Texture mapping and IFC material retrievement for virtual reality applications

Sebastian Duque Mahecha[1], Sergi Aguacil Moreno[1], Alexandre Dennis Stoll[1], Tiago De Sousa Pereira[1], Laurent Deschamps[1]
[1]Building2050 group, Ecole polytechnique fédérale de Lausanne (EPFL), Fribourg, Switzerland

Abstract

Most current software support the Industry Foundation Classes (IFC) transfer of Building Information Models (BIMs) to applications enabling Virtual Reality (VR) navigation (BuildingSMART, 2020; Kiviniemi, Tarandi, Karlshøj, Bell, and Karud, 2008; Poljanšek, 2017). However, a review of current industry practice and scientific literature shows a rupture between the information concerning the materiality of objects stored in the IFC model data and the texture mapping image-source determining the objects' appearance in VR environments. This document presents an application based on the *IfcOpenShell* library (IfcOpenShell, 2020), which provides a mapping table allowing to align the retrieved relevant IFC model data to any chosen set of image or pattern files. It produces a DAE file and a linked repertoire defining texture mapping onto model objects visualised in VR. The application also allows the automatic updating after modification of the original model.

Key Innovations

- Control of texture appearance of IFC model data
- Associativity between IFC model data material information and object's texture appearance
- Development towards IFC full data and texture integration for VR environments

Practical Implications

Among the various purposes of exploiting IFC data models for virtual reality applications are:

- Project visualization tool
- Lighting simulation
- Facility Management (FM) interface
- Internet of Things (IoT) device management.

Input file format is IFC (Industry foundation classes, versions 2.3 onwards). Output file format is DAE (digital asset exchange) compatible to a large number of game engines powering VR, visualisation, and simulation applications. We are currently developing a complement to the application allowing the re-introduction of the entire IFC data into the VR Environment.

Introduction

Successful experiments have been carried out in order to enable interactive IoT device management through immersive environments supported by Virtual-Reality (VR) and Augmented Reality (AR) (Chang, Dzeng, and Wu, 2018; Dalton and Parfitt, 2013; Jo and Kim, 2019; Müller, Aslan, and Krüßen, 2013; SwissLivingChallenge, 2017; Tang, Shelden, Eastman, Pishdad-bozorgi, and Gao, 2019; White, Cabrera, Palade, and Clarke, 2019). In these experiments, material texturing plays a key role in making the experience vivid and facilitating spatial orientation and object sight recognition. In parallel, Architecture, Engineering and Construction (AEC) professionals increasingly integrate into their practice Building Information Modelling (BIM) methodologies, notably relying on the IFC data model (BuildingSMART, 2020) as the open international standard (Kiviniemi et al., 2008; Poljanšek, 2017).

According to these methodologies, digital models synthesising informed-3D elements operate as central source of building data along the building lifecycle. In spite of the evident convenience of using such models for IoT device management, a review of current industry practice and scientific literature in achieving IFC-IoT integration shows that there is still a major difficulty in keeping together IFC parametric information and material textures. Thus, the general framework of this article is the analysis of different workflows to manage IoT interactive environments capable of preserving, from the original model, the exported IFC geometric and parametric information, as well as the possibility to preserve, or control, material-textures affected to the model's elements. More specifically, we describe an automated workflow IFC file > IFC verification definition correcting textures > Game Engine, allowing the integration of material-textures to an IFC2.3 export.

Virtual Reality (VR) plays more and more an important role for a variety of BIM usages. VR, being the exclusive visual media allowing simultaneously immersion and interaction experiences based on 3D models, serves well BIM purposes such as:

- Project promotion and visualisation, for which hyper-realistic rendering is of high importance.
- Natural and daylighting simulations, where accurate texturing play an important role.
- Facility management, and IoT device management, for which object data is fundamental. (Dalton and Parfitt, 2013)

To date, workflows available to import BIMs into VR environments grossly follow two possible paths:

https://doi.org/10.26868/25222708.2021.30541

- Path 1 : The original 3D model is directly imported from the native format (BIM supported software such as Blender, FreeCAD, Revit, ArchiCAD, etc.), case in which material-texture information contained in the model's native file is likely to be successfully restored in the VR application. This option is however limiting, as the formats supported by VR applications are necessarily chosen by game engine software providers.
- Path 2: The original model is converted into IFC format and then imported into the VR application, case in which a single open format (IFC) needs to be supported by game engine software providers in order to manage all possible exchanges. The limitation of this option is that the original material texturing information is lost.

