Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2018 Volume 24 Issue 6: 481–498
https://doi.org/10.3846/jcem.2018.5689

CHARACTERISTICS OF MODELS THAT IMPACT TRANSFORMATION OF BIMS TO VIRTUAL ENVIRONMENTS TO SUPPORT FACILITY MANAGEMENT OPERATIONS

Zhengbo ZOU[1*], Luiz ARRUDA[2], Semiha ERGAN[3]

Department of Civil and Urban Engineering, New York University, New York City, U.S.A.

Received 17 May 2018; accepted 08 August 2018

Abstract. Building information models (BIMs) have been used by the Architectural/Engineering/Construction (AEC) industry with a focus on storing and exchanging digital information about building components. However, the untapped potential of BIMs in facility operations and the experience of facility operators while they interact with digital building information have not been understood widely. One of the underlying bottlenecks in the use of BIMs in the FM phase is the lack of interactions with components to easily access information of interest, and the lack of ways to navigate in models with full spatial understanding. Virtual environments (VEs), which represent physical spaces digitally in virtual worlds, enable interactions with virtual components to access information with spatial understanding. The underlying challenges in the conversion of BIMs to VE hinder a streamlined process. This paper provides a detailed analysis of building size, geometric complexities of discipline models and level of geometric granularity as factors contributing to inefficient transformation of BIMs to VE. The paper also provides research findings on a set of computational approaches such as polygon reduction and occlusion culling to overcome challenges and improve the data transfer faced in converting BIMs into VEs over a range and size of facility models.

Keywords: BIM, virtual reality, facility management.

Introduction

Within the capital projects industry, approximate cost of inadequate interoperability to owners and operators is 1.6 billion dollars every year (Gallaher *et al.* 2004). Such costs are incurred by facility owners and operators due to inadequate information sharing, need for data translations between applications, and idle times for information access and verification in work processes. Building information models (BIMs) have been adopted by the Architectural/Engineering/Construction (AEC) industry to reduce such costs. Wider adoption of BIMs in the industry has led owners and operators ask for BIMs as project deliverables for facilities management (FM), however, with limited guidance on how to integrate models to daily work processes. A BIM provides information about a specific component or a system (e.g. family type, geometry, location, zones, and manufacturer) – information that is crucial from a FM perspective. Although models provide rich facility and component information, they fail to provide full spatial understanding of spaces, walkthroughs and interactions with virtual components simultaneously without interruptions (Bowman *et al.* 2006).

Virtual environments (VEs) are digital representations of physical spaces in virtual settings (Billger *et al.* 2004). They enable accessing facility information in an integrated and navigable environment with spatial understanding for FM operations. Such virtual worlds are possible by converting existing BIMs into gaming engines or using them in virtual reality (VR). However, within the current practice, there is not a single workflow of converting BIMs to VEs and there is a lack of knowledge on the impact of model characteristics on performance and efficiency of these conversion processes.

In any workflow of converting models to VE, there are various applications that can take part in the model conversion process, such as BIM authoring tools, rendering tools, game engines, and virtual reality tools. A key requirement in the conversion of BIMs to VEs is the use of a data exchange standard for interoperability between applications. Industry foundation classes (IFC) are widely used in the industry to enable seamless data exchange in the architectural, engineering, construction and facilities management (AEC/FM) industry, however, is not quite used in

*Corresponding author. E-mail: *zz1658@nyu.edu* the model to VE conversion. Also, to simulate first-person navigation and interactions within the VE, game engines provide tool sets for first person player and script development that enable the creation of customized interactions. The VE can then be exported to a virtual reality application for navigating the building with motion-capture devices. While these tools would greatly improve the workflow for facility managers, there are several characteristics of building information models that impact the performance (e.g. semantic information retained, component identity retained) and efficiency (e.g. time) of the conversion of BIMs to VEs. Within the current practice, VEs are mainly studied in software and game development fields. Since these fields rely on models developed directly within utilized gaming environments, the implications of models generated from typical BIM authoring tools on the conversion process require further analysis. Although the key players of the conversion of BIMs to VE are known, the efficiency of the conversion process and the performance of the resulting models in VE vary significantly depending on what the models contain and tools that are used in each phase.

The objective of the study is to identify the impact of BIM in the generation VE for FM use, and analyze the efficiency of the conversion process, and the performance of the resulting models in VE. A set of metrics was defined by the authors to analyze the impact of models on the VE conversion process. Model complexity is measured with respect to the number of objects represented in each discipline (e.g. architectural, structural, mechanical), the building size (i.e. total gross square feet or square meters floor size), and the level of geometric granularity (i.e. coarse, medium, fine) of components in a model. These metrics were varied to generate testbed models and used to analyze the efficiency and performance of the BIM to VE conversion process. The study compares how different complexities of BIMs affect the performance (i.e. model/component sizes in each step of the conversion process, semantic information and component identity retained in resulting VEs) and efficiency of the conversion process. A total of thirty-six different building information models was generated with variations in model complexity and used in structured experiments in a visualization laboratory. Furthermore, this study provides a set of approaches to overcome challenges (such as semantic information loss, increase in rendering time in heavy models, loss of realistic renderings, lack of interactions) faced in converting BIMs into VEs over a range of facility models and scenarios. Solutions to overcome the issues are discussed to improve the interaction of facility operators with BIM in virtual settings and improve the data transfer in the whole process chain. The next section discusses related background research on creating VEs for AEC/FM applications.

1. Background research

The use of VEs for facility operations has been discussed in various research studies in literature. The related research is discussed in three major subsections as (a) VEs and their use in facilities management, (b) approaches for converting BIM to VEs for AEC/FM, and (c) challenges in converting BIMs to VEs for AEC/FM use.

1.1. VEs and their use in facility management

Various bottlenecks exist in the facilities management industry in accessing facility information in a timely manner. Although the difficulties in accessing facility information in document-based archives are well known, the challenges in using building information models in facilities operations are not widely acknowledged. Two of the significant barriers in using building information models for accessing facility information are (a) the lack of customized interactions with components in models to access and query data, (b) the lack of free navigation in virtual spaces with immediate and first-person experience in a scaled setting with or without immersiveness (Nopachinda, Ergan 2016; Ergan, Yang 2017; Du *et al.* 2018). These challenges exist mainly due to perceptual reasons, where BIM based desktop visualizations lack the depth cues that human eye uses in the real world for visual processing of data sets in spatial layouts (Bowman *et al.* 2006). Depth cues generated in desktop 3D visualizations are static monocular (i.e. received by only one eye for a still image), and fail at binocular or oculomotor depth cues, which are essential to get the gap between two eyes and the sense of convergence, respectively. In addition, BIM based desktop visualizations cannot provide navigational cues with respect to the viewer's orientation and location (feeding to viewers' kinaesthetic senses to sense up/down, view change with head/body movements), which are essential to fully understand and experience the spatial environment. VEs overcome these barriers when BIMs are converted to VEs by allowing viewers to access data with a spatial understanding.

Virtual reality applications in the AEC industry are abundant to support processes such as worker training (e.g. Wang *et al.* 2004; Dünser *et al.* 2006), design review (e.g. Gopinath, Messner 2004; Yang *et al.* 2015), construction planning (e.g. Messner 2006; Savioja *et al.* 2003; Yerrapathruni 2003), architectural visualization and collaboration (e.g. Hammad *et al.* 2009; Goulding *et al.* 2014; Kasireddy *et al.* 2016; Du *et al.* 2017a, 2017b), constructability review (e.g. O'Connor, Davis 1988; Pulaski *et al.* 2006), and progress monitoring (e.g. Behzadan *et al.* 2008). Within the facilities management (FM), although the pace of adoption of VEs is still slow, it is gaining momentum with building information models being part of contract deliverables. So far, building information models are converted into VEs mainly for three major aims. The first group of studies that utilized VE for facilities management focuses on developing design with maintainability in mind. Several studies evaluated the efficiency of using VE in capturing FM operators' perspectives in design (e.g. Pitt *et al.* 2005), especially in the design of healthcare facilities (e.g. Wang *et al.* 2013). The second group of studies evaluated VE in

FM to access facility information (e.g. Nopachinda, Ergan 2016; Mozaffari *et al.* 2005; Lin, Su 2013; Chen *et al.* 2013; Shen *et al.* 2012; Lee, Akin 2011; Ergan, Yang 2015), sensor monitoring in facilities (e.g. Yang, Ergan 2014), and problem solving in FM domain (e.g. Shi *et al.* 2016). The third group of studies aims to understand the energy behavior of buildings with the use of VE, such as modeling and prediction of occupant behaviors (e.g. Saeidi *et al.* 2015), understanding occupant lighting preferences in commercial spaces (e.g. Heydarian *et al.* 2015), and energy benchmarking (e.g. Woo *et al.* 2016).

These research studies discussed in this section focused mainly on the value of using VE in supporting AEC/FM tasks. However, it is essential to understand to implications of model complexities on the performance and efficiency of the resulting VEs. Without identifying the boundaries of variations in the model complexities, resulting VEs will be bound to have issues in serving the information needed for FM tasks and will have inefficiencies in the conversion process. Such problems obviously create problems in handling the FM tasks in virtual settings.

1.2. Approaches for converting BIMs to VEs for AEC/FM

Workflows in converting BIMs to VEs in the AEC industry are limited due to the unique geometric, texture and semantic representation requirements that are associated with building components and hard to see in gaming engine libraries. Previous research efforts highlight workflows that utilize various application mediums that work around issues to address these unique requirements.

The workflows that utilize application mediums rely on a configuration of BIM authoring tools, 3D architectural rendering tools, gaming engines, virtual equipment and middleware in between these (Nopachinda, Ergan 2016). BIM authoring tools include applications (e.g. Autodesk Revit, Bentley MicroStation, ArchiCAD, Vectorworks) that are developed specific for the AEC/FM industry to generate the semantically rich BIMs; rendering tools include applications (e.g. 3DsMax, LightWave, Swift 3D Max, Cinema4D) that enable production of realistic visualizations with shading, lighting, shadowing, and texture effects; gaming engines are platforms used to create video games (e.g. CryEngine, Unity, Unreal Engine); virtual equipment enables tracking, immersiveness and interactivity of users with the VE generated (e.g. object tracking solutions, reverse projections, head mounted displays, immersive CAVE); and the middleware contains a set of software to enable communication of the virtual equipment and the gaming engines (e.g. middleVR, Vicon Pegasus). Three main workflows are currently known in the AEC/FM industry that rely on a combination of these application mediums.

