Project Acronym:	**BIMERR**
Project Full Title:	**BIM-based holistic tools for Energy-driven Renovation of existing Residences**
Grant Agreement:	**820621**
Project Duration:	**42 months**

DELIVERABLE D5.3
Innovative Scan-to-BIM tools for Automated BIM v1

Deliverable Status:	**Final**
File Name:	**BIMERR DELIVERABLE D5.3_v1.0.docx**
Due Date:	**31/08/2020 (M20)**
Submission Date:	**26/08/2020 (M20)**
Task Leader:	**UEDIN (T5.2)**

Dissemination level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

The BIMERR project consortium is composed of:		
FIT	Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Germany
CERTH	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis	Greece
UPM	Universidad Politecnica De Madrid	Spain
UBITECH	Ubitech Limited	Cyprus
SUITE5	Suite5 Data Intelligence Solutions Limited	Cyprus
HYPERTECH	Hypertech (Chaipertek) Anonymos Viomichaniki Emporiki Etaireia Pliroforikis Kai Neon Technologion	Greece
MERIT	Merit Consulting House Sprl	Belgium
XYLEM	Xylem Science And Technology Management Gmbh	Austria
UCL	University College London	United Kingdom
CONKAT	Anonymos Etaireia Kataskevon Technikon Ergon, Emporikon Viomichanikonkai Nautiliakon Epicheiriseon Kon'kat	Greece
BOC	Boc Asset Management Gmbh	Austria
BX	Budimex Sa	Poland
UOP	University Of Peloponnese	Greece
UEDIN	The University of Edinburgh	United Kingdom
NT	Novitech As	Slovakia
FER	Ferrovial Agroman S.A	Spain

Authors List

Leading Author (Editor)				
	Surname	First Name	Beneficiary	Contact email
	Valero	Enrique	UEDIN	e.valero@ed.ac.uk
Co-authors (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Bosché	Frédéric	UEDIN	f.bosche@ed.ac.uk
2	Mohanty	Dibya	UEDIN	d.mohanty@ed.ac.uk
3	Michal	Ceklarz	UEDIN	s1762931@ed.ac.uk
4	Fenz	Stefan	XYLEM	fenz@xylem-technologies.com
5	Rovas	Dimitrios	UCL	d.rovas@ucl.ac.uk

Reviewers List

List of Reviewers (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Demeter	Dominik	NT	demeter@novitechgroup.sk
2	Tsakiris	Thanos	CERTH	atsakir@iti.gr
3				
4				

Revision Control

Version	Author	Date	Status
0.01	UEDIN	18.03.2020	Setup (ToC)
0.1	UEDIN	10.07.2020	Section 1 + Minor update
0.2	UEDIN	24.07.2020	Section 2, 3, 4 and Executive Summary
0.3	UEDIN	28.07.2020	Section 5 and Conclusion
0.4	UEDIN	30.07.2020	Sent for internal review
1.0	UEDIN	26.08.2020	Submission to the EC

Table of Contents

List of Figures

List of Tables

ACRONYMS

Acronym	Meaning
1LSB	First Level Space Boundary
2LSB	Second Level Space Boundary
API	Application Programming Interface
ARIBFA	Augmented-Reality Enabled In-situ Building Feature Annotation tool
ASCII	American Standard Code for Information Interchange
BICA	Building Information Collection Application
BIF	BIMERR Interoperability Framework
BIM	Building Information Modelling
BIMERR	BIM-based holistic tools for Energy-driven Renovation of existing Residences
CC	CloudCompare
CERTH	The Centre for Research & Technology, Hellas
COCO	Common Objects in Context
EU	European Union
GDPR	General Data Protection Regulation
IFC	Industry Foundation Classes
IOU	Intersection Over Union
KPI	Key Performance Indicator
mAP	Mean Average Precision
MEP	Mechanical, Electrical and Plumbing
OIP	Open Infra Platform
PG	Photogrammetry
PWMA	Process & Workflow Modelling and Automation toolkit
R&D	Research & Development
R-CNN	Regional Convolutional Neural Network
RenoDSS	BIMERR Renovation Decision Support System
SfM	Structure from Motion
SPF	STEP Physical file
TLS	Terrestrial Laser Scanning
TUM	Technical University of Munich
UC	Use Case
UEDIN	The University of Edinburgh
UK	United Kingdom
UR	User Requirement
WP	Work Package

EXECUTIVE SUMMARY

The BIMERR Deliverable D5.3 “Innovative Scan-to-BIM Tools for Automated BIM Generation v1” aims at documenting the BIMERR Scan-to-BIM tool and concluding the first iteration of the development activities in T5.2 “Enhanced Scan-to-BIM Tools for Topology Reconstruction”. Overall, the Scan-to-BIM tool aims to generate a BIM model of a residential building to be renovated from survey data (laser scanning and photographic data). The BIM model generated by the Scan-to-BIM Tool is then used by the Renovation Decision Support System (RenoDSS) for conducting some energy analysis and selecting the most suitable renovation strategy. As such, the Scan-to-BIM Tool lies at the beginning of the overall BIMERR renovation process.

The Scan-to-BIM Tool is designed to be a single desktop software package made up of several sub-components, all built with free and open-source technology. First of all, the Scan-to-BIM Tool is built on a Scan+BIM Software Framework that enables the loading, visualisation and authoring of both Building Information Models and reality capture data (i.e. point clouds and imagery). Such software framework did not exist at the start of the BIMERR project and thus had to be built from existing open software libraries and solutions. Then, the Scan-to-BIM Tool is composed of two main sub-components: the *Scan-to-BIM (Structural)* sub-component aimed at automatically generating BIM models of the structure (i.e. walls, slabs, openings) of the buildings from point clouds; and the *Scan-to-BIM (MEP)* sub-component aimed at automatically augmenting the BIM models with objects such as radiators, electrical sockets and switches, etc. from photographic data.

The present documentation of the BIMERR Enhanced Scan-to-BIM Tool along with its subcomponents is oriented towards the functionalities they broadly deliver, the technology stacks they build upon, the APIs they expose, the installation instructions and end-to-end usage walkthroughs they offer to their users. In its initial release, the BIMERR Enhanced Scan-to-BIM Tool does not implement all the envisaged functionalities and is not presented as a single, unified tool. Instead, all functionalities developed to date are presented separately. It is estimated that the Enhanced Scan-to-BIM Tool, in its initial version presented here, is about 60% complete.

The final iteration of the BIMERR Enhanced Scan-to-BIM tool is anticipated to be released on M30 of the BIMERR project implementation. That final iteration will include all functionalities integrated in a single, unified software package, with results obtained using data from the pre-validation sites.

1. INTRODUCTION

1.1 SCOPE AND OBJECTIVES OF THE DELIVERABLE

This deliverable reports on the work conducted up to M20 on the Scan-to-BIM Tool being developed as part of T5.2. Scan-to-BIM can be seen as the initial step in the BIM-enabled renovation process proposed by the BIMERR consortium. The requirement for the Scan-to-BIM tool is that it enables building surveyors / BIM modellers to create, from reality capture data (i.e. point clouds and photographs), BIM models of existing and inhabited domestic buildings in a state as complete as possible to support energy analysis as well as refurbishment planning (see Section 1.2).