Figure 1 illustrates the economy of programming effort that a full adoption as universal standard of the open data exchange format IFC may imply for all software providers involved. Paradoxically, in practice, and probably due to a commercial interest in positioning their own format as the universal standard, the market strategies of certain software companies seem to favour exchanges based on native formats.

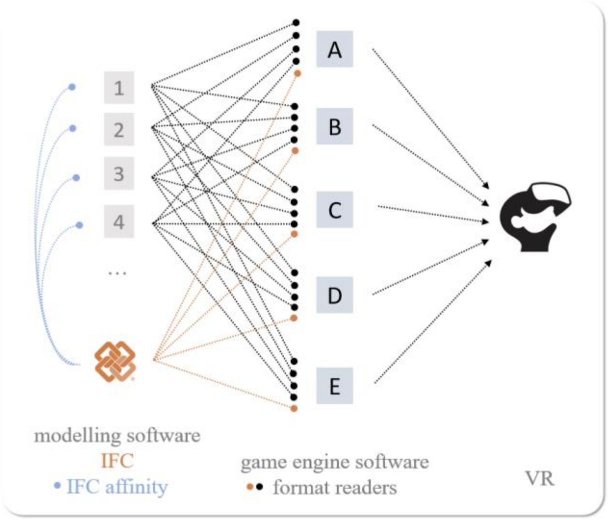

Figure 1: software theoretical affinity in BIM to VR workflows. (1, 2, 3, 4 …) Modelling (BIM) software, (A, B, C, D, E) Game engine software.

In turn, the IFC, which offers a well-structured data storage of industry-related entities and properties, provides remarkable service in coordinating multidisciplinary models in today's practice. However, this open format appears less effective for other BIM applications involving material texturing definitions. The fundamental reason for this is that the current version (IFC 4.3) does not suit a robust textures support. Until now, BuildingSMART International (bSI) have given priority to design coordination, including quantification and geometric coherence, leaving for the object's appearance a simple colour support. As stated by J.W. Ouellette, Technical Room International Projects Coordinator for BuildingSmart International (bSI): "The ISG along with current and previous bSI certification regimes have never pursued wide implementation, testing and certification of textures, even though they have been supported in IFC for some time. […] Effort has gone into coordination support (including quantity take off and spatial validation workflows) with less complex colour support." (BuildingSMARTforum and Ouellette, 2019a; BuildingSMARTforum, Ouellette, and Moult, 2019).

In conclusion, an IFC file does not contain information about the textures of the objects in the model, even if the source model has well-defined ones. Thus, when importing an IFC file to a VR environment, the user is confronted to an unsolvable dilemma: to keep all the properties specified in the IFC, or to transform it into a format supporting textures, at the expense of losing most IFC structured properties. This dilemma may disappear in time, since, according to what was announced by bSI, textures will be supported in the next IFC major release: the IFC 5 (BuildingSMARTforum and Ouellette, 2019b). Nevertheless, there is no certainty of how soon actual support will be available. Considering that, to the date, most modelling software is unable to satisfy import/export IFC requirements and has hardly made it to the IFC 2.3 certification (BuildingSMART, 2020)), even if the needed updates are efficiently implemented by bSI, IFC software certification may still delay the actual use of IFC 5 new benefits.

Given this context, we have opted to search the different alternatives to use effectively an IFC 2.3 file in order to obtain a VR-enabled BIM digital model enriched with textures. In the following pages, we will describe a procedure to retrieve material information contained in IFC model data, in order to align this information to texture mapping based on a set of images or patterns chosen by the user.

Methods

Modelling and VR game engine software choice

Outside the scope of this paper is:

- A full review of IFC export quality of different modelling software.
- A full review of IFC import quality of different VR game engine software.

For all tests presented, we used the digital model of an office space with basic architecture elements (floor, ceiling, walls, and windows) as well as few pieces of furniture (a desk, a chair and a lamp). For practical reasons (mainly skills and educational licence availability), we used Revit 2020 (Autodesk, 2019) which, at the date had a 2.3 release certification for IFC export (BuildingSMART, 2020).

https://doi.org/10.26868/25222708.2021.30541

Figure 2: Source model in wireframe and render views

Using the software standard palette, we assigned to each element of the model a material that we judged realistic enough. It is important to note that our interest is to analyse the transfer of texture information, rather than texture realism itself. Thus, we did not put special care into the qualitative texturing result displayed by the modelling software.