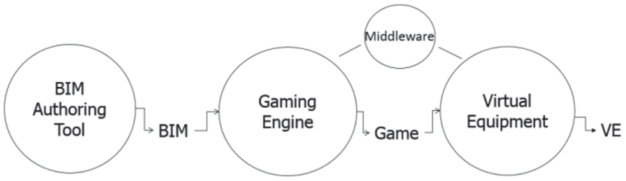


Figure 1. Workflow 1 for generating VEs from a model

The first workflow includes exporting models from BIM authoring tools in a format that is recognizable in the gaming engine platform as shown in Figure 1. This workflow has been mentioned in various research settings (e.g. CIC 2014; Dalton, Parfitt 2013; Kumar *et al.* 2011). To give an example for this workflow, models exported in BIM authoring tools (e.g. Revit, MicroStation, Navisworks, Sketchup Pro) in FBX format can be imported into gaming engines (e.g. Unity 3D and Unreal Engine).

Due to the fact that the texture of materials represented in BIM authoring tools does not always map to the standard texture types, the models loaded in the game engines will appear as white meshes without textures, as shown in Figure 2. Thus, although this workflow looks shorter, the resulting VE is far from realistic representations. Another major challenge in this workflow is that the semantic information on components represented in a BIM authoring tool is lost in the translation, including the texture and material information, which corresponds to the visible part of the lost information.

The second workflow includes utilization of a rendering tool between the BIM authoring tool and the gaming engine to bring back the lost textures to the gaming environment (Figure 3). This workflow requires manually reapplying the material and texture data in the rendering tool,

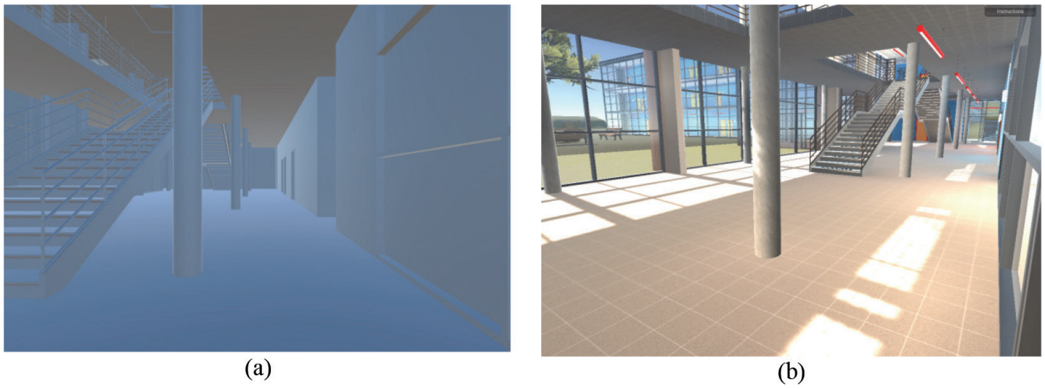


Figure 2. A model view: (a) without textures and (b) with textures

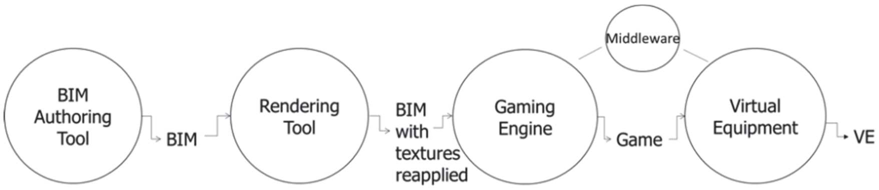


Figure 3. Workflow 2 for generating VEs from a model

which is tedious and labor intensive. Some components have more than one texture to reapply. For example, a window could have two textures, one for glasses and the other for the window frames. A work around for this workflow is to use material rendering converters to automatically convert material and texture information to the gaming engine libraries. This is an effective way of reapplying the materials automatically without the manual process. Still, in this workflow the challenge is the lost invisible semantic information about components (e.g. the notion of being a component, connectivity, component properties).

The last workflow includes direct importing of native BIM files into the VE (Kang *et al.* 2012; Johansson *et al.* 2014). This workflow bounds the users to the data exchange format used in the virtual equipment platform, limiting the interoperability of BIM to VE conversion (Figure 4). This workflow is the fastest workflow among the three workflows available. Although they are effective to bring in BIMs directly to VE without loss of texture, user interactions with components in VE are restricted to the plug-in used during conversion (e.g. TechViz) or to the software platform running directly on the virtual equipment (e.g. EonReality ICube). In this workflow, it is not possible to create customized interactions if only a plug-in is used to visualize a BIM in virtual settings. In the latter case, extensive simulations should be scripted in the software platform running on the virtual equipment to enable interactions with objects. Research based efforts to enable VE from BIM also fall under this category, where BIM authoring tool APIs are used to develop the VE (e.g. Yan *et al.* 2011).

In order to better understand the implications of BIM complexity on the conversion process, the authors used the longest path (with the highest number of application mediums used), which is Workflow 2 to conduct the experiments.

1.3. Challenges in converting BIMs to VEs for AEC/FM use

Challenges listed in the literature regarding BIM conversion to VE are mainly from the perspective of technological obstacles in the conversion, rather than the value of information and information loss when models are converted to VE for specific applications. Reapplication of textures, heaviness of the models generated in BIM authoring tools in the rendering and VE navigation, manually re-entering information to gaming engines, and interoperability between the BIM authoring tools and the gaming engines are highly referred challenges in the literature (e.g. Nopachinda, Ergan 2016; Shen *et al.* 2007, 2012; Dalton, Parfitt 2013; Figueres-Munoz, Merschbrock 2015; Bille *et al.* 2014).

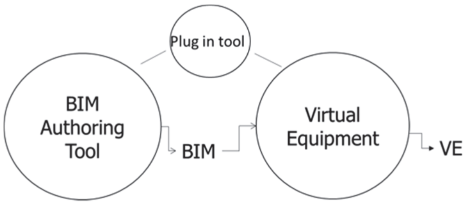


Figure 4. Workflow 3 for converting BIMs to VE

However, it is also known that the VE generated from these workflows has a high demand on computer graphics hardware (Regan, Ronald 1994). A building information model generally creates complex 3D meshes with a large number polygon count (CIC 2014). In order to create a good user experience, the model should run smoothly. Slow frame rate can create discomfort or user sickness (Chen, Thropp 2007). Several optimization techniques (Bern, Eppstein 1992; Evans *et al.* 1996; Zhang 1998) have been studied to enhance game performance with acceptable frame rates. Polygon reduction is an important part of these optimization models and can drastically decrease the number of objects being rendered during runtime. However, the reduction in polygons can result in fewer details, making the models less realistic. One way of overcoming this downsize is using textures for regaining some of the geometric details lost during the polygon reduction process.

The previous research studies and explorations of VEs list challenges from technical, interoperability and texturing information, however previous studies are limited in shedding light on understanding of key factors in models that impact the conversion process, the types of data lost in the conversion process, the quality of models after reducing model complexities and implications of these on performing FM operations. The experiment results reported in this paper aim to improve the state of the art on BIM to VE conversion with such an understanding.

2. Research method

The objective of the study is to identify the impact of BIM complexity in the generation of VEs for FM use, and analyze the efficiency of the conversion process, and the performance of the resulting models in VEs in serving FM tasks. The research method consisted of conducting structured experiments in each step of the workflow with different complexities of BIMs. During the experiments, several characteristics about BIMs were modified and used in order to understand the implications of these characteristics on the BIM to VE conversion. A summary of these characteristics that define model complexity are provided in Table 1. These characteristics included the *number of components* represented in models from different disciplines (e.g. architectural, structural, mechanical and electrical) (in object count), *building size* (in square foot or square meter), and *level of geometric granularity* (e.g. coarse, moderate, fine). Object count in each discipline model is different than polygon count in the sense that objects are physical representations of building components (e.g. wall, column) and each object has its own polygon count. Polygon count individually should not be used to define model sizes, as a small set of objects can cause a very large polygon count. Hence, we used the object count and also introduced the building size and level of geometric granularity metrics to analyze the results.

Table 1. Metrics used and varied in the experiments for defining model complexity

Metrics for defining model complexity	Description
Object count in each discipline	Refers to the number of physical objects per discipline (e.g. Architectural, Structural) in a given BIM.
Building size	Refers to the total gross square foot or square meter floor size to differentiate small, medium and large size buildings.
Level of geometric granularity	Refers to the coarse, medium and fine representation of geometry in models.

Building size is a key factor to investigate in the BIM to VE conversion as it dictates the model complexity and hence the conversion process. Commercial buildings are categorized by Energy Information Administration as small (with less than 5,000 sf of floor space), medium (with floor space between 5,000–50,000 sf) and large (with more than 50,000 sf floor space) size buildings depending on the total floor space they have (EIA 2012). The authors also note that depending on the type of building (e.g. medical office building vs. an office building), the complexity of the models can change irrespective of the size of the building. Hence, the authors only focus on commercial and administrative buildings in their evaluations when defining the building size categories.

Given a BIM, level of geometric granularity (LGG) defines the amount of discrete geometric components to be exported in a view from a BIM authoring tool and affects how the same geometry will be viewed in different scales. It is quantified by polygon counts in each granularity for a given model. The different LGGs had a considerable impact on the availability of the components in the models. For instance, for a sample model, the coarse level did not display ducts and only displayed the equipment in the mechanical trade model, as shown in Figure 5. Hence, LGGs was an important factor to analyze with respect to the data loss in the workflow of converting BIMs to VE. It should be noted that Level of Detail (LOD) and Level of Geometry Granularity (LGG) are different concepts. LOD is used to define the development level of a model for model content reliability, and contains both geometric and semantic information. On the other hand, in the computer science community LGG is only a representation that specifies the level of detail regarding the geometric information of a model (Stell, Worboys 1998). Since the objective of this paper is to pinpoint the bottlenecks in the process of bringing design models to virtual environments, LGG was used to represent the geometric complexity of the models. In the sense of model completeness, the testbeds used in this study are well developed to LOD 500. In practice, commercial software like Autodesk Revit, adapts LGG as one parameter of model rendering, and the term "Detail Level" is used to represent LGG (Autodesk 2017). LGG could be retrieved from the model through Autodesk Revit's Developer API. The dedicated parameter is called Detail Level and has both get and set method (Revit API 2017). However, if no programming environment is provided, the user would not be able to identify if a model is in coarse, moderate or fine LGG outside of Revit.

2.1. Preparation of testbeds for the experiments

In this study, BIMs from real buildings in categories small, medium and large buildings were used to create variations of models with components from different disciplines. The small building was a one-story office building with a gross area of 1,300 square feet (121 square meter) (Figure 6), the medium building was a two-story office building with a gross area of 40,000 square feet (3,716 square foot) (Figure 7), and the large building was a fourteen-story academic building with a gross area of 566,000 square feet (52,583 square meter) (Figure 8).