This deliverable more specifically reports on three areas of development of the Scan-to-BIM Tool:

- **Scan+BIM platform:** the platform developed to uniquely enable the visualisation and manipulation of: *scan* data (3D point clouds, visible light pictures, and potentially infrared pictures), calling the Scan-to-BIM tools presented below; and *material/component* information, communicating with the BIMERR Interoperability Framework (BIF).
- **Scan-to-BIM (Structural) tool:** the data processing set of algorithms that take as input scan data (principally point cloud) and output a BIM model containing the *structural* components of the building (i.e. walls, floors, ceiling, openings) as well as the topological relationships (i.e. spaces, boundary representations). The tools employ both automated and manual processes. The latter are principally used to populate the material-related properties of all structural entities by using information from the BIMERR Material database.
- **Scan-to-BIM (MEP) tool:** the data processing tool that takes as input the scan data (principally visible light pictures) and outputs a Mechanical, Electrical and Plumbing (MEP) components (e.g. radiators, switches, lights) that are integrated the structural model to produce an enriched BIM model. The tools employs automated processes only.

1.2 RELATION TO OTHER TASKS/DELIVERABLES

This deliverable reports the first version (v1) of the Scan-to-BIM Tool, with a second and final version (v2) expected to be delivered at M30 and documented in deliverable D5.4.

Relation to other BIMERR Tools

BIM models outputted by the Scan-to-BIM tool will be principally used by:

- the **ARIBFA** tool to complete/correct the BIM model on site (WP5; Deliverables D5.9 and D5.10), and
- the **RenoDSS** tool for assessing the energy performance of an existing building and designing its energy-focused refurbishment (WP7; Deliverables D7.9 and D7.10).

Other BIMERR Tools (e.g. BICA, PWMA) will also use BIM model information, initially coming from the Scan-to-BIM Tool, but after having been revised and processed by the ARIBFA and RenoDSS tools.

As mentioned before, the input data for the Scan-to-BIM tool includes *scan* data (3D point clouds, visible light pictures, and potentially infrared pictures) and *material/component* information. While the first dataset is delivered by building surveyors, the latter is provided by the *material/component* database (WP7; Deliverables D7.1 and D7.2).

Input and output data of the Scan-to-BIM tool will be stored in the BIMERR Data Stores and pulled/pushed using the *BIMERR Interoperability Framework (BIF)* (WP4; Deliverables D4.8 and D4.9).

Relations to Other Deliverables

The Scan-to-BIM tool is principally used to support the Use Case 1 (UC1) and meet the corresponding User Requirements that are all described in deliverable D3.1. The functional requirements, input and output requirements, and connections with other BIMERR components or tools (as presented in the previous sub-section) were all defined earlier in this project and reported in deliverable D3.5.

The User Requirements (URs) identified in D3.1 and relevant to the Scan-to-BIM tools are listed in Table 1. In Table 1, the right column indicates the level to which each of those URs are currently met.

Table 1: User Requirements relevant to the Scan-to-BIM tool and the level to which they shall be met.

ID	User Group	Description	Priority	Met	% completed
3	Architect	Architect shall be able to import BIM models to their design software in IFC format	High	Yes	100%
11	Architect	Architects shall be able to get material specifications (coefficients and properties) in order to complete their work	High	Yes	90%
12	Architect	Architect shall be able to have an up-to-date representation of the building to be renovated	High	Yes	60%
13	Architect	Architect shall be able to use current survey results for construction representation	High	Yes	100%
14	Architect	Architect should be able to use BIM model in order to check geometrical properties of the building.	Low	Yes	100%
35	BIM Expert	BIM experts shall be able to get point clouds in order to generate BIM models	High	Yes	100%
40	BIM Expert	Point clouds shall be stored in basic formats	High	Yes (ply)	100%
41	BIM Expert	Point clouds shall be stored in enriched formats	High	Yes (E57)	100%
43	BIM Expert	BIM experts shall be able to produce BIM models in IFC format	High	Yes	100%
87	Surveyor	The level of accuracy for the design models shall be less than 10 mm for building constructions	High / Medium	TBC	N/A
148	Surveyor	Surveyors shall be able to use open formats for delivering as-is data.	High	Yes	100%

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

1.3 Structure of the document

In order to address all the aspects relevant to the scope of D5.3, the present deliverable has been structured as follows:

- Section 1 introduces the work performed and the scope of this deliverable along with its relevance to other BIMERR tasks and the deliverable's structure.
- Section 2 presents the strategy followed to deliver a Scan+BIM platform, where both reality capture and BIM models can be visualised and, potentially, modified.
- Sections 3 provides an overview of the Scan-to-BIM tool, which is divided into structural and MEP subcomponents.
- Section 4 delivers a comprehensive documentation of the algorithms developed for the *Scan-to-BIM (Structural) sub-component*, which are devoted to identify structural components and subsequently produce an IFC file.
- Section 5 offers a comprehensive documentation of the algorithms developed for the *Scan-to-BIM (MEP) sub-component*, that aim to identify MEP elements in images of the building and enrich the IFC file with those MEP objects.
- Section 6 presents the conclusions to the deliverable, along with the release plan for the second and final version of the BIMERR Scan-to-BIM tool.

2. SCAN+BIM PLATFORM

2.1 OVERVIEW

Although meaningful research has been recently conducted at the intersection of Building Information Modelling (BIM) and reality capture data processing, no open-source, libre, and cost-free tools have been developed for simultaneously handling point clouds and BIM models in an effective manner. In the last decade, relevant software companies, especially those developing graphic design tools, have delivered a number of solutions devoted to visualise and manipulate BIM models and/or point clouds. These commercial tools present, in occasions, limitations that cannot be easily addressed by users who may be interested in implementing their own functionalities and requirements. In contrast, open-source solutions are generally developed ad-hoc to deal with particular tasks, not being able to deal with a variety of data formats or not providing sufficient operations to modify reality capture or BIM data. Nonetheless, free software facilitates the addition of new functionalities to existing tools according to the users' needs and to the benefit of all. This is particularly valuable to research communities.

In order to compare existing solutions able to handle both BIM and point clouds (Scan+BIM), Cyberbuild Lab, at The University of Edinburgh, prepared a questionnaire to analyse the way R&D professionals from academia and industry work with BIM and reality capture data, and ask them about their preferences in terms of the functionalities they would like to see in Scan+BIM software (Valero et al., 2020). This study identified a number of advantages and limitations of existing tools and subsequently supported the articulation of a new open-source solution considering various relevant functionalities highlighted by the R&D professionals.

The following subsections present the proposed solution, which combines Open Infra Platform (Hetch and Jaud, 2019) , as a scalable engine for the visualisation of BIM and point cloud data, and the xBIM toolkit (Lockley et al. 2017) that provides supplementary libraries for the manipulation (i.e. generation and alteration) of BIM models in IFC format. The Scan-to-BIM (Structural) and Scan-to-BIM (MEP) tools reported in Sections 4 and 5 shall be integrated as modules or plugins to the Scan+BIM platform.

2.2 TECHNOLOGY STACK AND IMPLEMENTATION TOOLS

The Open Infra Platform (OIP) is an open-source software solution developed by the Technical University of Munich (TUM) that enables importing and visualising both point clouds and BIM models (in IFC format). Figure 1 shows the interface of OIP.

