was extracted from the training model (Figure 2). A cone frustum shape is a cone shape with its top sliced off (WMW 1999).



**Figure 2. A cone frustum-shaped pier**

(3) *Process the object using the classification algorithm, if there is no match with any pattern, go to step (4), otherwise, go to step (5).* In this step, because the



classification algorithm was just initialized, it did not have any rules yet, so there was no match found with any pattern, we moved to step (4).

(4) *Study the representation of this object in IFC, add/revise a sub-algorithm using pattern-based rules for identifying this type of object into the classification algorithm.* In this step, through studying the shape representation of the cone frustum-shaped bridge pier, the following geometric representation details were found: to represent the shape of the pier, a faceted boundary representation (Brep) was used, which was composed of a closed shell. The closed shell was further composed of 18 faces. Sixteen of the faces were rectangular, and two of the faces were hexadecagons. The hexadecagons were used to approximate circular shapes. The 16 rectangular faces were on the outer boundary of the object, to represent the side faces. These side faces together are approximating a smooth side face in one piece, in a similar way as the hexadecagons were used to approximate circles. Through analysis, the authors identified three intrinsic geometric properties of a cone frustum shape:

- a. Top and bottom faces are circles.
- b. Top and bottom faces are in parallel.
- c. The axis of the shape is perpendicular to the top and bottom faces.

Correspondingly, the following pattern-based rules were developed:

**Rule #1:** There are two and only two faces that are circles. From the mathematical definition, all points on the boundary of a circle have the same distance from the center of the circle. This definition was used to identify a circle. To find the center of the circle, Equation (1) was used:

$$(x, y, z) = \frac{1}{n} \sum_{i=1}^n (xi, yi, zi) \quad (1)$$

In Equation (1),  $x, y, z$  are the coordinates of the center, and  $xi, yi, zi$  are the coordinates of vertex # $i$  of the hexadecagon. The average of coordinates of all 16 vertices were calculated and used as the coordinates of the center. A distance check was conducted by comparing the distances from all 16 vertices to the center. Theoretically this distance should be exactly the same. But due to rounding errors in practical data, a strict equation criterion would have failed. Therefore, a tolerance of error was needed. To analyze the needed tolerance, the distances from each of the 16 vertices to the center for both the top and bottom surfaces were computed, using Equation (2). Figure 3 shows the results of the distances.

$$distance = \sqrt{(xi - x)^2 + (yi - y)^2 + (zi - z)^2} \quad (2)$$

Units of measure were ignored because they would not affect the results of the rules. Based on observing the computed distances and the maximum rounding error, the tolerance of error was set to  $\pm 0.000001$ .

===Function checkCircle Begin===	===Function checkCircle Begin===
Average of x is: 6.4731809999999985	Average of x is: 6.4731809999999985
Average of y is: -0.47683500000000006	Average of y is: -0.47683500000000006
Distances are:	Distances are:
0.16000002403300104	0.16000002403300104
0.1600003252980008	0.1600003252980008
0.16000002403300048	0.16000002403300048
0.15999999999999998	0.15999999999999998
0.1600000240329994	0.1600000240329994
0.1600003252979993	0.1600003252979993
0.16000002403299907	0.16000002403299907
0.15999999999999887	0.15999999999999887
0.1600000240329991	0.1600000240329991
0.16000032529799937	0.16000032529799937
0.16000002403299948	0.16000002403299948
0.16000000000000006	0.16000000000000006
0.16000002403300057	0.16000002403300057
0.1600003252980009	0.1600003252980009
0.16000002403300106	0.16000002403300106
0.16000000000000001	0.16000000000000001
===Function checkCircle End===	===Function checkCircle End===

**Figure 3. The distances from all vertices to the center of the circle, for top and bottom surfaces**

**Rule #2:** The top face and the bottom face must be in parallel. This was checked through comparing the mathematical representation of the two faces, using Equation (3) and criterion (1) below.

$$ax + by + cz + d = 0 \quad (3)$$

In Equation (3), x, y, and z are the coordinates of any point in the shape. A, b, c, d are four parameters used to define a plane. To decide these parameters for a hexadecagon, three points were randomly chosen from its 16 points. The parameters a, b, and c were solved by plugging in the coordinates of the three selected points. And plugging in the solved a, b, c, and coordinates of any one point can give the value of d.