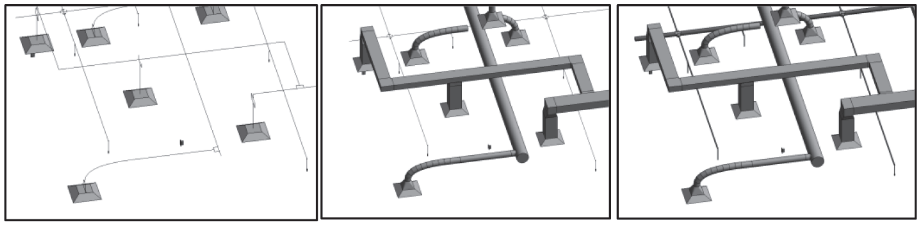
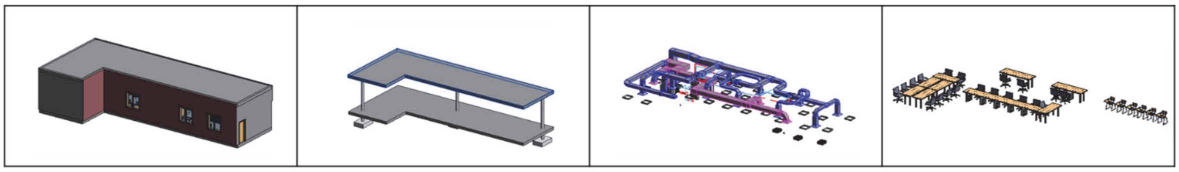
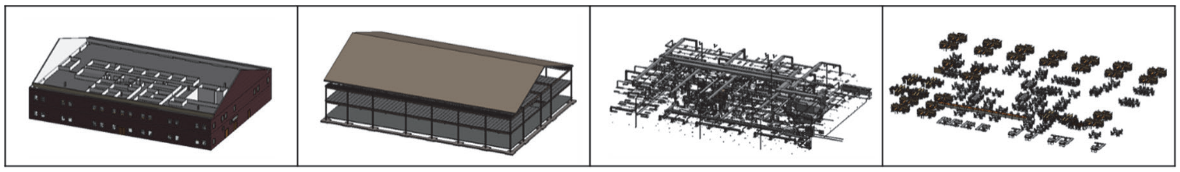


Figure 5. Mechanical components displayed in coarse, moderate and fine LGG (left to right)



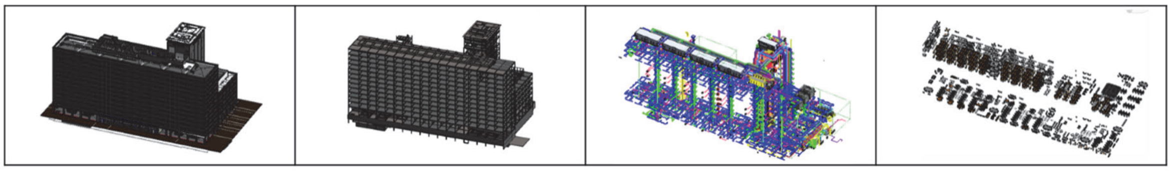
Architectural Structural Mechanical Furniture

Figure 6. Snapshots from the BIM of the small building



Architectural Structural Mechanical Furniture

Figure 7. Snapshots from the BIM of the medium building



Architectural Structural Mechanical Furniture

Figure 8. Snapshots from the BIM of the large building

The models were not selected arbitrarily as the polygon count for each component can significantly affect the model complexity, regardless of building size. Hence, the models were selected based on the square foot or square meter of the buildings for small, medium, and large buildings, and then existing BIMs of these selected buildings were modified to create a controlled set of models with reasonable polygon counts for each object. To be able to achieve this, each discipline model (of each building selected) was reviewed to compare the corresponding polygon counts of each component type in each model and find complex geometries in the models for all component types. These geometries were modified to represent the components in coplanar surfaces instead of with geometries having high areas of curvature. Geometries were simplified in such cases to have a controlled set of BIMs.

For all three modified building models, each discipline model was incrementally integrated to the architectural model to increase the complexity of each model to generate 12 different models. Each of these 12 models was then exported from the BIM authoring tool using three levels of geometric granularities (LGG) as: coarse, moderate, and fine- resulting in 36 models to test. Different types of components are required in each discipline and the model complexity has an influence on geometric complexities required to represent them in models. Hence, component types for each discipline were also considered as a model characteristic that should be investigated in the BIM to VE conversion. In the experiments, the architectural discipline models consisted of architecture walls, doors, windows, ceiling tiles, curtain systems, ramps and stairs. The structural discipline models consisted of structural walls, beams, columns, foundation components, ceilings, slabs, and reinforcement components. The mechanical discipline models consisted of ducts, duct placeholders, air terminals, mechanical equipment, pipes, and pipe placeholders. The furniture discipline models consisted mainly of desks, chairs, computers and electronic devices, landmarks, shelfs, closets and lockers. In each model, the number of physical objects represented (i.e. object count) was calculated for tracking the type of component represented.

The experiment outcomes would be impacted by the software application used, the details of families used in the BIM authoring tools and the hardware. To control the impact of these factors in the experiments, we utilized the fine level of detail in families used in the BIM authoring tool for the initial models, did not run any other processes on the hardware running the experiments, and used the same tools for the import/export process. The experiments were conducted on a machine with 2.4 GHz 2 processor CPU, 64 GB RAM, and Quadro K5200 GPU, and using Autodesk Revit as the BIM authoring tool, 3D Studio Max as the rendering tool, Unity 3D as the gaming engine, and MiddleVR as the solution between the gaming engine and the virtual equipment. There are plenty of BIM authoring tools, rendering tools and game engines on the market. For the purpose of this study (i.e. creating interactive virtual environments for the AEC industry), the authors selected the most widely used BIM authoring tool, Autodesk Revit, since it has a complete software eco-system built around

Revit, including rendering tools (i.e. 3D Studio Max) and design tools (i.e. AutoCAD). The file format within this software eco-system can be shared freely without any file type conversion, which requires less effort in the model transferring process. For the game engine, Unity3D is known for its short learning curve, which is the reason of more AEC professionals starting to integrate Unity3D in their workflow of creating virtual environments.

2.2. Description of measured metrics in the experiments

For the experiments, we utilized the longest path workflow (i.e. Workflow 2) to get a holistic view of the model characteristics on the BIM to VE conversion process. By varying the model complexity (i.e. building size, object count in each discipline, LGG), we conducted experiments to quantify the efficiency in each step of the conversion process, and assess the performance of resulting models in VE. A description of the metrics used in the experiments is provided in Table 2. The performance of the resulting VEs was assessed by tracking the model and component sizes, semantic information retained/lost, and component identity retained/lost in each step of the conversion process, as detailed in Table 2. To define the performance of resulting VEs, we measured the polygon and vertices count as metrics for defining model/component sizes in VE, and the object count for tracking the component identity retained in the process, and the attribute count to track the semantic information retained in the process. The logic of selecting these metrics can be tied to one of the objectives of this study (i.e. improving the user experience in the virtual environments). Frame per second (FPS) is one of the most widely used metric to measure the fluidity of a graphicly intensive application. It is well known that rendering large number of polygons can stress the computer system into producing low FPS, and causing the application to slowdown. When bringing design models to the virtual environments, it is essential to create a smooth experience for the users. As a result, the measurement of the file sizes and the number of polygons are vital to monitor the performance of the application.

For the efficiency of the conversion process, we measured the time it takes to export/import each of the 36 models into the corresponding tools. We particularly measured the time it takes to export a model from a BIM authoring tool, to import it to a rendering tool, to render materials in the rendering tool, to export from the rendering tool, and to import the model to a game engine.

The more geometric detail an object has, the more polygons and vertices it contains. While rendering an object on the screen, its polygon count impacts the CPU. If a scene rendered has more than 3 million vertices, it is likely that the VE will have performance issues (Unity Technologies 2016). No additional polygon reductions have been performed in the experiments on the models except when the models are generated with different levels of geometric granularities (i.e. coarse and medium).

3. Research results and discussions

The results of the experiments are provided in two sections as results related to (a) the effect of model complexity on the efficiency of BIM conversion to VEs, and (b) the effect of model complexity on the performance of resulting VEs.

3.1. Effect of model complexity on the efficiency of BIM conversion to VEs

The object count was a metric used to track the number of components from each discipline in the model. Table 3 shows the object count in each model per discipline across building sizes and LGGs at the beginning of the experiments.

Table 2. Measured parameters in the experiments and their descriptions

VE Performance	Model and component sizes	Vertices count Polygon count	Refers to vertex and polygon counts in a model, which relate to run time performance. A vertex is a point in the 3D space with XYZ coordinates. Two vertices connected by a line form an edge. Three connected vertices create the simplest polygon (a triangle). Vertices count refer to the total number of vertices that form meshes in a model, whereas polygon count refers to the total number of triangles that form components in a model.
		File size	Refers to the size of the file generated in each step of BIM to VE conversion process.
	Semantic information retained	Attribute count	Refers to the number of attributes retained after a BIM is exported from a model authoring tool.
	Component identity retained	Object count	Represents the number of components that maintained their identity in each LGG (i.e., still appeared as an object when geometric granularity is reduced).
Efficiency	Time (seconds)	Import /Export/ Rendering times	This metric includes the time it takes: a) to export a model from a BIM authoring tool; b) to import it to a rendering tool; c) to render material in the rendering tool; d) to export from the rendering tool; e) to import the model to a game engine.