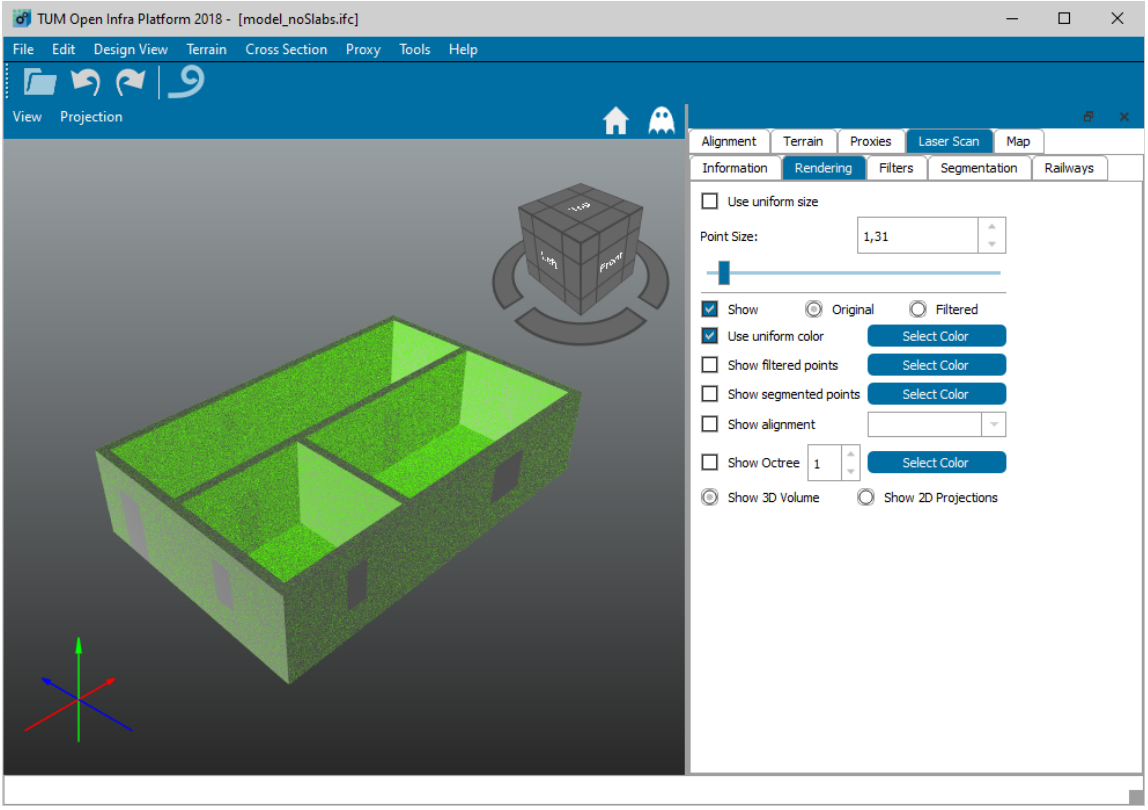


Figure 1: Point cloud and BIM model visualised into OIP. Note that the interface shown here has been customised with the BIMERR colour palette.

Reality Capture data: OIP integrates the CloudCompare (CC) engine (CloudCompare, 2020) which facilitates the visualisation and processing of point clouds. A number of formats can be read by OIP/CC, both proprietary and open format, codified (i.e. binary) or human-readable (i.e ASCII). Particularly, the open-source standard E57 format (Huber, 2011) is supported, as required by the BIMERR specifications detailed in Deliverable D3.1 (Stakeholder requirements for the BIMERR system).

BIM data: IFC is a data model used to describe architectural, building and construction industry data. It is an open, international standard proposed by buildingSMART (ISO, 2018) that is continuously developed and already widely promoted and used in the industry. The use of IFC data models and corresponding file formats are an important BIMERR User Requirement (see Deliverable D3.1). The OIP platform supports the loading and visualisation of IFC data in “Step-file” forma. OIP can open various versions of the schema, in particular IFC version 4x1 which was agreed to be used by the BIMERR solutions. Note that support for 2nd level space boundary (2LSB) representations, which is important for energy-related data analysis, was only introduced in IFC 4.

OIP is mostly written in C++ language and its source code is hosted in a Github repository (https://github.com/tumcms/Open-Infra-Platform). Libraries used in the project are enumerated in Table 2.

Table 2: Tools and libraries used in the Open Infra Platform

Library	Version	Licence
Qt	5.12.2	LGPL version 3
boost	1.65.1	Boost Software License
Eigen	3.3.7	MPL2
IfcPlusPlus	-	MIT License
BlueFramework	a0.0.2	GPL v3 License
CloudCompare	2.10.3	GPL v2 License
libLAS	1.6	BSD license
qhull	-	Without modification
raptor2	-	GPL v2 License

2.3 API Documentation

Not relevant.

2.4 Assumptions and Restrictions

OIP has a number of assumptions and restrictions which are presented in the following.

- OIP has been uniquely developed to be run in Windows OS.
- Both reality capture and BIM datasets can be simultaneously visualised but the platform is currently designed to load one point cloud file and one BIM (IFC) file. Also, the point cloud currently must be loaded first.
- Openings in constructive elements cannot be properly visualised in the current version of the software.
- Navigation functionalities remain a little limited (rotation is only achieved through a navigation cube).
- Object selection is currently not available.

The last three restriction will be resolved for the next version of Scan-to-BIM Tool (v2).

2.5 Installation Instructions

Regular users uniquely have to download a compressed folder (i.e. OIP_BIMERR.zip file in the release "v1.0"), from the CyberBuild-forked OIP repository (https://github.com/CyberbuildLab/Open-Infra-Platform), which contains the files required for proper execution of the software, and run the "OpenInfraPlatform.UI.exe" file.

Advanced users or developers may be interested in contributing to the development of the tool, include their own functionalities or modify certain operations. In this case, a more complex process, including the configuration, the generation and the compilation of the OIP solution is required (see https://github.com/CyberbuildLab/Open-Infra-Platform/blob/development/Installation.md). Then, OIP requires the installation of third party libraries and toolkits. Almost all these tools are automatically installed during the configuration of the project with CMake. More detailed information about these libraries, such as tested versions and licences is provided in the CyberBuild-forked OIP repository.

2.6 USAGE WALKTHROUGH

Inside the "release" folder mentioned in Section 2.5, double-click on *OpenInfraPlatform.UI.exe*, highlighted in Figure 2(a), to run OIP. Inside the OIP software, click on the folder icon, as shown in Figure 2(b), to load files into the software. First, load a point cloud (e.g. cloud_coloured.e57); then load an IFC model (e.g. model.ifc) following the order illustrated in Figure 2(c). After loading the point cloud, its colour and the size of the points can be modified by using the tools labelled "1" and "2" in Figure 2(d). Navigation (i.e. rotation) can be done by using the navigation cube on the top-right of the viewer (see Figure 2 (f)).

Exemplar Dataset

Examples of a point cloud and an IFC model created after the 3D dataset can be found at https://datashare.is.ed.ac.uk/handle/10283/3718.

2.7 LICENSING

TUM Open Infra Platform is free software; you can redistribute it and/or modify it under the terms of the **GNU General Public License Version 3** as published by the Free Software Foundation (https://www.gnu.org/licenses/gpl-3.0.en.html).