From this model, we exported a standard IFC 2x3, following the software- provider recommended procedure, which included the pre-installation of a set of Shared IFC parameters. We obtained a satisfactory IFC file, which included the elements' basic proprieties.

We chose to work with Unity version 2019.4.17f1 (UnityTechnologies, 2005), as our experimental VR game engine software. Based upon internet ratings, we stablished a list of the principal VR game engines software, observing in particular their IFC compatibility. Table 1 shows the list of preselected software.

Table 1: Top 5 VR / Game engines in the market and his compatibility with IFC according to (G2.com Inc, 2021)

	Software	Rate*	IFC	via Plugin
1	Unity	91	Yes	Pixyz
2	3ds Max Design	79	Yes	IfcMax
3	Unreal Engine (UE4)	68	Yes	Datasmith
4	Maya	42	-	-
5	CryEngine	30	-	-

(*) G2 Satisfaction Score *"calculated by a proprietary algorithm and this score is normalized for each Highest Rated list, meaning the scores are relative to other products in the list."*

Besides Unity's evident popularity, Unity was satisfactory to our research due to its strong VR performance (even in free version) and its IFC compatibility. In addition, the engine also supports quite well the live-link between assets and asset-related data, a quality that suited well a parallel research on VR-enabled building data management, in which we are currently involved.

IFC Open Shell

To import the IFC file directly into Unity, we initially used the Pixyz plugin for Unity (Pixyz software, 2020). Pixyz reads, converts and optimises several formats, including IFC, in order to import 3D CAD, point cloud and mesh models into Unity.

The plugin efficiently handles IFC model data, in the sense that all IFC proprieties of de model elements can be retrieved and exploited in the VR environment. That is the case for the IFC building structure (Site/ Building/ Floor/ Element) as well as the IFC Property sets (PSets) defined in the source modelling software. However, for the reasons explained in the introduction, the obtained objects appearance was expectedly devoid of any mapping texture set at origin (Figure 3).

After a number of import iterations we reached a conclusion confirmed by reports of other experiences (Sultan, 2020): in order to incorporate textures to objects contained in the IFC file, the use of a third application mediating between the IFC and the VR game engine was definite.

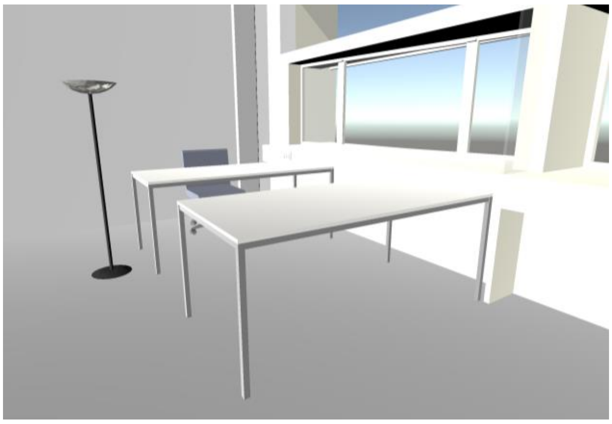

Figure 3: Model in Unity after import via Pixyz

We chose to work with the IfcOpenShell schema: an open source (Lesser General Public License (LGPL)) software library to work with IFC files. (IfcOpenShell, 2020). This library, allows accessing the IFC data model, associating a parameter to a new data structure exportable in a new format. We used the IfcOpenShell library in order to add texture information, in two steps:

- IFC conversion into DAE
- IFC data reintegration into the game engine

IFC conversion into DAE

Our first step was to follow a workflow (Figure 4), using the IfcOpenShell to convert the IFC into the Digital Asset Exchange (DAE) format, loosing most of IFC information, but gaining the possibility to control material textures.