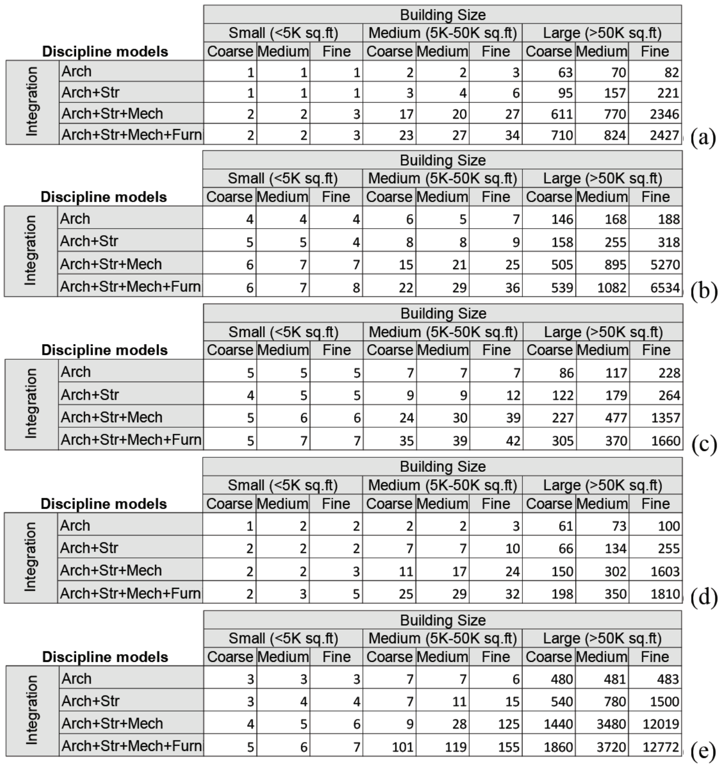
Table 3. Object count by discipline

Discipline models	Building size								
	Small (<5K sq. ft or 465 sq. m)			Medium (5K–50K sq. ft or 465–4645 sq. m)			Large (>50K sq. ft or 4645 sq. m)		
	Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Architectural	94	95	95	896	897	897	26,211	26,213	26,213
Structural	8	8	8	60	495	495	240	7,258	7,258
Mechanical	67	237	356	1,828	3,287	6,023	19,494	36,364	116,497
Furniture	43	43	43	1,536	1,536	1,536	2,305	2,305	2,305
TOTAL	212	383	502	4,320	6,215	8,951	48,250	72,140	152,273

As shown in Table 3, the model size remained in less than ten thousand physical components for small and medium sized buildings as the LGG and the type of components from each discipline changed. However, large building models had physical object counts ranging between 26K and 152K components. The large model selected was on the upper scale.

The results of the experiments for assessing the efficiency of the conversion process are provided in Figures 9 and 10. Using each model, we measured the time in five discrete points in the conversion process, corresponding to each figure below. The measurements are in seconds.

Small to medium models take half a minute to import and export between BIM authoring and rendering tools, and takes up to 2.5 minutes maximum to bring them in to the gaming environment. This process is quite efficient for small to medium size building models, regardless of what type of discipline models they integrate in them. However, the efficiency of the conversion process is significantly impacted when BIMs belong to large size buildings and they are presented with a fine level of geometric granularity. The last three columns on Table 4 belong to time it takes to import/export from the utilized tools in the VE conversion process. The coarse representations of geometry result in import/export times between one minute (for models containing single disciplines) and eight minutes (for models that integrate major disciplines) in BIMs of large size buildings. However, as shown in the last column of



Discipline models		Building Size								
		Small (<5K sq.ft)			Medium (5K-50K sq.ft)			Large (>50K sq.ft)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integration	Arch	1	1	1	2	2	3	63	70	82
	Arch+Str	1	1	1	3	4	6	95	157	221
	Arch+Str+Mech	2	2	3	17	20	27	611	770	2346
	Arch+Str+Mech+Furn	2	2	3	23	27	34	710	824	2427

(a)

Discipline models		Building Size								
		Small (<5K sq.ft)			Medium (5K-50K sq.ft)			Large (>50K sq.ft)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integration	Arch	4	4	4	6	5	7	146	168	188
	Arch+Str	5	5	4	8	8	9	158	255	318
	Arch+Str+Mech	6	7	7	15	21	25	505	895	5270
	Arch+Str+Mech+Furn	6	7	8	22	29	36	539	1082	6534

(b)

Discipline models		Building Size								
		Small (<5K sq.ft)			Medium (5K-50K sq.ft)			Large (>50K sq.ft)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integration	Arch	5	5	5	7	7	7	86	117	228
	Arch+Str	4	5	5	9	9	12	122	179	264
	Arch+Str+Mech	5	6	6	24	30	39	227	477	1357
	Arch+Str+Mech+Furn	5	7	7	35	39	42	305	370	1660

(c)

Discipline models		Building Size								
		Small (<5K sq.ft)			Medium (5K-50K sq.ft)			Large (>50K sq.ft)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integration	Arch	1	2	2	2	2	3	61	73	100
	Arch+Str	2	2	2	7	7	10	66	134	255
	Arch+Str+Mech	2	2	3	11	17	24	150	302	1603
	Arch+Str+Mech+Furn	2	3	5	25	29	32	198	350	1810

(d)

Discipline models		Building Size								
		Small (<5K sq.ft)			Medium (5K-50K sq.ft)			Large (>50K sq.ft)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integration	Arch	3	3	3	7	7	6	480	481	483
	Arch+Str	3	4	4	7	11	15	540	780	1500
	Arch+Str+Mech	4	5	6	9	28	125	1440	3480	12019
	Arch+Str+Mech+Furn	5	6	7	101	119	155	1860	3720	12772

(e)

Figure 9. Durations for import/export from each tool in the conversion process (measured in seconds): (a) duration of exporting a model from the BIM authoring tool; (b) duration of importing a model to the rendering tool; (c) duration of rendering a model in the rendering tool; (d) duration of exporting a model from the rendering tool; (e) duration of importing a model to the gaming engine

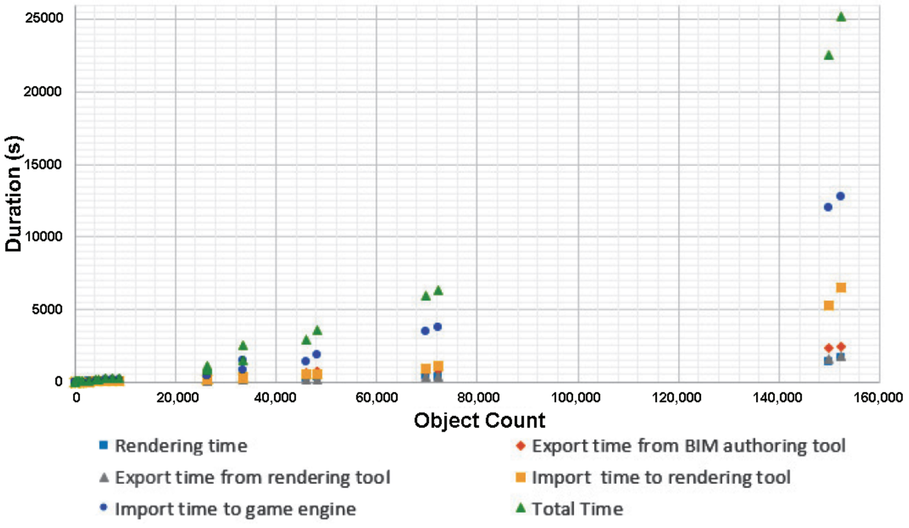


Figure 10. Time elapsed in the BIM to VE conversion steps

Figure 9, this time fluctuates drastically for fine geometric representations. The import/export times range from 12 minutes to 3.5 hours for fine LGG models of large buildings, and significantly impact the VE generation process.

The results of the efficiency experiments are also combined and plotted with respect to the object counts (Figure 10). The green highlighted region in Figure 10 corresponds to models that belong to small and medium size buildings, whereas the red highlighted region corresponds to fine LGG models that belong to large size buildings. The rest belongs to large size building models for coarse and medium LGGs. The difference between the two regions are significant, meaning that the process efficiency slows down to hours from a high conversion process of seconds.

BIM generated for large buildings and exported at fine granularity (with all disciplines represented) will result in a total of 7 hours in import/export times to bring a BIM to a gaming engine. However, the small and medium size building models (regardless of which LGG they were exported at and what types of components they contained), did not impact the conversion process and the total time was in the range of 14 seconds to 5 mins (green highlighted area in Figure 10). The order of magnitude changed from 5 minutes to 7 hours between BIMs of medium size buildings (<10,000 objects) and large size buildings (>100,000 objects). Exports from the rendering tool and exports from the BIM authoring tool were correlated hence overlapped in Figure 10. In general, import times were higher as compared to export times. In overall, the number of objects represented in each discipline and their LGG significantly impacted the import/export time in the conversion process.

3.2. Effect of model complexity on performance of resulting VEs

In order to compare the VE performance, one group of metrics measured model/component sizes through polygon/vertices counts and file sizes; and the other group of metrics measured the semantic information and component ID retained after the conversion process. The findings are discussed below accordingly.

3.2.1. Effect of model complexity on the model/component sizes

The performance of the VEs were influenced by the variations in the complexity of building information models. As object counts in each discipline model increased and components had fine level of geometric granularity (LGG) (compared to medium/course LGGs), the rendering and response performance of VEs significantly dropped. This effect was especially apparent in the BIMs for large size buildings. Tables 4 and 5 show the polygon and vertex count of each model exported from the BIM authoring tool, respectively. As expected, the vertices count (Table 5) was proportional to the polygons count. In general, the number of polygons were 1.75 times the number of vertices in the thirty-six experiment models tested. In order to avoid performance issues, a rule of thumb from gaming engines is that the vertex count per frame should be below 3 million for personal computers and below 200 thousand for mobile computers (Unity 2016). Given this rule of thumb, models that result in performance issues are marked with orange to red regions in Tables 4 and 5.

As the data in Tables 4 and 5 show, the three different LGGs did not impact the number of polygons and vertices for the BIMs that only contained architectural components, which means that the geometry of architectural components did not change between the different LGGs. Even the small changes observed in the medium model were not related to architectural components. The fine LGG for the medium size building models had around a thousand more vertices than medium and coarse LGGs. Table 4 shows that the count for structural discipline polygons increased for the models that belong to medium and large size buildings, when models are represented with medium and fine LGGs. This increase is attributed to the

Table 4. Polygon Count (Color coding: Colors towards green show no performance issues, colors toward red show performance issues)

		Model size								
		Small			Medium			Large		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integrated	Arch	64,992	65,016	65,052	107,675	107,675	109,692	4,143,847	4,150,621	4,782,357
	Arch+Struc	67,892	67,916	67,952	109,059	155,027	987,360	4,161,237	15,967,854	31,593,479
	Arch+Struc+MEP	76,151	84,806	134,223	253,194	574,509	2,193,449	12,314,478	25,718,571	78,404,498
	Arch+Struc+MEP+ Furn	147,714	156,368	205,785	3,692,323	4,014,090	5,634,595	21,358,442	34,762,535	87,448,462

Table 5. Vertex Count (Color coding: Colors towards green show no performance issues, colors toward red show performance issues)

		Model size								
		Small			Medium			Large		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integrated	Arch	37,959	37,975	37,999	57,032	57,032	58,101	2,238,013	2,241,657	2,582,157
	Arch+Struc	39,497	39,513	39,537	57,868	83,064	546,527	2,246,406	10,138,599	19,053,621
	Arch+Struc+MEP	43,871	48,629	75,539	135,475	307,712	1,222,378	6,001,450	14,759,930	43,368,427
	Arch+Struc+MEP+ Furn	85,635	90,393	117,303	2,713,191	2,885,635	3,801,163	12,148,177	20,906,657	49,515,154

steel columns and beams in these models. At a coarse level, steel columns and beams are not represented in 3D. BIM authoring tool transformed those components to 2D lines that were lost during the conversion since the game engine and rendering tools only import 3D objects from the rendered file. The small model was not impacted because it had concrete structural elements and this type of component was not affected by the level of granularity. Table 3 shows that the architectural and structural components did not present variations in object count between medium and fine LGGs. Mechanical discipline was the only discipline that impacted the object count between medium and fine LGGs. Most of the impact was caused by the loss of duct components when LGG was set to medium.