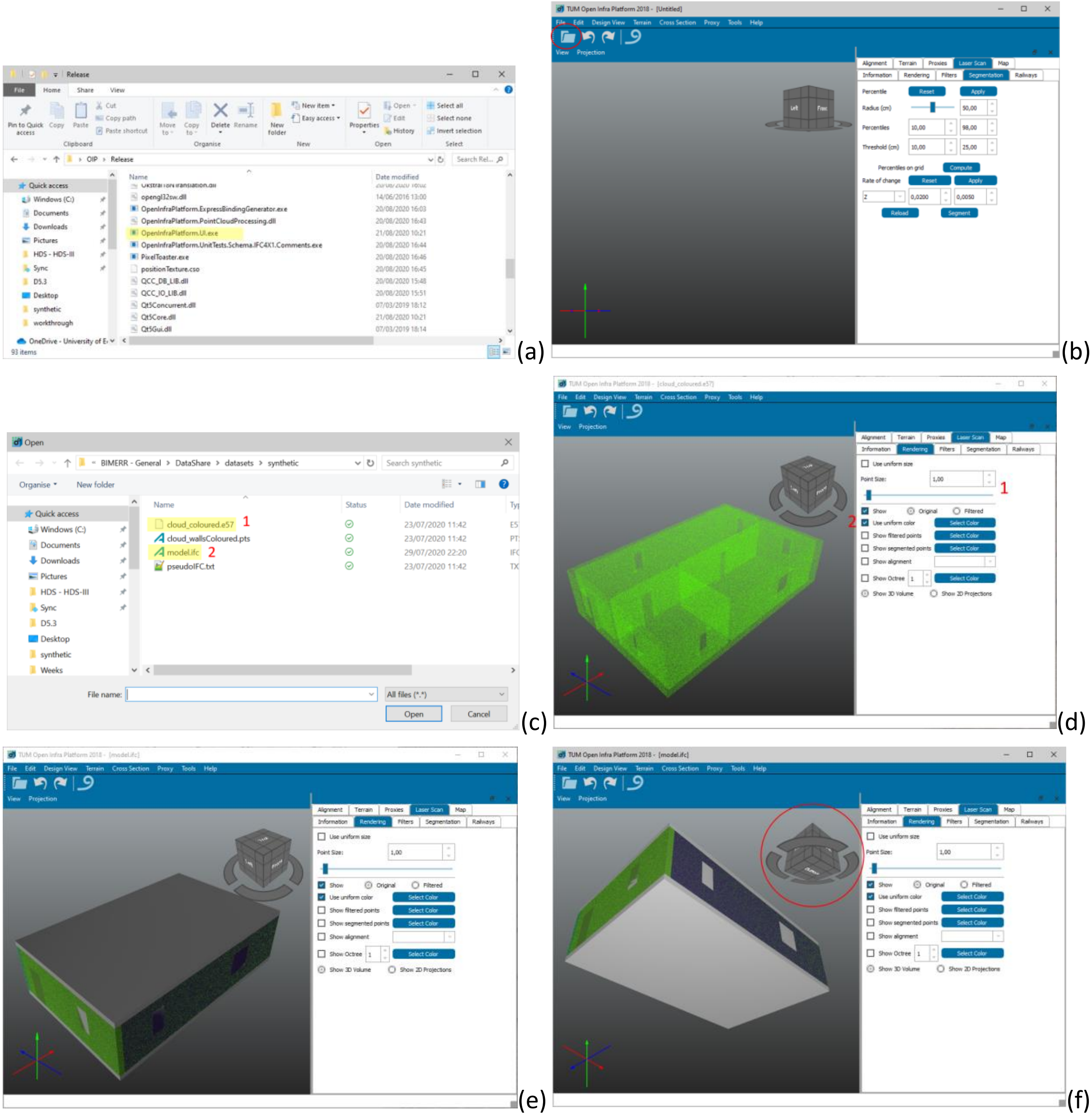


Figure 2: OIP walkthrough. (a) "Release" folder of the BIMERR OIP solution; (b) Main window of the OIP. (c) "Open" menu dialog, where point clouds and ifc files can be selected; (d) Point cloud loaded in OIP, coloured in green with point size 1; (e) Point cloud and ifc model simultaneously loaded in OIP; (f) Bottom view of model and point cloud.

3. Scan-to-BIM Overview

The aim of the proposed Scan-to-BIM Solution is to (semi-)automatically generate BIM models (in IFC formats) from reality capture data, including unstructured point clouds obtained by Terrestrial Laser Scanning (TLS) devices and/or photogrammetric (PG) techniques, as well as external data sources (e.g. material databases).

Scan-to-BIM is a complex process that can be divided into a variety of subtasks performed by different tools employing a range of computer vision algorithms. We organise the Scan-to-BIM process into two main stages and corresponding sub-components:

- **Scan-to-BIM (Structural) sub-component:** this sub-component takes as input a dense point cloud (typically from laser scanning). It then detects 'structural' elements (i.e. floors, ceilings, walls, doors and windows) in that data. The sub-component then outputs that `structural' model in an IFC format.
- **Scan-to-BIM (MEP) sub-component:** this sub-component takes as input the `structural' IFC model output from the first sub-component as well as a photogrammetry dataset (i.e. images and structure-from-motion point cloud) of the same environment. It then detects MEP components in the images and add (i.e. model) those detected components to the IFC model.

The above sub-components are presented in Sections 4 and 5, respectively.

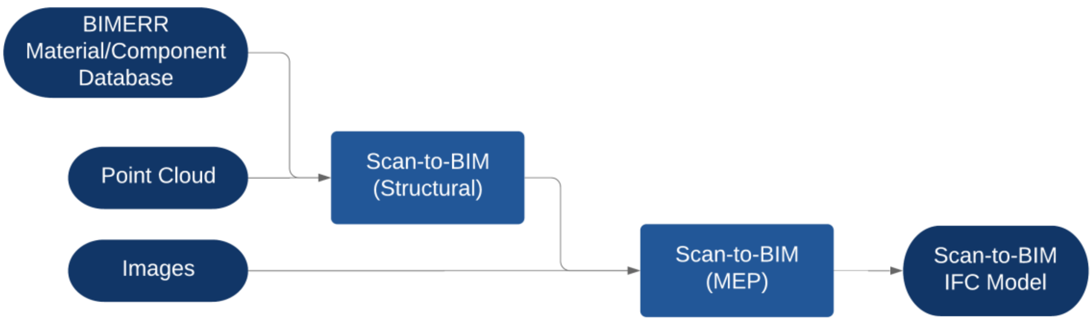


Figure 3: Overview of the overall Scan-to-BIM process

4. SCAN-TO-BIM (STRUCTURAL) SUB-COMPONENT

4.1 OVERVIEW

The objective of the Scan-to-BIM (Structural) sub-component is the identification of the main structural components of buildings, their parametrisation for the generation of BIM models and the subsequent codification of the delivered information into IFC-SPF files. As illustrated in Figure 4, the tool is divided into a number of tasks (or algorithms), which analyse the building from general to specific.

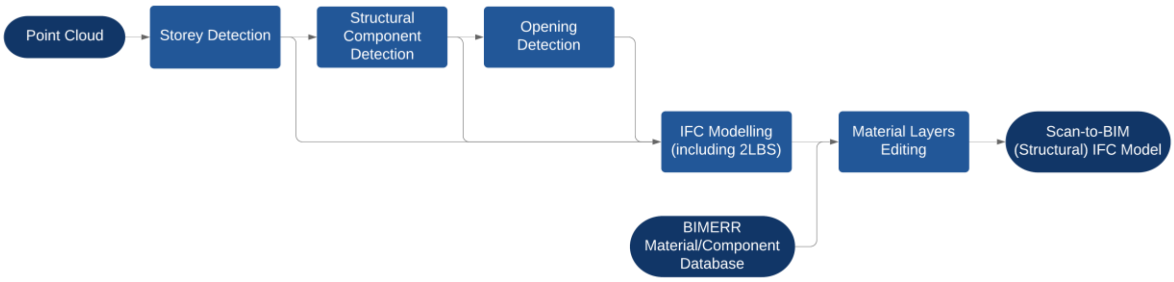
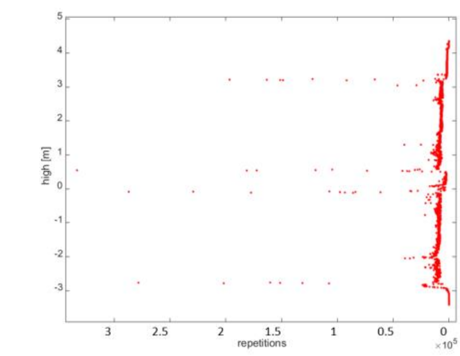