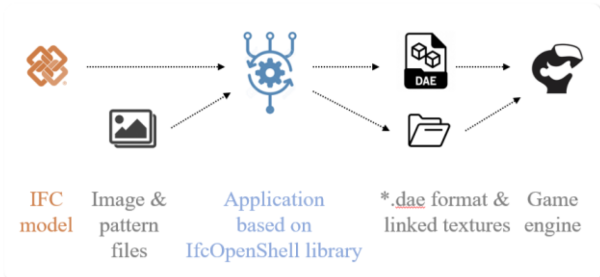

Figure 4: Workflow converting IFC file into DAE in order to add textures into a VR environment

The DAE format is suitable for various graphic purposes, such as images, textures, and 3D models. *.dae is the extension of files based on the 3D Asset Exchange Schema: COLLADA (Collaborative Design Activity), which defines an XML-based schema to transport 3D assets between applications enabling diverse 3D authoring and content processing tools to be combined into a production workflow, including: geometry, shaders and effects, physics, animation, kinematics, and multiple representations of the same asset. (Sony Computer Entertainment - Khronos Group, 2004).

On a first phase, we tried the open source software Blender and BlenderBIM, as the latter is able to import IFC model data into Blender, including geometry, object

https://doi.org/10.26868/25222708.2021.30541

properties and IFC building structure (Blender Foundation, 2019). In addition, Blender had recently implemented improved support to create an external link to material information for mapping textures, lighting simulation and some additional functionalities (BuildingSMARTforum et al., 2019).

Using Blender's interface, the procedure consists in re-assigning to each material present in the IFC model, a texture from a user-defined source of images or patterns. Once a texture (ex. wood) affects a given material, the texture is systematically applied to all model elements affected by this material. For practical reasons, we decided to move to a second phase in which we created our own definition for texture mapping, leaving unexplored the alternatives offered by Blender in relation to external referenced textures. Nevertheless, it is worth adding that, as far as we explored, we could not find the way to control the texture mapping settings to manage a satisfactory rendering outcome. Since the same image may be applied to faces being affected by the same material but having despair analogies (i.e., the working surface vs the leg of a wooden desk), the resulting image pattern on the object-surface would be deformed for at least one of these faces.

The procedure we followed proved highly inefficient in relation to changes and updates in the original IFC exported-model, since at each new import of the IFC, the whole process of reassigning textures needed to start all over again.

Based on the IfcOpenShell library, we created an application serving as interface for the user to match IFC material information and textures. This interface presents the user to a couple of information sources: "Images" and "Materials" (Figure 5).

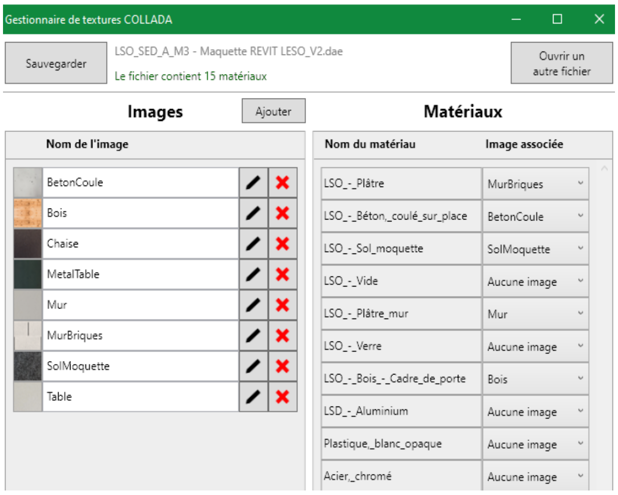

Figure 5: Interface of our mapping application based on IfcOpenShell library

The Image content, to the left, gathers all images and patters that the user has added by simple drag-and-drop or via the system's file explorer. The "Material" section consists on a double-column mapping table: the left column, lists all materials retrieved from the selected IFC file. The right column displays, in the form of dropdown menus, all the texture files the user has added to the "Images" section.

Once the user has matched all materials retrieved respectively to a desired texture, the conversion is ready to take place. The application produces a DAE file with geometry information of the IFC model and a folder of textures linkable from the game engine software. Once imported, the right textures will be mapped onto the right elements (Figure 6).

IFC data reintegration into the game engine

In the DAE file produced by our application, each model element retains only one IFC parameter: the IfcGloballyUniqueId. As said above, no other IFC data, besides geometry and certain element properties such as material, is transferred into the VR environment. Thus, the next step in our research was to re-integrate the missing IFC information into the VR environment. This is a work in progress, so the procedure will be discussed later.