Mechanical and furniture components were responsible for the majority of the polygons and vertices in the models. Mechanical discipline was the discipline responsible by far for the largest object count (Table 3). For example, in the fine LGG model for large size buildings, the model had 116,497 objects, while the other three disciplines together had 35,777 objects. On the other hand, furniture had just 2,305 objects, but each component had in average 3924 polygons. The reason for changes in the object count values when models had coarse, medium and fine geometries was that Architectural and Structural components were mainly generic geometric representations (i.e. bounding box), whereas Mechanical and Furniture components were not regular geometries, had a lot of irregular details in geometry and had rounded surfaces. It was evident from the analysis that regardless of which discipline a component belonged to, when the geometry was represented with regular geometries without rounded or irregular representations, the components did not lose their identity in the model regardless of which LGG was used to export the models from BIM authoring tools.

Polygons and vertices count also dictated the size of the files generated on each phase of the conversion. Tables 6 and 7 show the file size for each LGG after the model was exported from the BIM authoring tool and after the model is exported from the rendering tool with textures reapplied. Based on this data, the file size increased by a factor of 2.5 after textures were embedded in it. Furthermore, as the LGG of the model increased, the file size increased by a factor of around 1.4 when models are exported with medium LGGs as compared to coarse LGGs, and increased by 1.6 times when models are exported with fine LGGs as compared to medium LGGs. These results are correlated with the polygon/vertex count analysis.

3.2.2. Effect of model complexity on semantic information and component identities retained

Regarding semantic information, the experiments included tracking attributes retained for a sample component from each model throughout the conversion process. For this purpose, a sample component was selected from each discipline model. An architectural wall, a structural column, a mechanical air diffuser and a furniture item (i.e. a table) were selected and traced for the attributes represented in the process of BIM to VE conversion. The components were chosen because they are among the most commonly seen components in each of the four categories, and they are easy to access in VE models. In addition to

Table 6. File size exported from the BIM authoring tool (MB)

	Discipline models	Building size								
		Small (<5K sq. ft or 465 sq. m)			Medium (5K–50K sq. ft or 465–4645 sq. m)			Large (>50K sq. ft or 4645 sq. m)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integrated	Arch	1.0	1.0	1.0	5.3	5.3	5.3	137.2	137.5	138.0
	Arch+Struc	1.1	1.1	1.1	5.7	6.5	10.3	142.9	315.9	510.1
	Arch+Struc+Mech	2.2	2.9	3.7	12.6	22.8	49.3	315.0	547.7	1,592.0
	Arch+Struc+Mech+Furn	2.2	2.9	4.0	20.5	30.7	57.2	333.0	567.7	1,610.8

Table 7. FBX file size exported from the rendering tool (MB)

	Discipline models	Building size								
		Small (<5K sq. ft or 465 sq. m)			Medium (5K–50K sq. ft or 465–4645 sq. m)			Large (>50K sq. ft or 4645 sq. m)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integrated	Arch	2.6	2.6	2.6	9.4	9.2	9.3	256.5	257.6	272.4
	Arch+Struc	2.7	2.8	2.8	20.3	23.5	47.6	259.2	642.2	1,125.0
	Arch+Struc+Mech	4.0	5.2	7.2	34.8	55.2	121.7	553.6	1,081.9	3,048.7
	Arch+Struc+Mech+Furn	6.0	7.2	9.3	130.2	150.5	217.7	803.4	1,331.7	3,298.0

this, a use case for FM operations has been defined based on earlier work of the authors (Yang, Ergan 2016). The use case refers to the geometric and semantic information needed for troubleshooting HVAC related problems. For this problem, basic information that is needed from a BIM to troubleshoot a given work order is defined as the space type, space ID, supply/return route sequence of components (i.e. connectivity between components and system they belong to), type of components (e.g. heating coil, supply air fan), function of each component, type of control system (pneumatic, electric), type of HVAC system (e.g. constant air volume, variable air volume, all air, air-water).

The findings of the analysis are provided in Table 8. Typically for every component inserted in a BIM authoring tool, several topological, geometrical, spatial, type and material information items are generated, as shown in Table 8. Out of all these semantic details, only identifier (name), *x,y,z* position, polygon mesh and material information are retained. The rest of the rich data generated in the BIM authoring tool is lost in the conversion process regardless of model component types, size or the level of geometric granularity. None of the required semantic information for the use case listed above is maintained in the VE.

Table 8. List of semantic information retained for components in the conversion process

Component type	Semantic information generated in the BIM authoring tool	Semantic information in the rendering tool	Semantic information in the gaming engine
Wall Column Air diffuser Furniture-Table	Globally unique ID Owner of component Name of component Definition of component Object type Component identifier Object Placement (Local *x,y* and *x,y,z* Cartesian points) Geometric representation Component type enumeration Connections to other components Contained in spatial elements Material and material layers associated with components Classification reference Surface styles Property sets common defined for the components (e.g. wall common, column common) System type for mechanical components	Name Dimensions Number of vertices Number of faces Parent geometry Number of children (geometry) Group/assembly	Name Position (*x,y,z*) Rotation (*x,y,z*) Scale (*x,y,z*) Material type Material size

Table 9. Object count in integrated models

	Discipline models	Building size								
		Small (<5K sq. ft or 465 sq. m)			Medium (5K–50K sq. ft or 465–4645 sq. m)			Large (>50K sq. ft or 4645 sq. m)		
		Coarse	Medium	Fine	Coarse	Medium	Fine	Coarse	Medium	Fine
Integrated	Arch	94	95	95	896	897	897	26,211	26,213	26,241
	Arch+Struc	102	103	103	956	1,392	1,392	26,451	33,471	33,472
	Arch+Struc+Mech	169	340	459	2,784	4,679	7,415	45,945	69,835	149,969
	Arch+Struc+Mech+Furn	212	383	502	4,320	6,215	8,951	48,250	72,140	152,274

Although model complexity did not have an effect on semantic information retained for each component in the conversion process, it had a significant impact on the component identities maintained in the conversion process. A significant contributor in component identity loss in the translation was due to the component geometries represented by discipline models. As shown in Table 9, when models contained components from architectural and structural disciplines, components mainly maintained their identities. The reason for majority of the architectural and structural components to maintain their identities is because of the bounding box representations of these components. This finding, however, was not valid when the structural models contained circular columns instead of rectangular columns, which resulted in a higher number of polygons and vertices in the exported models. For example, models for medium size building showed variations in object counts in structural components due to this problem. Two structural columns with identical height and close cross section area, but with rectangular and circular geometries result in 12 polygons/ 8 vertices and 771 polygons/ 388 vertices, respectively. Hence, geometric shapes had great influence on the component identity losses when geometries required large number of polygons to represent. As mechanical components had irregular geometries that required extensive number of polygons, the resulting models in coarse and medium LGGs lost almost half of the component identifies as models were exported in fine to medium, and medium to coarse LGGs.

The results in this section showed that model complexity played a significant role in the efficiency and performance of the model to VE conversion process. Also, they had a significant impact on the time it takes to convert a model to a VE, as shown in Figures 9 and 10. Given that mainly large size buildings models impacted from this conversion process, the next section provides a set of approaches and tests performed on improving the conversion process, mainly for models of large size buildings. The following section discusses approaches to retain semantic information in the VEs.

4. Approaches for improving the efficiency and performance of VEs generated from BIMs

BIMs for large buildings have a high potential of having efficiency and performance issues. Given also that the design cannot be changed for a given building, it is essential to change other aspects of the models to improve the performance of VEs and the conversion efficiency. Several optimization approaches (such as polygon reduction, occlusion culling, baked lighting, virtual occluding, and overdraw) are possible to run heavy models in an efficient way. The commonly used ones are (a) polygon reduction (geometric simplification) before and in the gaming environment, and (b) occlusion culling in the gaming environment (Germs, Jansen 2001; Selcuk *et al.* 1998). Baked lighting, virtual occluding and overdraw are advanced techniques usually used in gaming industry and not widely adopted in the AEC industry. Due to the complexity of implementing those techniques, they are not suitable for this test. The experiments and resulting discussions are provided below.

4.1. Polygon reduction before and in the gaming environment

To evaluate the implications of these optimization approaches in the resulting model performances, we used the same models utilized in the initial experiments. One of the major reasons for polygon increase in models is due to the utilization of circular geometries instead of rectangular shapes. As it was detailed in the previous section, among all the discipline models analyzed, structural and mechanical models had components had with irregular/ circular geometries and resulted in higher polygon/vertices counts – hence more prone to performance issues. Hence, both disciplines were analyzed in details for the impact of polygon reduction approaches for VE.

In order to better investigate the impact of curved components in the models, a separate medium LGG structural model (representing medium buildings' models) containing curved components was imported in the rendering software. The polygons count was 404,302 and the vertices count was 214,714. After that, all rounded columns were substituted by rectangular columns and the same process was applied to foundation components. The model with rectangular components had a significant decrease in vertices and polygons count, polygons count dropped to 79,005 and vertices count dropped to 47,951. Round representation of structural components resulted in 5 times more polygons and vertices counts. The transformation

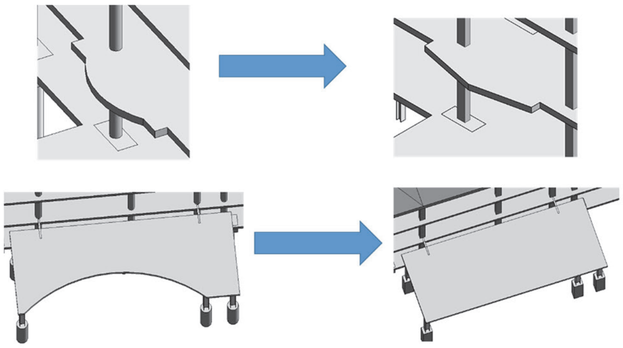


Figure 11. Transformation of structural component geometries into rectangular bounding boxes

was also performed on components with arc sections, as shown in Figure 11. The final geometries optimized had 49, 924 polygons and 32,195 vertices.