Figure 4: Overview of the Scan-to-BIM (Structural) process

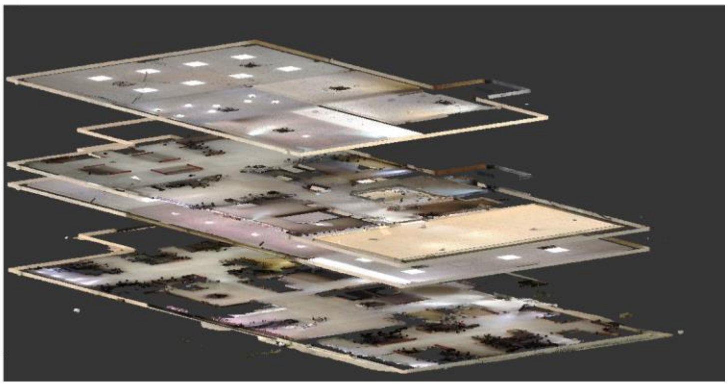
After scanning works and pre-processing of individual point clouds, all the unstructured data are aligned under the same universal coordinate system and stored in an E57 file. The resulting point cloud (see example in Figure 5) is the input to the Scan-to-BIM (Structural) sub-component process. The process goes through the successive steps:

- **Storey identification:** the first stage of the Scan-to-BIM process consists in the segmentation (i.e. labelling) of the points into storeys. The proposed approach for this operation is based on the analysis of point density along the vertical axis. A histogram is generated and ceilings and floors are extracted. Then, these are used for defining the storeys, and points with a z-value between z_i^{floor} and $z_i^{ceiling}$ are labelled as part of storey *i*. This process is illustrated in Figure 6.

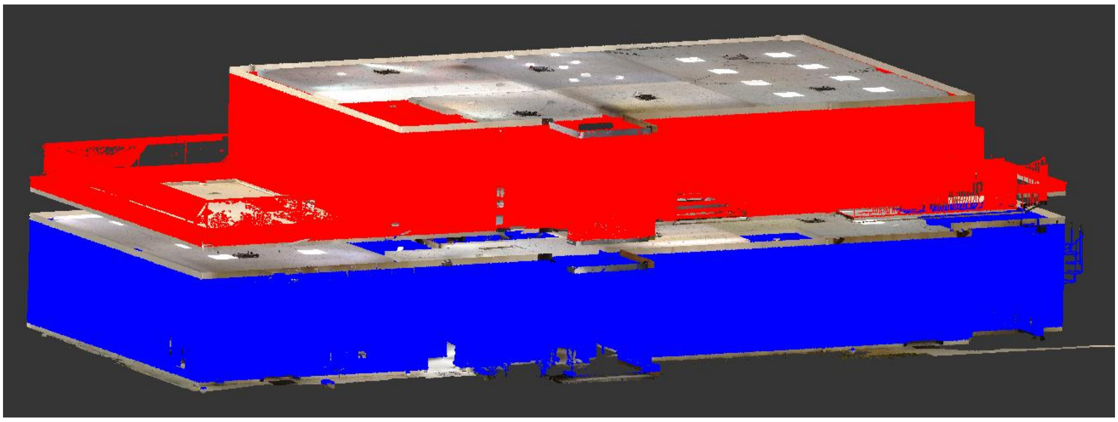
Figure 5: Coloured point cloud of a two-storey building. Kripis house



(a)



(b)

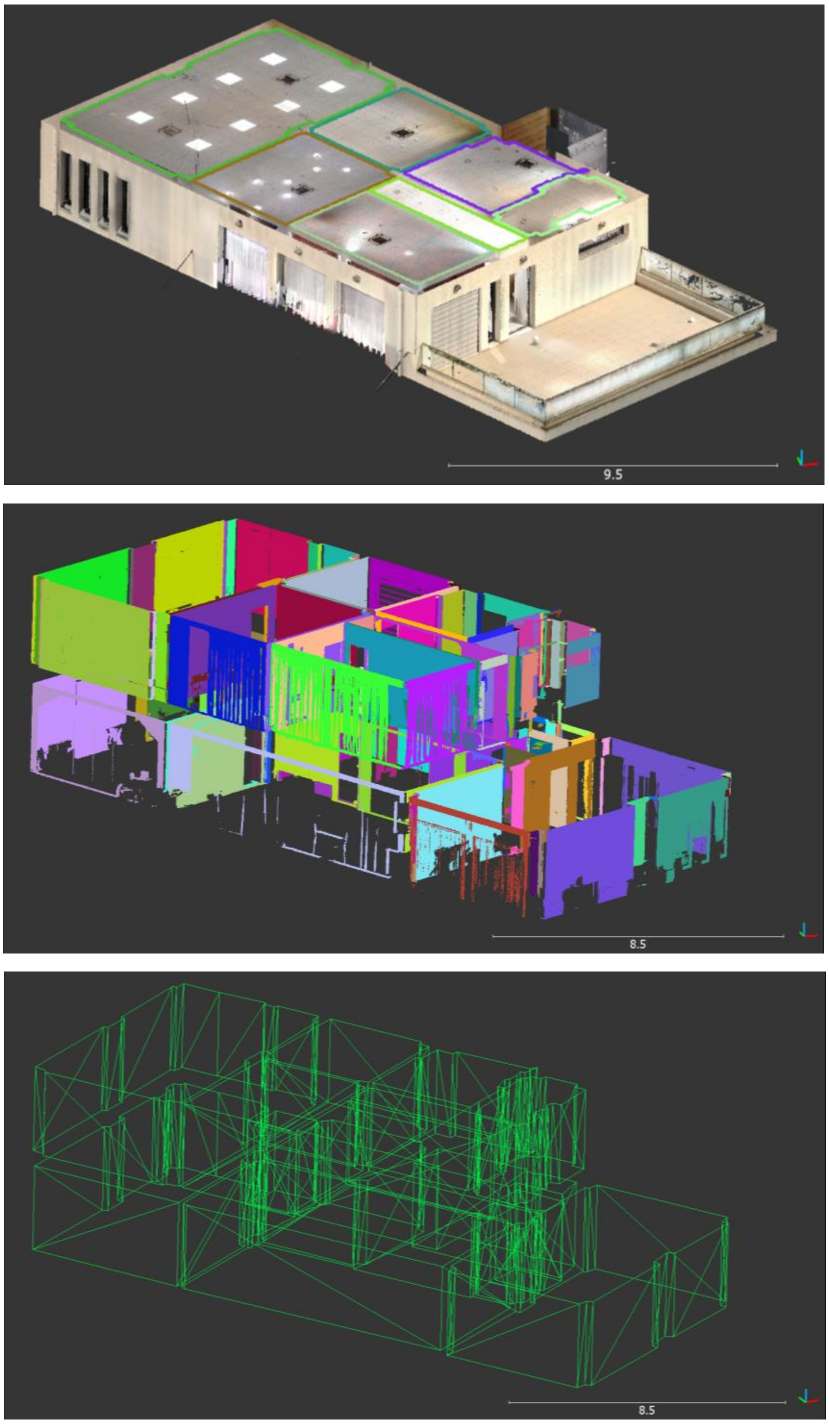


(c)

Figure 6: Storey Detection: (a) Histogram showing density of points along the vertical axis; (b) segmented floors and ceilings of a building (right); and (c) overall point cloud segmented into storeys.

- **Structural Component Detection:** Once the global point cloud is segmented into storeys, each individual cloud representing a storey of the building is processed to extract the surfaces of the walls visible to the scanner.