Results

The results obtained are satisfactory, in the sense we have managed to control mapping textures using an IFC model. Moreover, our current progress shows that we will be able to restore the IFC information lost by the conversion into DAE format. In any case, the transmission of data along the workflow remains associative.

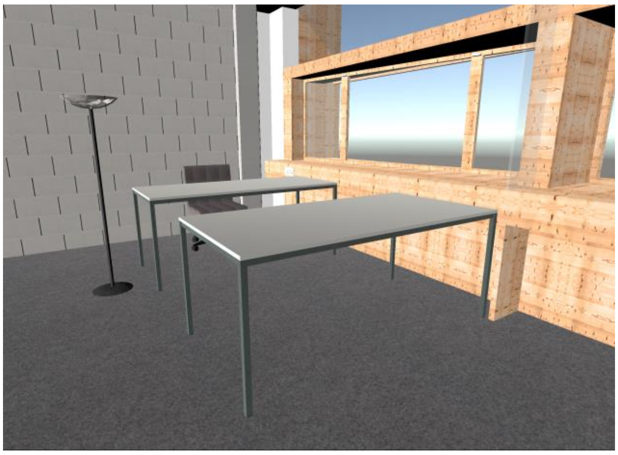

Figure 6: Model in Unity after import via the IfcOpenShell library application

That is, our definition is capable to automatically transmit and update the eventual changes in the original digital model. Thus, even though it is not formally an IFC what we exploit in the VR environment at the end of the chain, the IFC data, as structured information, remains unaltered and retrievable. In more detail:

Mapping textures

The application allows users to affect each material defined in the IFC file, using an image or pattern file of their choice. This texture is projected onto each face belonging to every model element affected by the specified material.

Through the interface, the user can also define the size in meters (height and width) to which the image will be scaled up / down when projected and repeated onto the

https://doi.org/10.26868/25222708.2021.30541

model's surfaces. In this way, the pattern will remain coherent, in particular when applied to faces having remarkably different proportions. The user in this step can as well define the image insertion point, as well as its orientation (Figure 7).

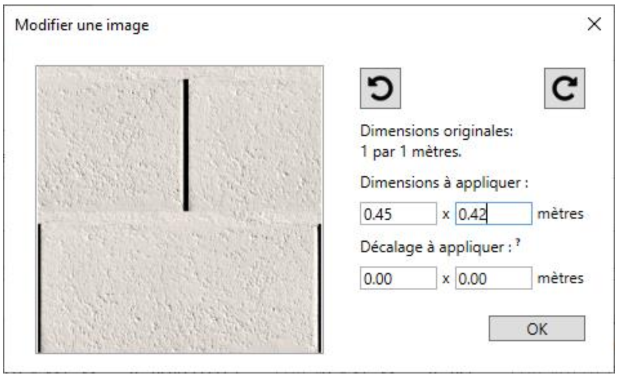

Figure 7: Application's Texture editing option

We have succeeded to apply a diffuse mapping (projection of pixels into the surfaces). However, we are currently working to integrate other mapping-operations in order to enrich the model with other material properties such as reflectiveness, specularity and rugosity.

Retrieved IFC Data

The application retrieves and transmits a portion of the IFC data to the DAE file, adding mapping texture information. Besides geometry, the DAE file associates elements to their IFC globally unique identity (IfcGloballyUniqueId). This piece of information is the key to re-associate the element and the related set of properties contained in the IFC file. Our team is currently working on a second application to obtain this (Figure 8).

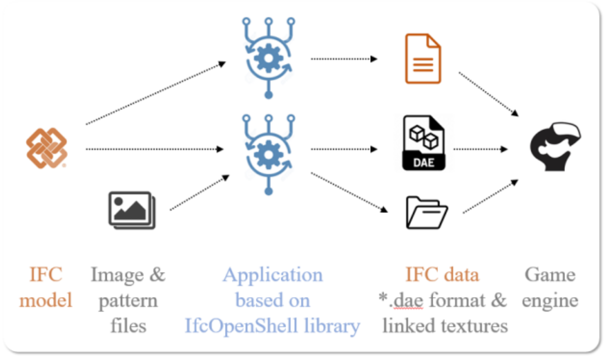

Figure 8: Workflow splitting IFC file information in order to add full IFC data and textures into a VR environment

The second application, as well based on the IfcOpenShell library, produces an IFC-information extract, which re-directs all inherited properties associated to the unique ID of each element. Once imported into the VR environment, this IFC properties will be linked to the element, restoring in this way the information contained in the original IFC file.