This geometric transformation however was invasive on the model geometry and represented the extreme polygon reduction. Hence, in addition to this, we performed polygon reduction in the form of using less segments on the curved surfaces by changing geometry curvature instead of turning a curve into a bounding box representation. Geometry curvature represents a categorization of surfaces with low to high areas of curvature, where high curvature areas requiring more polygons and have higher number in geometry curvature value. The algorithm utilized here is the progressive edge collapsing algorithm developed by Hoppe (1996). In pseudo terms, the algorithm works by collapsing vertices for any given edge. Assuming that *u*, *v* are two vertices connecting an edge, the algorithm searches and finds any triangles that include edge *u*, *v*, removes the triangles on the edge *u*, *v*, remove vertex *u* and update remaining triangles that has *u* as a vertex with *v*. This is an iterative process until a geometric curvature number is reached. The results in Table 10 show that the geometry curvature changed between 0–14 where 8 represent the default geometry and 0 represent the extreme optimization applied for polygon reduction to all component geometries. Results show that geometries are distorted after geometry curvature was set to 4 significantly.

The Mechanical model was also tested in this optimization process. The mechanical model with fine LGG for the large size building, representing the upper bound of the model complexity for mechanical models, was chosen for this analysis. The model had more than 5.35 million polygons and 2.37 million vertices. Since it was a complex model with varying geometric shapes, it was not easy to differentiate the components that contributed the most to the polygon/vertices counts. So, a filtering task was executed to identify components with the increasing number of polygons. The results of the filtering process are provided in Table 11.

Table 10. Polygon reduction by varying geometry curvature

Geometry curvature	Polygon count	Curvature count
14	2,517,373	1,281,790
12	2,365,377	1,205,476
10	1,114,218	571,072
8 (default)	404,302	214,714
6	243,669	135,655
4	72,489	44,920
2	32,981	20,969
0	27,407	17,415

Table 11. Filtering of components with respect to the number of geometric faces represented

# of Faces	Objects	Polygons	Vertices
12	8,455	101,460	67,640
>12	11,548	5,249,283	2,304,298
>100	4,051	4,918,514	2,117,627
>200	2,497	4,711,798	2,010,168
>500	390	4,093,218	1,668,905
>1,000	160	3,954,602	1,621,190
>2,000	98	3,863,627	1,573,645
>5,000	39	3,687,707	1,477,497
>10,000	24	3,597,816	1,428,753
>100,000	13	3,049,920	1,047,296
>200,000	11	2,845,392	891,066

Table 11 shows that the number of rectangular components using 12 faces (polygons) is 8,455. Although these components represented 42% of the total objects (with a total of 101,460 polygons), they only contributed to 1.9% of the total polygon count in the model. So, it is possible to argue that there are very complex components with high polygon counts in this model. Table 12 also shows that only 39 objects represented 69% of the total polygons presented in the mode – meaning that a group of components mainly contributed to the total number of polygon counts in the mechanical model analyzed. The most complex component in the mechanical model was a surface grid and had 258,672 polygons. Since the model had 11 components of the same type, the polygon count for these components were more than 2.8 million, representing 53% of the total polygons in the model. These eleven complex components were removed from the model to test if it would affect the file size and duration to import it into the game engine. While the original model exported from rendering tool had 255,374 KB, the new file generated without the complex components had 191,068. This means that the file size was reduced by 33%. The impact on duration time to import it in the game engine was even more significant. While the original mechanical model took 14 minutes and 10 seconds to be imported, the new model took only 7 minutes and 36 seconds; a decrease of almost 50%.

Filtering helps to detect components that with large number of polygons in a given model. Once such components are identified, polygon reduction can be performed on those components. Geometries could also be simplified by applying texture to reduce the polygon count of these geometries in the conversion process. This analysis shows that individual component representations have a significant impact on resulting model sizes and the VE performance. Careful examination and filtering of polygons on models and reduction of polygons by changing geometries can help to reduce problems in the downstream steps in the VE generation.

In the gaming environment, it is also possible to identify the regions of the model that have higher number of polygons to reduce. In the gaming environment, one can do overdrawing, meaning that models are viewed in a line format, with red outlines enclosing a single object, as shown in Figure 12. As more and more polygons are overlaid in a given region in the model, the lines turn into white regions, showing the areas of highly concentrated polygons. With overdraw, it is possible to identify such regions and focus on the components in those regions for polygon reduction.

4.2. Occlusion culling in the gaming environment

Another optimization technique to improve the performance of VEs is occlusion culling, which consists of leaving out objects in the rendering process that users do not see in a given view. For example, if the user is looking to a wall, the software should not render what is behind the wall. Occlusion culling boxes are drawn in the model, so if the user is inside a box, features outside that box are culled out from the rendering process. To study the effect of long views, we analyzed the CPU usage with and without occlusion culling. Figure 13 shows the views during runtime when the user is inside the model. Blue boxes in the left image of Figure 13 show the occlusion boxes defined for culling. For the location of the user given on the right figure, the CPU had a usage of 46.4 ms and 14,396 batches being processed without occlusion culling at that scene, processing all the far end components that are visible in that scene. Figure 13 middle image shows the components that are rendered for the same scene on the right with occlusion culling. The model is partially rendered based on

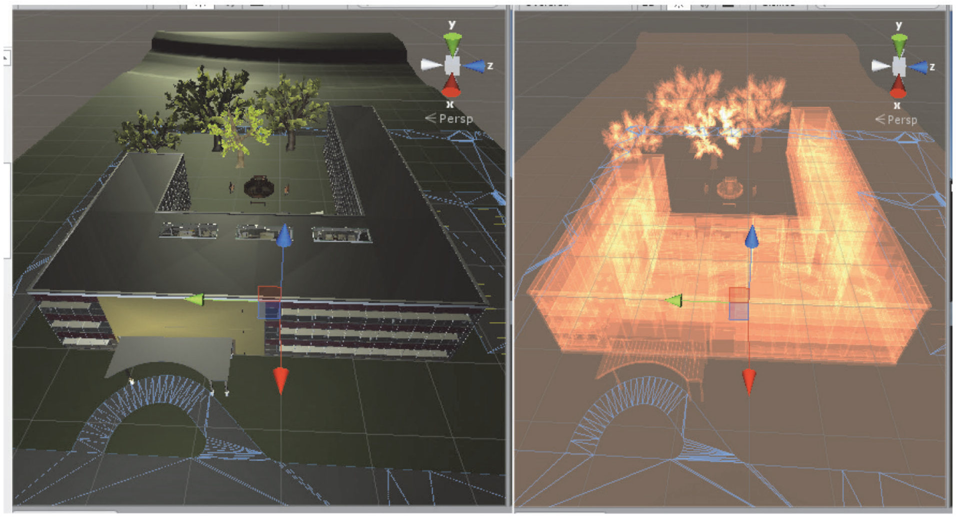
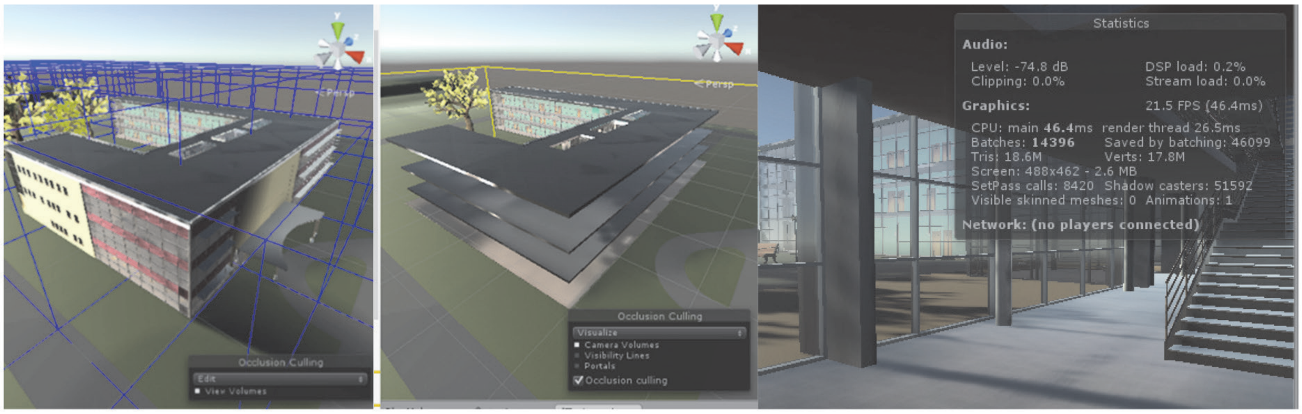


Figure 12. A model shown in regular and overdraw mode.
Areas of high polygon concentrations are highlighted with circles on the right image



Occlusion Culling
Edit
■ View Volumes

Occlusion Culling
Visualize
■ Camera Volumes
■ Visibility Lines
■ Portals
☑ Occlusion culling

```
                    Statistics
Audio:
  Level: -74.8 dB              DSP load: 0.2%
  Clipping: 0.0%               Stream load: 0.0%
Graphics:                      21.5 FPS (46.4ms)
  CPU: main 46.4ms  render thread 26.5ms
  Batches: 14396     Saved by batching: 46099
  Tris: 18.6M        Verts: 17.8M
  Screen: 488x462 - 2.6 MB
  SetPass calls: 8420  Shadow casters: 51592
  Visible skinned meshes: 0  Animations: 1
Network: (no players connected)
```

Figure 13. Occlusion culling off (left image) and on (right image) when a user is inside the modeled space

Table 12. List of modules developed for mapping IFC data to the gaming environment

Module name	Language	Description
Parser- IFC data to SQL Database	Python	Parses Step 21 files for individual components into an SQL database.
Connector- SQL database to Unity model	C#	Provides a connection between SQL database and the game engine to extract data for individual components.
Display	C#	Extracts data specific to the component from the database. Provides interactivity for components to allow users to display data on screen.

what the user can see in the scene. The CPU usage was of 35.5 ms and 11,528 batches, which means that at least 20% of computer usage was saved with this technique.