The points corresponding to the ceiling of each storey are first extracted and subsequently classified into clusters. Note that, the lack of points inside the wall elements facilitates the identification of clusters, and therefore, storeys. Results for the labelling of the spaces in a building storey are illustrated in (a), where the boundary of each space is plotted in a different colour.



(a)

(b)

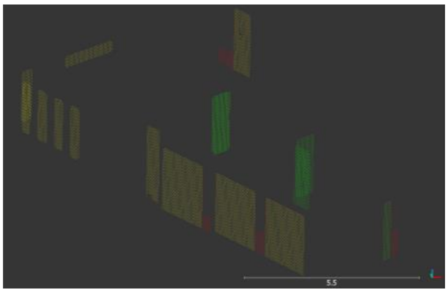
(c)

Figure 7: Structural Component Detection: (a) storey spaces labelled in different colours; (b) Internal walls detections from the ceiling boundaries; and (c) 1LSB representation for identified spaces.

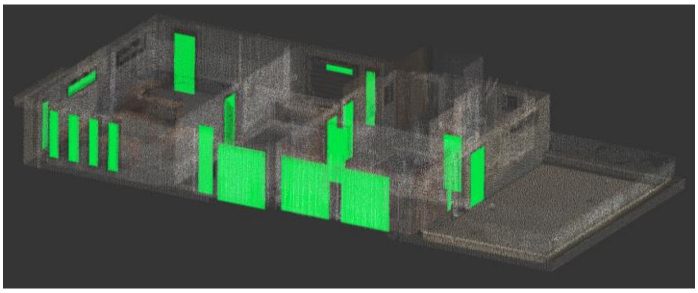
Assuming that the vast majority of walls are vertical, the boundary of the ceiling of each space is divided into consecutive straight segments where points of the visible surfaces of walls are projected. Therefore, the point cloud corresponding to each storey is segmented into walls (i.e. surfaces), as seen in (b) and, subsequently, faces of walls are extracted, defining the Boundary Representation (i.e. 1LSB) for each space, as shown in (c).

Finally, nearest neighbours are searched for each wall (or slab) surface, amongst those with parallel (but opposite) normal vectors, and subsets are assigned to the same wall object. If there are no parallel surfaces, then, an artificial surface is created to define the wall. The outcome of this process is the set of 3D walls for the given storey. *Note*: this step has not been implemented yet.

- **Opening Detection:** After all the spaces and walls are defined (i.e. the 1st level space boundaries (1LSB)), opening elements (e.g. doors and windows) are identified for walls and slabs by processing point clouds representing both sides of each wall. Bounding boxes are calculated for all empty areas and openings are detected based on the overlap between pairs of bounding boxes on both sides of the walls, as illustrated in Figure 8 (right).



(a)



(b)

Figure 8: Opening Detection: (a) Scorings for potential openings; and (b) rectangular patches represented identified openings.

- **IFC Modelling (including 2LSB):** Once the structural entities are parametrised, this information is codified in an IFC (STEP) file. Basic relationships between certain entities (e.g. a wall and its openings) are established, and importantly, more complex connections between surfaces and spaces are produced to define 2nd Level Space Boundaries (2LSB), as can be seen in Figure 9, after loading the delivered model into Solibri Anywhere (Solibri Inc. 2020), to check the relations between space boundaries, functionality not available in most IFC viewers.

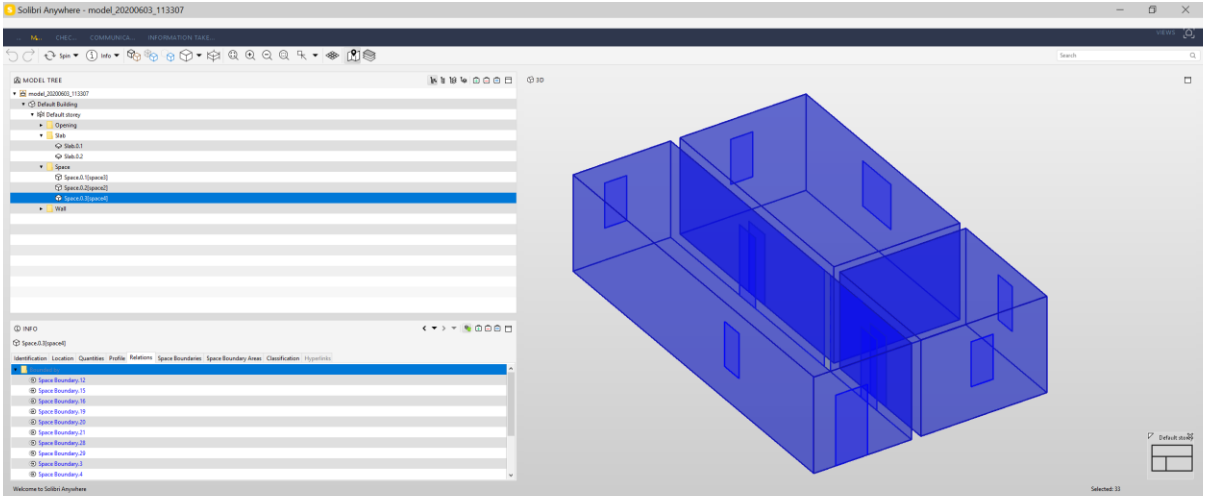


Figure 9: IFC model, including 2LSB-related information, outputted by the Scan-to-BIM (Structural) sub-component and loaded on Solibri Anywhere (Solibri Inc., 2020).

- **Material Layers Definition/Editing:** this is the only manual step of the process. This has to be the case because there is no technology that can be employed and integrated within the survey process that would enable the automated population of the material information for walls, slabs, windows, etc. Furthermore, this material information must match that of the materials stored in the BIMERR Material Database (see Deliverable D7.1) to enable a smooth integration with RenoDSS. As a result, a manual interface has been developed, in which users can select structural components and edit their material information (including entities with multiple layers), using information from the BIMERR Material Database. Figure 9 shows how materials can be added and/or edited. After selecting a particular material and specifying the thickness of the related layer, the set of layers can be organised in the correct order within the boundaries of the wall. When the editing is confirmed, the IFC file is updated to include all the modifications.