The DAE file, besides most game engine software already compatible (Unity, 3Ds Max, Maya, CryEngine, Game Engine, etc), can be as well imported into building information modelling software such as Blender, FreeCAD, or ArchiCAD.

Model update

As said, our definition is capable to update automatically the eventual modifications made to the original digital model. During the evolution of a project, AEC professionals need to produce various versions of the same model in order to integrate corrections and design evolution. Users may add, remove or transform elements; eventually producing a new IFC version that may replace the one imported into the VR environment.

When the user reloads this new version into our application, it will re-establish the links between IFC materials and texture sources as defined in the last use with the previous file version. This means that, apart from new materials added to the digital model of origin, which necessarily will need a manual update via the interface, all previously assigned materials will be automatically updated and the files produced will be ready for importation into the VR environment.

Discussion

We have explained the procedure to solve a problem that may no longer exist in the years to come. That is, the IFC 5 (BuildingSMART international, IFC schema 5), which is expected to be launched in the near future, will seemingly support mapping textures, so fully integrated "textured-IFC" workflows between BIMs and VR environments will be available.

However, the practical implementation of IFC 5, including eventual release addendums, modelling software certification, and its integration in the current practice and construction standards, may take yet several years before it is a reality.

In addition to this, the definition presented, together with the methodological path we described, raises new questions and opens new fields for exploration.

In the first place, we would like to point that, although our methodology did not explored further into different alternatives using BlenderBIM's improved support for externally defined materials, this application may well consolidate itself into a single open-source platform supporting modelling, IFC management and VR compatibility (BuildingSMARTforum et al., 2019).

Second, we would like to discuss a possibility that our research has opened for us: the exploitation of texture-integrated IFCs for Simulation modelling (SIM). Textures are more than a visual artifice to produce hyper-realistic rendering effects.

Texture mapping serves not only to imprint realism to objects in a digital model, but also to manipulate a variety of surface characteristics, such as colour, transparency, specularity, luminosity, bumpiness, and displacement. Such characteristics are fundamental to other BIM usages, such as modelling simulation concerning natural and artificial lighting, light reflectance, solar irradiance, sound propagation, physiological perception of space, etc. Currently, the IfcOpenShell library supports IFC-SPF files. In the future, when support for IFC-XML and IFC-

https://doi.org/10.26868/25222708.2021.30541

ZIP will be added (IfcOpenShell, 2020), we may be able to add to our application the possibility to generate, besides DAE files, XML files.

This development, namely the method of adding textures via an IfcOpenShell library IFC-XML schema, may well indicate ways in which we could exploit IFC's embedded material properties to retrieve physical material information for simulation modelling applications, such as DesignBuilder, Relux, Lesosai, EnergyPlus, and others (DesignBuilder, 2020; E4tech, 2020; Relux Informatik AG, 2021; US Department of Energy (DOE), 2020). This may open a new branch to our workflow (Figure 9).

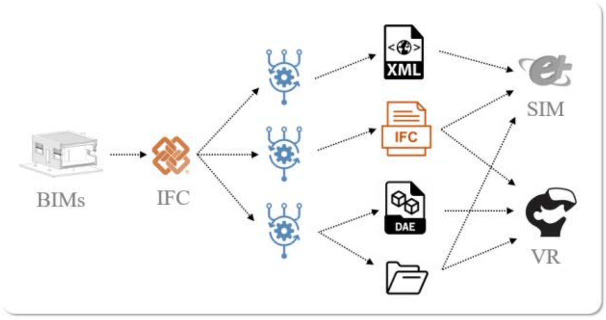

Figure 9: Three applications based on IfcOpenShell library to exploit IFC models in VR and SIM environments

Conclusion

First, we have explored the importance of the IFC standard as central data model within open BIM workflows. Second, we have studied the relevance of texture information in certain BIM usages, such as VR immersive and interactive experiences. Third, we have discussed the problematics of transferring together texture and IFC information from BIMs into VR environments.