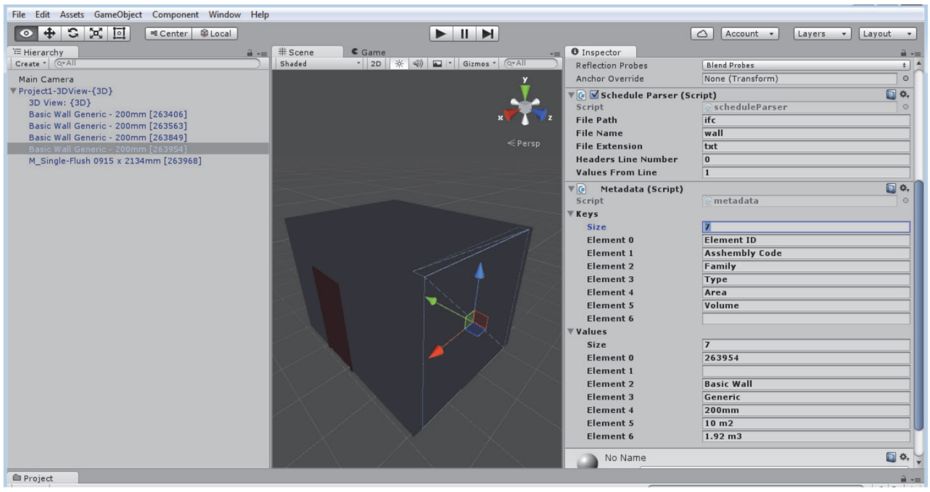
Since the occlusion culling is a rendering saving technique, a better performance can be achieved when long views are minimized. The more view blocks a model has, the less components need to be rendered. Therefore, avoiding long corridors for buildings through "zigzag" designs and having fewer transparent elements can help to improve runtime performance.

5. Approaches for improving semantic information storage in VE

One of the main purposes of utilizing VEs in FM operations is to have access to facility and component information with a spatial understanding. The loss of component's properties might not be an issue when visualization is the focus of the VE; however, for facility operations, the semantic information is critical and needed to solve problems. For example, to identify which valves are in a certain space or what area will be affected by turning off an air conditioner, the metadata and topological connections of each component is needed.

One effective solution to store the required semantic information in VEs is to use Industry Foundation Classes (IFC), a data standard with a comprehensive scope for exchanging building information models. This solution consisted of exporting an IFC file from the BIM authoring tool, and using the globally unique ID of each component to map the component information to the same component in the gaming environment. Each component instance in IFC has a globally unique ID and the gaming environment stores this information between brackets at the end of the name of that object. For this purpose, scripts were written in C#, Javascript, and Python to automatically process IFC files into a database and to pull data for respective components into the game engine dynamically. A list of modules developed by the authors to enable information mapping from IFC to the gaming engine is provided in Table 12.

These modules were implemented using the testbeds described in this research and modules proved to be effective to map the lost semantic information in the conversion process back to the geometric representations. Another solution that was explored in this research was also to use the globally unique ID as an identifier, however without the need to parse IFC files. This solution is limited in bringing component attributes only. In this solution, information needed for each component is exported from the BIM authoring tool as schedules into CSV files, then is parsed into the game engine through the globally unique ID. The advantage of this method was that only the needed information was brought into the game engine, optimizing the parsing process. An output of the parsing process with ID, assembly code, family type and name, assembly description and volume information for wall components is provided in Figure 14.



Main Camera
Project1-3DView-{3D}
3D View: {3D}
Basic Wall Generic - 200mm [263406]
Basic Wall Generic - 200mm [263563]
Basic Wall Generic - 200mm [263849]
Basic Wall Generic - 200mm [263954]
M_Single-Flush 0915 x 2134mm [263968]

Reflection Probes: Blend Probes
Anchor Override: None (Transform)

☑ Schedule Parser (Script)

Script	scheduleParser
File Path	ifc
File Name	wall
File Extension	txt
Headers Line Number	0
Values From Line	1

Metadata (Script)

Script	metadata
Keys	
Size	7
Element 0	Element ID
Element 1	Asshembly Code
Element 2	Family
Element 3	Type
Element 4	Area
Element 5	Volume
Element 6	
Values	
Size	7
Element 0	263954
Element 1	
Element 2	Basic Wall
Element 3	Generic
Element 4	200mm
Element 5	10 m2
Element 6	1.92 m3

No Name

Figure 14. Component information parsed and mapped to the component in the VE

Conclusions

This research is focused on the identification of impact of building information model complexity in the conversion process of BIMs to VEs. Several metrics were defined to represent model complexity and the resulting VE's performance evaluations. The performance of the resulting VEs was assessed by tracking the performance through polygon/vertex counts and file sizes, the capability to retain component identities and semantic information associated with them, and the time it takes to convert a model to VE. Model complexity has been defined with the number of physical objects represented in BIM per discipline, building size, and level of geometric granularity (LGG). Thirty-six BIMs, corresponding to different model complexities were used to evaluate the impact of model complexities in the conversion of BIMs to VEs.

The results of the experiments showed that models with components having irregular and circular geometric representations have performance and efficiency issues as compared to components with rectangular and bounding box representations. Typically, architectural and structural components do not have problems due to their geometric representations, whereas furniture and mechanical models generate majority of the problems due to their complex geometric representations. Especially, the problems are exacerbated when the level of geometric granularity is defined as fine for these models. Regarding building size, BIMs of buildings that fall under small and medium size (<5K square foot or 465 square meters) do not show performance issues and the conversion process is efficient. The authors acknowledge the lack of repetition in the 36 test projects, since the process of bringing design models to virtual environments can take up to 7 hours.

Regarding component identities, the information is retained when models are kept in any LGG for architectural and structural components with regular geometries. However, mechanical duct work and steel structural components lose their identities when exported at coarse and medium LGGs. This result was mainly attributed to cross section areas of components and their geometric representations. The research findings suggest that only the fine level of granularity should be used for converting BIMs to VE. In case certain components are not required in the VE, it is better to filter them individually instead of risking to mistakenly get components filtered with the medium and coarse LGGs. For semantic information about components, model complexity did not play a role, as all the semantic data is lost in the conversion process. The results should be interpreted with the given fact that geometric representations in BIMs are directly obtained from typical BIM authoring tools that the AEC industry utilize to create models. This study did not focus on alternative or ideal solutions of geometric representations that gaming engines can recognize, but aimed to analyze the challenges faced in the VE conversion process, given a model is available from a generic BIM authoring tool.

In summary, the model characteristics played significant impact on the efficiency and performance of the resulting VEs. Especially, BIMs of large size buildings experienced the negative impact as compared to small and medium size building models. To improve the performance of the resulting VEs, several approaches such as occlusion culling and polygon reduction were suggested and feasibility results were presented. For bringing semantic information back to the VEs, approaches for parsing and mapping IFC data and exports from BIM as .csv files were discussed.

As future works, this research could continue to explore more buildings with various curvatures in different building components to understand the impact of curved surfaces have to the VE performance. Research regarding to building size and corresponding component count could be carried out to examine the relationship between the two factors, and potential impacts on conversion process and VE performance. Another future direction could be to develop a plug-in in BIM authoring tools to simplify curved components and reduce the number of polygons of each element of the model, as polygon reduction was an effective way to improve the performance of resulting VEs. Finally, another direction of future research is exploring the options of using multiple platforms (i.e. BIM authoring tools, rendering tools, and game engines) in the process of bringing design models to virtual environments, since various tools might have different LGG and LOD settings that can affect the file size, polygon counts and component count.

Author Contributions

Dr. Ergan and Mr. Arruda conceived the study and were responsible for the design and development of the data collection. Mr. Zou, Mr. Arruda and Dr. Ergan were responsible for data analysis. Mr. Zou, Mr. Arruda and Dr. Ergan wrote the draft of the article.

Funding

This work was supported by the Defence Advanced Research Projects Agency (DARPA) under Grant D15AP00098.

References

Autodesk. 2017. *Detail levels in Revit* [online], [cited 21 May 2017]. Available from Internet: http://autodesk.com/view/RVT/2017

Behzadan, A. H.; Timm, B. W.; Kamat, V. R. 2008. General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering, *Advanced Engineering Informatics* 22: 90–105. https://doi.org/10.1016/j.aei.2007.08.005

Bern, M.; Eppstein, M. 1992. Mesh generation and optimal triangulation, *Computing in Euclidean Geometry* 1: 23–90. https://doi.org/10.1142/9789814355858_0002

Bille, R.; Smith, S.; Maund, S.; Brewer, G. 2014. Extending building information models into game engines, in *Proceedings of the 2014 Conference on Interactive Environment*, 2–3 December 2014, Newcastle, NSW, Australia, 1–8. https://doi.org/10.1145/2677758.2677764

Billger, M.; Heldal, I.; Stahre, B.; Renstrom, K. 2004. Perception of color and space in VR: a comparison between a real room and virtual reality models, in *Proceedings of SPIE – The International Society for Optical Engineering*, 2–3 August 2004, Denver, CO., USA.

Bowman, D.; Ray, A.; Gutierrez, M.; Mauldon, M.; Dove, J.; Westman, E.; Setareh, M. 2006. Engineering in three dimensions: Immersive VEs, interactivity, and 3D user interfaces for engineering applications, in *Proceedings of GeoCongress*, 26 February –1 March 2006, Atlanta, Georgia, 1–17.

Chen, H. M.; Hou, C. C.; Wang, Y. H. 2013. A 3D visualized expert system for maintenance and management of existing building facilities using reliability-based method, *Expert Systems with Applications* 40(1): 287–299. https://doi.org/10.1016/j.eswa.2012.07.045

Chen, J. C.; Thropp, J. E. 2007. Review of low frame rate effects on human performance, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 37(6): 1063–1076. https://doi.org/10.1109/TSMCA.2007.904779

CIC. 2014. *Computer integrated construction research group-workflow of exporting Revit models to Unity*. Penn State CIC Research Group.

Dalton, B.; Parfitt, M. 2013. *Immersive visualization of building information models*. Design Innovation Research Center Working Paper 6 [1.0]. Reading: Design Innovation Research Centre, University of Reading.

Du, J.; Shi, Y.; Zou, Z.; Zhao, D. 2017a. CoVR: Cloud-based multiuser virtual reality headset system for project communication of remote users, *Journal of Construction Engineering and Management* 144(2): 04017109. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001426

Du, J.; Zou, Z.; Shi, Y.; Zhao, D. 2017b. Simultaneous data exchange between BIM and VR for collaborative decision making, in *ASCE International Workshop on Computing in Civil Engineering 2017*, 25–27 June 2017, Seattle, Washington, USA, 1–8. https://doi.org/10.1061/9780784480830.001

Du, J.; Zou, Z.; Shi, Y.; Zhao, D. 2018. Zero latency: Real-time synchronization of BIM data in virtual reality for collaborative decision-making, *Automation in Construction* 85: 51–64. https://doi.org/10.1016/j.autcon.2017.10.009

Dünser, A.; Steinbügl, K.; Kaufmann, H.; Glück, J. 2006. Virtual and augmented reality as spatial ability training tools, in *ACM International Conference Proceeding Series*, 6–7 July 2006, Christchurch, New Zealand, 125–132. https://doi.org/10.1145/1152760.1152776

EIA. 2012. *A look at the US building stock: results from EIA's 2012 commercial buildings energy consumption*. US Energy Information Administration [online], [cited 21 May 2017]. Available from Internet: http://www.eia.gov/consumption/commercial/reports/2012/buildingstock/

Ergan, S.; Yang, S. 2017. Visualization support for facility operations and maintenance, in D. Willis, W. Braham, K. Muramoto, D. Barber (Eds.). *Energy accounts: Architectural representations of energy, climate and the future*. New York: Routledge, 142–148.