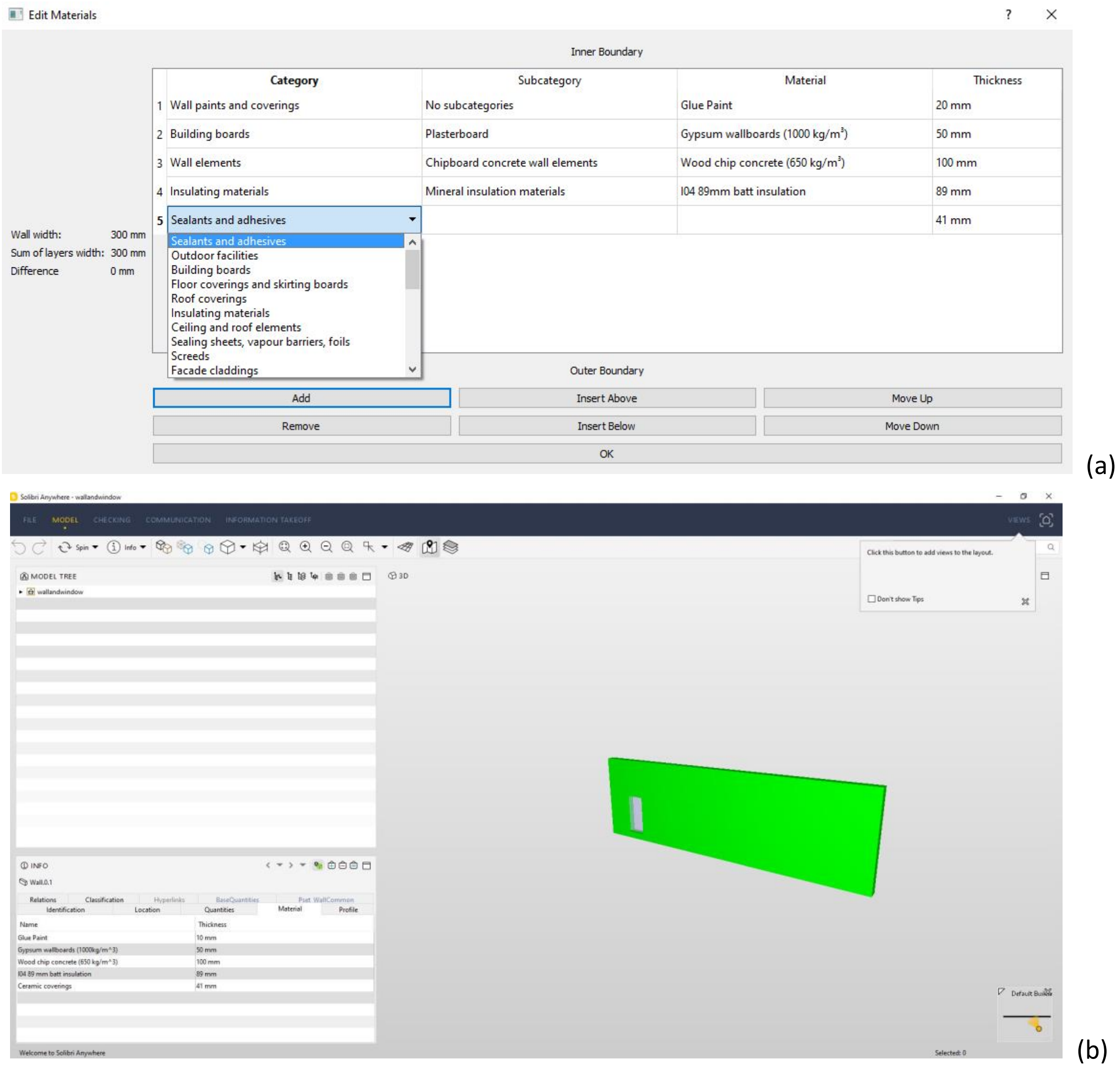


Figure 10: Material Layers Definition/Editing: (a) Interface enabling the user to add and edit materials of an IFC wall. The materials are selected from the BIMERR Material Database; and (b) outputted IFC model, including internal layers information, loaded in Solibri software package.

4.2 Technology Stack and Implementation Tools

The Scan-to-BIM sub-component focused on the analysis of structural elements can be divided into three subprojects:

- Identification and parametrisation of structural components, and further analysis of the relationships amongst these.
- Codification of the obtained information into IFC (STEP) files, following the IFC 4x1 version.
- Definition and edition of material layers for structural entities, acquiring information from the BIMERR material database.

These tools will be run as standalone projects, invoking them from the command prompt, or as a plugin inside OIP. The first one is written in C++ and delivers a pseudo-IFC file, which contains information about the geometry of structural elements and the existing relationships between them. The second tool, written in C#, reads the output from the first project and produces IFC files. Finally, the third tool has been developed using C++ and C# and edits IFC files by adding material properties to the selected structural components. Libraries used in the above-mentioned projects are enumerated in Table 3.

Table 3: Tools and libraries used in the structural subcomponent of Scan-to-BIM.

Library	Version	Licence
Qt	5.12.2	LGPL version 3
Eigen	3.3.7	MPL2
OpenCV	4.1.2	Apache 2
PCL	1.9.1	BSD Licence
Xbim.Essentials	5.1.297	CDDL

4.3 API/Software Documentation

All the algorithms will be merged under a unique project and included as a single tool in the Scan+BIM Platform. However, individual processes can be also run from the command prompt if the user requires only one part of the structural Scan-to-BIM to be executed.

- **extractStoreys:** this algorithm takes a point cloud of a building as a main input, divides it into storeys, extract structural elements and parametrises their geometry and relationships. Input arguments are the following**:** (1) path to file (i.e. point cloud), (2) size of voxels to divide the dataset and identify the spaces, (3) resolution of the point cloud. Binaries can be found at https://datashare.is.ed.ac.uk/handle/10283/3738.
- **calculateOpenings:** this program reads pairs of point clouds corresponding to both sides of every wall, identifies where holes are on the surfaces and calculate openings, by analysing the overlapping between patches extracted from the two sides. Input arguments are: (1) path to the folder where the two point clouds of the walls are, (2) size of voxels to divide the cloud and identify the openings. Binaries can be found at https://datashare.is.ed.ac.uk/handle/10283/3736.
- **IFCGenerator2LSB:** this program reads a pseudo-IFC file (automatically produced after *extractStoreys* and *calculateOpenings*), containing semantic information about the structural entities defining a building and produces an IFC (STEP) file. The only input argument is the path to this file. Binaries can be found at https://datashare.is.ed.ac.uk/handle/10283/3737.
- **EditMaterials:** this program reads IFC files and enable the user to manual modify material information of selected structural components using materials from the BIMERR Material

Database. The single input argument is a path to the IFC file. Binaries can be found at https://datashare.is.ed.ac.uk/handle/10283/3735.

4.4 Assumptions and Restrictions

The structural component of the Scan-to-BIM tool has a number of assumptions and restrictions which are presented in the following:

- It has been uniquely developed to be run in Windows 10 OS.
- *extractStoreys* and *calculateOpenings* currently support .ply format. E57 format is already supported by the Scan+BIM platform (i.e. OIP). The Scan-to-BIM (Structural) sub-component will be able to work with E57 input data once it is integrated in OIP.
- Point clouds are considered to be geo-referenced, or at least oriented, with storeys along the Z axis.
- Both floors and ceilings (i.e. slabs) are considered to be horizontal and most walls, vertical.
- The tool is not yet entirely integrated in a single executable, but as a set executables for its different sub-processes.

4.5 Installation Instructions

No complex installation is required. Binaries for individual processes are provided in the Cyberbuild Datashare collections.

The Scan-to-BIM (Structural) tool, when delivered integrated to the Scan+BIM Platform, will be configured during the compilation of the overall solution.

4.6 Usage Walkthrough

The standalone projects are invoked from the command prompt. For this, users only need to navigate to the folder where the binaries are (e.g. cd “C:/Path/to/your/bin/”) and execute the program.

For executing the program, user must type the name of the .exe file of the tool followed by the input arguments. Examples for the tools presented in Section 4.3 are given in the following.

- For extractStoreys:

  >> extractStoreys.exe "C:/Path/to/your/file/pointCloud.ply" 0.04 1

- For calculateOpenings:

  >> calculateOpenings.exe "C:/Path/to/the/folder/with/WallsPairsFolder" 0.05

- For IFCGenerator2LSB:

  >> ifcGenerator2LSB.exe "C:/Path/to/your/file/pseudoIFC.txt"

- For editMaterials:

  First, double-click on editMaterials.exe, highlighted in Figure 11(a). Once the main dialog window is opened, press the button "…" (see Figure 11(b)) and select an IFC (STEP) file in the file explorer window. Then, click on "Load Components" and the list of components will appear in the table underneath, as illustrated in Figure 11(c). Select the structural component from the list and click on the button "Edit material layers". A new dialog window will appear (Figure 11 d).

  For each layer (i.e. each row), use first the drop-down menu in the column "category" and select one of the categories. Do the same for the "subcategory" column and, finally, the "material" column. Input the "thickness" of that layer, as shown in Figure 11(d).

  If needed, add, remove or rearrange the order of layers using buttons at the bottom of the window.