In order to solve this problem, via an IfcOpenShell (IfcOpenShell, 2020), through an application of our own making, we have extracted from the IFC data model, the geometry, the material information and the unique IFC ID for each element in the model.

Using these three elements, we have created a DAE file linked to a set of user-defined textures, which is importable to most engine software supporting VR navigation. The workflow is useful, especially because it allows automatic updates, following the modification of the original BIM model. We have also discussed the application improvement that we are currently developing in order to re-integrate all IFC properties into the VR environment. Our progress shows that soon we will have a VR enabled fully integrated texture-supported IFC application. Finally, we have opened a new theme for research involving the integration of IFC-XML supported format, which, hypothetically, may allow applying our definition into a texture supported-IFC Building Simulation Modelling.

References

Autodesk. (2019). Autodesk Revit Software. Retrieved from https://www.autodesk.com/

Blender Foundation. (2019). BlenderBIM. Retrieved January 28, 2021, from https://www.blender.org/

BuildingSMART. (2020). Industry Foundation Classes (IFC). Retrieved January 28, 2021, from https://technical.buildingsmart.org

BuildingSMARTforum, and Ouellette, J. (2019a). *IFC - Is it possible to attach texture mapping information to ifc files?*

BuildingSMARTforum, and Ouellette, J. (2019b). *IFC - Problem about Elements with Texture*.

BuildingSMARTforum, Ouellette, J., and Moult, D. (2019). *IFC - Problem about Elements with Texture*.

Chang, K. M., Dzeng, R. J., and Wu, Y. J. (2018). An automated IoT visualization BIM platform for decision support in facilities management. *Applied Sciences, 8* (7).

Dalton, B., and Parfitt, M. (2013). Immersive Visualization of Building Information Models. *Design Innovation Research Centre working paper 6, (6)*, 1–20.

DesignBuilder. (2020). DesignBuilder software v.6. Retrieved from http://www.designbuilder.co.uk/

E4tech. (2020). LESO-SAI software. Retrieved from http://www.lesosai.com/

G2.com Inc. (2021). The Top 6 VR Game Engine Software. Retrieved January 29, 2021, from https://www.g2.com/

IfcOpenShell. (2020). IfcOpenShell, the open source ifc toolkit and geometry engine. Retrieved January 28, 2021, from http://ifcopenshell.org/

Jo, D., and Kim, G. J. (2019). AR enabled IoT for a smart and interactive environment: A survey and future directions. *Sensors 19*, (19).

Kiviniemi, A., Tarandi, V., Karlshøj, J., Bell, H., and Karud, O. J. (2008). Review of the Development and Implementation of IFC compatible BIM Executive Summary. *ERA Build*, 1–2.

Müller, L., Aslan, I., and Krüßen, L. (2013). GuideMe: A mobile augmented reality system to display user manuals for home appliances. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8253 LNCS*, 152–167.

Pixyz software. (2020). Pixyz plugin for Unity. Retrieved January 28, 2021, from https://www.pixyz-software.com/plugin/

Poljanšek, M. (2017). *Building Information Modelling (BIM) standardization*. JRC Techincal Reports.

Relux Informatik AG. (2021). Relux software. Retrieved January 29, 2021, from https://reluxnet.relux.com/

Sony Computer Entertainment - Khronos Group. (2004). COLLADA.

https://doi.org/10.26868/25222708.2021.30541

Sultan, A. (2020). Revit to Unity with textures. Retrieved January 29, 2021, from https://vrforcad.com/revit-to-unity-with-textures

SwissLivingChallenge. (2017). NeighnourHUB. Retrieved January 29, 2021, from http://www.swiss-living-challenge.ch/

Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-bozorgi, P., and Gao, X. (2019). Automation in Construction A review of building information modeling (BIM) and the internet of things (IoT) devices integration : Present status and future trends. *Automation in Construction 101*, 127–139.

UnityTechnologies. (2005). Unity cross-platform game engine. Retrieved January 28, 2021, from https://unity.com/

US Department of Energy (DOE). (2020). EnergyPlus. Retrieved January 21, 2021, from https://energyplus.net/

White, G., Cabrera, C., Palade, A., and Clarke, S. (2019). Augmented reality in IoT. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11434 LNCS*, 149–160.

https://doi.org/10.26868/25222708.2021.30541