Ergan, S.; Yang, X. 2015. Value of 3D gaming engine based virtual models in understanding behaviors of facility operators during FM, in *CIB W78 Conference*, 27–29 October 2015, Eindhoven, Netherlands.

Evans, F.; Skiena, S.; Varshney, A. 1996. Optimizing triangle strips for fast rendering, in *IEEE Proceedings on Visualization*, 28–29 October 1996, San Francisco, California, 319–326. https://doi.org/10.1109/VISUAL.1996.568125

Figueres-Munoz, A.; Merschbrock, C. 2015. Overcoming challenges in BIM and gaming integration: the case of a hospital project, *WIT Transactions on the Built Environment* 149: 329–340. https://doi.org/10.2495/BIM150281

Gallaher, M. P.; O'Connor, A.; Dettbarn, J. L.; Gilday, L. T. 2004. *Cost analysis of inadequate interoperability in the US capital facilities industry*. NIST Report 04-867, NIST Advanced Technology Program, Information Technology and Electronics Office, Gaithersburg, Maryland.

Germs, R.; Jansen, F. 2001. Geometric simplification for efficient occlusion culling in urban scenes, in *Proceedings of WSCG*, 5–9 February 2001, Plzen, Czech Republic.

Gopinath, R.; Messner, J. I. 2004. Applying immersive virtual facility prototyping in the AEC industry, in *Proceedings of the Conference on Construction Applications of Virtual Reality (CONVR 2004)*, 14–15 September 2004, Lisbon, Portugal, 14–15.

Goulding, J. S.; Rahimian, F. P.; Wang, X. 2014. Virtual-reality based cloud BIM platform for integrated AEC projects, *Journal of Information Technology in Construction* 19: 308–325.

Hammad, A.; Wang, H.; Mudur, S. P. 2009. Distributed augmented reality for visualizing collaborative construction tasks, *Journal of Computing in Civil Engineering* 23(6): 418–427. https://doi.org/10.1061/(ASCE)0887-3801(2009)23:6(418)

Heydarian, A.; Pantazis, E.; Carneiro, J. P.; Gerber, D.; Becerik-Gerber, B. 2015. Towards understanding end-user lighting preferences in office spaces by using immersive VEs, in *Proceedings of Computing in Civil Engineering*, 21–23 June 2015, Austin, Texas, USA, 475–482. https://doi.org/10.1061/9780784479247.059

Hoppe, H. 1996. Progressive meshes, in *ACM SIGGRAPH Proceedings*, 1996, 99–108.

Johansson, M.; Roupé, M.; Tallgren, M. 2014. From BIM to VR – Integrating immersive visualizations in the current design process, in *Proceedings of the 32nd eCAADe Conference*, 10–12 September 2014, Newcastle upon Tyne, England, UK, 261–269.

Kang, J.; Ganapathi, A.; Nseir, H. 2012. Computer aided immersive VE for BIM, in *The 14th International Conference on Computing in Civil and Building Engineering*, 27–29 June 2012, Moscow, Russia.

Kasireddy, V.; Zou, Z.; Akinci, B.; Rosenberry, J. 2016. Evaluation and comparison of different virtual reality environments towards supporting tasks done on a virtual construction site, in *Construction Research Congress 2016*, 31 May – 2 June 2016, San Juan, Puerto Rico, 2371–2381. https://doi.org/10.1061/9780784479827.236

Kumar, S.; Hedrick, M.; Wiacek, C.; Messner, J. 2011. Developing an experienced-based design review application for healthcare facilities using a 3D game engine, *Journal of Information Technology in Construction* 16(6): 3–22.

Lee, S.; Akin, Ö. 2011. Augmented reality-based computational fieldwork support for equipment operations and maintenance, *Automation in Construction* 20(4): 338–352. https://doi.org/10.1016/j.autcon.2010.11.004

Lin, Y. C.; Su, Y. C. 2013. Developing mobile-and BIM-based integrated visual facility maintenance management system, *The Scientific World Journal*. Article ID 124249. https://doi.org/10.1155/2013/124249

Messner, J. 2006. Evaluating the use of immersive display media for construction planning, in *Intelligent Computing in Engineering and Architecture. EG-ICE 2006.* Lecture Notes in Computer Science, Vol. 4200. Springer, Berlin, Heidelberg, 484–491. https://doi.org/10.1007/11888598_43

Mozaffari, E.; Hammad, A.; El-Ammari, K. 2005. Virtual reality models for location-based facilities management systems, in *The 1st CSCE Specialty Conference on Infrastructure Technologies, Management and Policy*, 2–4 June 2005, Toronto, Canada, 132.1–132.10.

Nopachinda, S.; Ergan, S. 2016. Challenges in converting building information models into VEs for FM operations and user studies in the built environment, in *The 16th International Conference on Computing in Civil and Building Engineering (ICCCBE 2016)*, 6–8 July 2016, Osaka, Japan, 758–765.

O'Connor, J.; Davis, V. 1988. Constructability improvement during field operations, *Journal of Construction Engineering and Management* 114(4): 548–564. https://doi.org/10.1061/(ASCE)0733-9364(1988)114:4(548)

Pitt, M.; Goyal, S.; Holt, P.; Ritchie, J.; Day, P.; Simmons, J.; Robinson, G.; Russell, G. 2005. An innovative approach to facilities management in the workplace design brief: Virtual reality in design, *Facilities* 23(7/8): 343–355. https://doi.org/10.1108/02632770510600290

Pulaski, M. H.; Horman, M. J.; Riley, D. R. 2006. Constructability practices to manage sustainable building knowledge, *Journal of Architectural Engineering* 12(2): 83–92. https://doi.org/10.1061/(ASCE)1076-0431(2006)12:2(83)

Regan, M.; Ronald, P. 1994. Priority rendering with a virtual reality address recalculation pipeline, in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, 24–25 July 1994, Orando, FL, USA, 155–162. https://doi.org/10.1145/192161.192192

Revit API. 2017. *Detail level property representation for geometry extraction* [online], [cited 21 May 2017]. Available from Internet: http://www.revitapidocs.com/2018/887c4c25-fe14-2633-b84c-09d2f1279c9e.htm

Saeidi, S.; Rizzuto, T.; Zhu, Y.; Kooima, R. 2015. Measuring the effectiveness of an immersive VE for the modeling and prediction of occupant behavior, in *Proceedings of Sustainable Human-Building Ecosystems*, 5–6 October 2015, Pittsburgh, Pennsylvania, USA, 159–167. https://doi.org/10.1061/9780784479681.017

Savioja, L.; Mantere, M.; Olli, I.; Äyräväinen, S.; Gröhn, M.; Iso-Aho, J. 2003. Utilizing VEs in construction projects, *International Journal of IT in Construction* 8: 85–99.

Selcuk, A.; Gudukbay, U.; Ozguc, B. 1998. A survey of interactive realistic walkthrough techniques in complex graphical environments, in *Proceedings of the 13th International Symposium on Computer and Information Sciences*, 26–28 October 1998, Antalya, Turkey.

Shen, Z.; Issa, R.; Gu, L. 2007. Semantic 3D CAD and its applications in construction industry – an outlook of construction data visualization, in *Advances in Visual Information Systems.* Lecture Notes in Computer Science, Vol. 4781. Springer Berlin / Heidelberg, 461–467. https://doi.org/10.1007/978-3-540-76414-4_45

Shen, Z.; Jiang, L.; Grosskopf, K.; Berryman, C. 2012. Creating 3D web-based game environment using BIM models for virtual on-site visiting of building HVAC systems, in *Proceedings of Construction Research Congress*, 21–23 May 2012, West Lafayette, Indiana, USA, 1212–1221. https://doi.org/10.1061/9780784412329.122

Shi, Y.; Du, J.; Lavy, S.; Zhao, D. 2016. A multiuser shared VE for facility management, *Procedia Engineering* 145: 120–127. https://doi.org/10.1016/j.proeng.2016.04.029

Stell, J.; Worboys, M. 1998. Stratified map spaces: A formal basis for multi-resolution spatial databases, in *Proceedings of the 8th International Symposium on Spatial Data Handling*, 11–15 July 1998, Vancouver, Canada, 180–189.

Unity Technologies. 2016. *Optimizing graphics performance* [online], [cited 15 April 2016]. Available from Internet: http://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html.

Wang, X.; Dunston, P.; Skibniewski, M. 2004. Mixed reality technology applications in construction equipment operator training, in *Proceedings of the 21st International Symposium on Automation and Robotics in Construction*, 21–25 September 2004, Jeju, Korea. https://doi.org/10.22260/ISARC2004/0069

Wang, X.; Kim, M. J.; Love, P.; Kang, S. 2013. Augmented reality in built environment: classification and implications for future research, *Automation in Construction* 32: 1–13. https://doi.org/10.1016/j.autcon.2012.11.021

Woo, J.; Peterson, M. A.; Gleason, B. 2016. Developing a virtual campus model in an interactive game-engine environment for building energy benchmarking, *Journal of Computing in Civil Engineering* 30(5). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000600

Yan, W.; Culp, C.; Graf, R. 2011. Integrating BIM and gaming for real-time interactive architectural visualization, *Automation in Construction* 20(4): 446–458. https://doi.org/10.1016/j.autcon.2010.11.013

Yang, S.; Ergan, S. 2014. Evaluation of visualization techniques for use by facility operators during monitoring tasks, *Automation in Construction* 44: 103–118. https://doi.org/10.1016/j.autcon.2014.03.023

Yang, S.; Ergan, S.; Knox, K. 2015. Requirements of integrated design teams while evaluating advanced energy retrofit design options in immersive VEs, *Journal of Buildings* 5(4): 1302–1320. https://doi.org/10.3390/buildings5041302

Yang, X.; Ergan, S. 2016. Leveraging BIM to provide automated support for efficient troubleshooting of HVAC related problems, *Journal of Computing in Civil Engineering* 30(2): 04015023.1-10. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000492

Yerrapathruni, S. 2003. *Using 4D CAD and immersive VEs to improve construction planning*: MS thesis. The Pennsylvania State University, Pennsylvania.

Zhang, H. 1998. *Effective occlusion culling for the interactive display of arbitrary models*: PhD Dissertation. University of North Carolina, Chapel Hill.