**Criterion (1):** Given two planes  $a_1x + b_1y + c_1z + d_1 = 0$ , and  $a_2x + b_2y + c_2z + d_2 = 0$ , they are parallel if and only if  $a_1 = a_2$ ,  $b_1 = b_2$ ,  $c_1 = c_2$ , and  $d_1 \neq d_2$ .

**Rule #3:** A line that connects the centers of the top and bottom faces must be perpendicular to both faces. This rule guarantees the cone frustum shape not to be a skewed cone frustum. Cylinder is a special case of a cone frustum. This case can be distinguished by the following Criterion (2);

**Criterion (2):** If the top and bottom faces of the cone frustum have the same size (i.e., radius), then the cone frustum is a cylinder.

After developing these rules and criteria, we moved to step (5).

(5) Check the classification result, if it is correct, go to Step (2), otherwise, go to step (4). In this step, a quick check on the classification result of the cone frustum

shape using the above sub-algorithm found it correct. We should have moved to Step (2), but as a preliminary and exemplary small experiment, we only considered one shape category (i.e., cone frustum shape). So we moved to step (6).

(6) *When all objects in the training model is processed, apply the classification algorithm to testing data for evaluation.* In this step, piers from three other bridges retrieved from online sources were used as the testing data (Mandava and Zhang 2016). Table 1 shows some parameters of these piers. Applying the developed classification algorithm on these three piers successfully classified them into the cone frustum shape category.

**Table 1. Parameters of the three piers in the testing data.**

<b>Margins</b>	<b>Axis Length</b>	<b>Top Face Radius</b>	<b>Bottom Face Radius</b>
Bridge Pier	2.36	0.04	0.16
Test Pier 1	10.00	2.25	9.00
Test Pier 2	5.00	0.40	4.00
Test Pier 3	1.16	0.03	0.14

## ANALYSIS

In this experiment, an algorithm was developed based on the IFC model of a simple bridge pier, to identify the cone frustum shape. The developed algorithm was successfully tested on three more bridge piers retrieved from online sources. An error tolerance in equal distance check was decided by taking the maximum rounding error observed. Although it was tested to be successful in the testing data, it may not be the case in all possible testing data. Because such rounding errors depend on the software used to create the model as well as the parameters/configurations chosen by the modeler. In the ultimate classification algorithm that includes sub-algorithms for all known categories, a good estimation of this error tolerance may be obtained by analyzing a large amount of training data. Also, the use of a relative value to represent error tolerance (i.e., a percentage of a dimension of the shape) may be more robust than the use of an absolute value to represent error tolerance, such as the value  $\pm 0.000001$  used in this experiment. But this may also be dependent on the range of the absolute sizes of the dimensions. E.g., when comparing two data values close to zero, their relative difference can be large where the absolute error stays small (i.e., bounded by the two data values). The same rationale used in comparing absolute value with relative value for error tolerance applies to relations used in the patterns of the sub-algorithms as well. Specifically, relative relations are expected to be more robust than absolute relations. For example, the relative relation (e.g., parallel relation) between the top and bottom surfaces of a cone frustum shape is more robust than the absolute relation (e.g., horizontal) between each surface and the world coordinate. But if certain assumptions are known to be valid, then absolute relations may be preferred because they are simpler to use. For example, if the assumption that all piers stand vertically is true, then it is easier to use the criterion that the top and bottom surfaces are horizontal. In terms of computing efficiency, the ultimate classification algorithm developed based on the



proposed method will have a time efficiency of  $O(n)$  with regard to the number of predefined categories for the classification.

## CONCLUSIONS

Automated object classification is needed in many BIM applications to support BIM interoperability. However, few researches have focused on automated object classification of IFC models, and error-free classification is beyond the state of the art. As part of an initial effort to investigate the use of scientific theories and intrinsic properties of BIM objects to support seamless BIM interoperability, the authors propose a rule-based algorithm development method for developing IFC object classification algorithms. To test the proposed method, a preliminary experiment was conducted where an algorithm for classifying IFC objects into the cone frustum shape was developed based on one bridge pier of such shape. The algorithm was successfully tested on three other unseen bridge piers. The use of scientific theories and intrinsic properties of objects in such classification algorithms is expected to avoid errors from subjectivity embedded in expert knowledge. The use of a rule-based method makes it one step closer to error-free classification, which is hard (if not impossible) to achieve using machine learning-based methods.

## LIMITATIONS AND FUTURE WORK

A main limitation of the reported research is the small scope of the preliminary experimental testing. The authors only tested the proposed method on one shape and a small amount of data. While this may well serve the purpose of explaining the proposed method and showing its potential impact, the testing on more categories is needed to reveal practical issues such as error tolerance variation. In addition, as the number of categories keep growing, an object may be found to match more than one categories. While this may be prevented theoretically by a well-defined taxonomy of categories, how this problem will reveal itself in practice and how it should be addressed need further investigation. In future work, the authors plan to extend the experiment by incorporating more categories and more data.

## ACKNOWLEDGMENTS

The author would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1745374. 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

- Afsari, K., and Eastman, C.M. (2016). "A comparison of construction classification systems used for classifying building product models." Proc., 52nd ASC Annual International Conference, Associated Schools of Construction, Fort Collins, CO.
- Akanbi, T., and Zhang, J. (2017). "Automated wood construction cost estimation." Proc., 2017 ASCE Intl. Workshop on Comput. in Civ. Eng., ASCE, Reston, VA, 141-148.

- Belsky, M., Sacks, R. and Brilakis, I. (2015). "Semantic enrichment for building information modeling." *Computer-Aided Civil and Infrastructure Engineering*, 31, 261-271.
- Dragut L., and Blaschke, T. "Automated classification of landform elements using object-based image analysis." *Geomorphology*, 81(3-4), 330–344.
- General Service Administration (GSA). (2007). "BIM guide overview." GSA BIM Guide Series 01., U.S. General Services Administration, Washington, DC. <<http://www.gsa.gov/portal/getMediaData?mediaId=226771>> (Aug. 31, 2017)
- Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). "A survey on visual surveillance of object motion and behaviors." *IEEE Transactions on Systems, Man, and Cybernetics, Part C*.
- Li, W., Han, J., and Pei, J. (2001), "CMAR: accurate and efficient classification based on multiple class-association rules" Proc., Data Mining. ICDM, Doubletree Hotel, San Jose, California, 7169318.
- Ma, L., Sacks, R., and Kattel, U. (2017). "Building model object classification for semantic enrichment using geometric features and pairwise spatial relationships." Proc., The Joint Conference on Computing in Construction (JC3), Heraklion, Crete - Greece, 373-380.
- Mandava, B., and Zhang, J. (2016). "A new automated quantity takeoff method for BIM-based bridge designs." Proc., CIB W78, Conseil International du Bâtiment (CIB), Rotterdam, The Netherlands.
- Muller, M.F., Garbers, A., Esmanioto, F., Huber, N., Loures, E.R., and Canciglieri, O.J. (2017). "Data interoperability assessment though IFC for BIM in structural design- a five-year gap analysis." *Journal of Civil engineering and management*, 23(7), 943-954.
- National Ocean Service (NOS). (2017). "What is LIDAR?" OCEAN FACTS, <<https://oceanservice.noaa.gov/facts/lidar.html>> (Jul. 24, 2017)
- Poirier, E.A., Forgues, D., and Staub-French, S. (2014). "Dimensions of interoperability in the AEC industry." Proc., 2014 ASCE Construction Research Congress (CRC), ASCE, Reston, VA, 1987-1994.
- Rabatel, A., Deline, P., Jailliet, S., and Ravel, L., (2008). "Rock falls in high-alpine rock walls quantified by terrestrial lidar measurements: A case study in the Mont Blanc area." *Geographical Research Letter*, AGU PUBLICATIONS. 35(10), L10502.
- Steel, J., Drogemuller, R. and Toth, B. (2012). "Model interoperability in Building Information Modelling." *Softw. Syst. Model*, 11(1), 99-109.
- Weiss, U., Biber, P., Laible, S., Bohlmann, K., and Zell, A. (2010). "Plant Species Classification using a 3D LIDAR Sensor and Machine Learning." Proc., Ninth International Conference on Machine Learning and Applications (ICMLA). AMLA, WSU, CSUB and IEEE SMCS, Bethesda, Washington, D.C.
- Wolfram Math World (WMW). (1999). "Conical Frustum." <<http://mathworld.wolfram.com/ConicalFrustum.html>> (Aug. 3, 2017)