  Finally, confirm changes by pressing "OK" button.

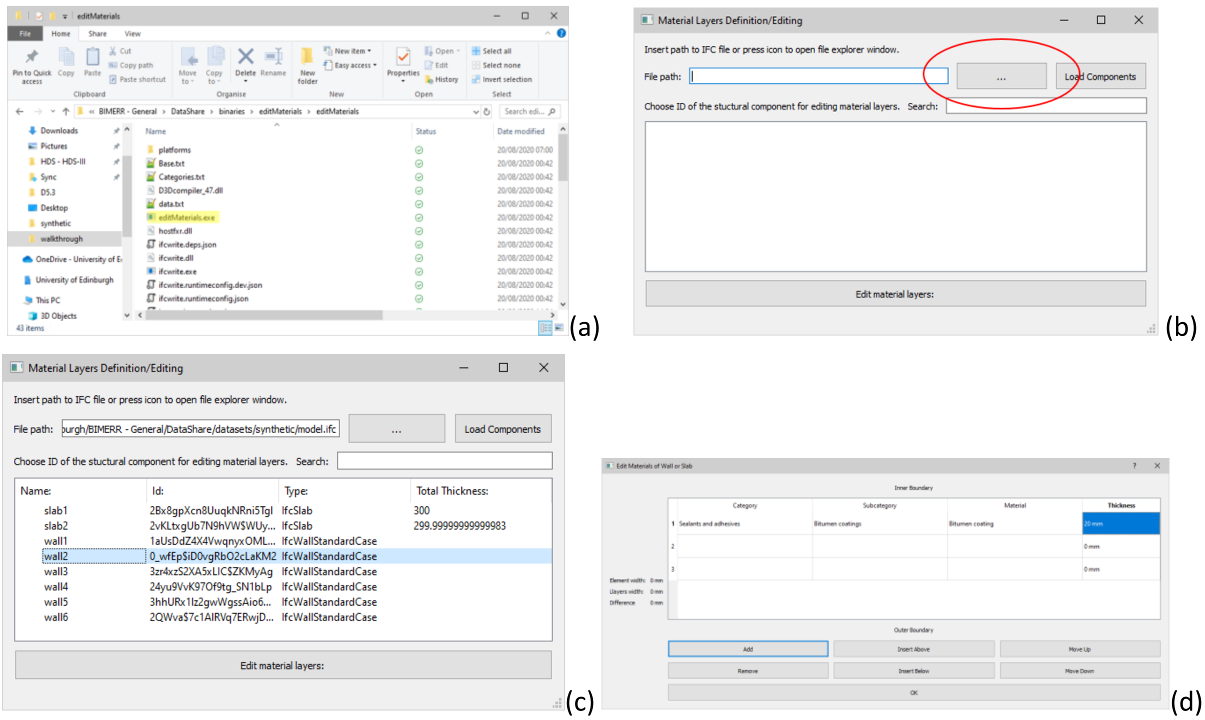


Figure 11: Walkthrough for adding material information of ifcSlabs and ifcWalls. (a) "Release" folder of the editMaterial tool; (b) Main window of editMaterial; (c) Wall and slab entities of an ifc model; (d) Dialog for editing the entities' layers.

In the final version, the Scan-to-BIM (Structural) sub-component will be embedded into the Scan+BIM Platform. For running the tool, the user will only need to load a point cloud, click on the "Scan-to-BIM (Structural)" icon and a dialog window will appear to ask for the required input arguments.

Exemplar Dataset

First, a synthetic point cloud of a simple 1-storey building was created (see https://datashare.is.ed.ac.uk/handle/10283/3718) for investigating and designing a scan-to-BIM strategy and further developing the scan-to-BIM tool. From this point cloud, a pseudo-IFC was produced (pseudoIFC.txt), containing semantic information about the building entities, in order to create an IFC file with 2LSB information. This text file can be used as input to IFCGenerator2LSB (https://datashare.is.ed.ac.uk/handle/10283/3737) as previously detailed.

A point cloud of one of the pre-validation sites, Kripis House, has been acquired and pre-processed by CERTH. It has been subsequently cleaned (i.e. points not corresponding to the building removed) and subsampled to improve the drafted version of structural sub-component of the scan-to-BIM tool. This is an example of a real inhabited environment, with complex structural entities in a two-storey building. This point cloud can be used as input to *extractStoreys* algorithm (https://datashare.is.ed.ac.uk/handle/10283/3738). Additionally, point clouds corresponding to wall entities (two clouds per wall) have been extracted (wallPairsKripis.zip) to identify the openings by means of *calculateOpenings* (https://datashare.is.ed.ac.uk/handle/10283/3736).

4.7 LICENSING

The Scan-to-BIM tool is free software; you can redistribute it and/or modify it under the terms of the **GNU General Public License Version 3** as published by the Free Software Foundation (https://www.gnu.org/licenses/gpl-3.0.en.html).

5. Scan-to-BIM (MEP) Sub-component

5.1 Overview

The objective of Scan-to-BIM (MEP) sub-component is the identification and localisation of the MEP components of buildings in images of the environment, and subsequently adding these entities to the IFC-SPF files obtained by the Scan-to-BIM (Structural) component. The process includes several sub-processes as shown in Figure 12.

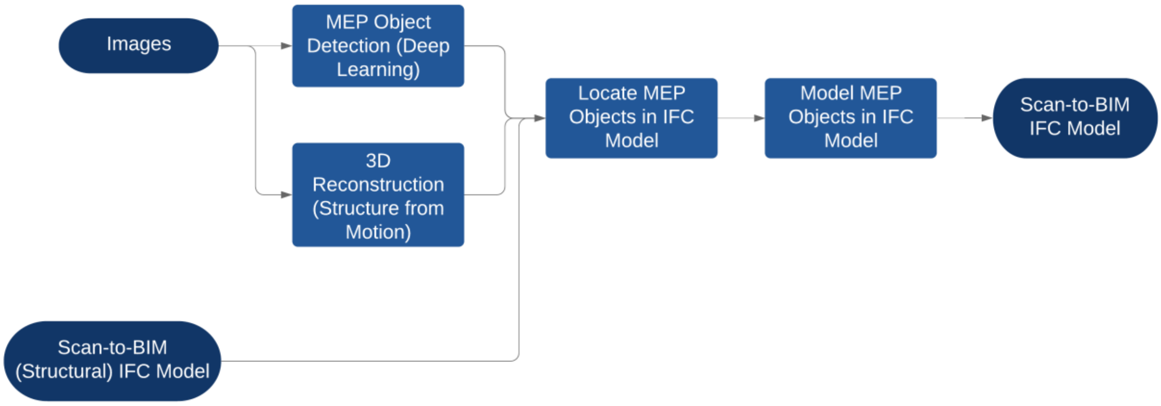


Figure 12: Overview of the Scan-to-BIM (MEP) process

The process encompasses the following steps:

- **MEP Object Detection:** The first step of the Scan-to-BIM (MEP) process is the identification and localisation of different MEP components (e.g. radiators, sockets, and switches) in the input images. The algorithm used is based on deep learning and it is structured as follows. First, a Faster Regional Convolutional Neural Network (Faster R-CNN) (Ren et al., 2015) is trained using a dataset of MEP components. The Faster R-CNN uses Neural Architecture Search Net (NASNet) (Zoph et al., 2018) as a feature extractor and has been previously trained on the Microsoft Common Objects in Context (MS-COCO) (Lin et al., 2014) detection dataset. Then, the trained model is tuned through a cycle of validation and optimization to achieve the best possible performance on the dataset. For the radiators, a mean average precision (mAP) of 1 at 75%IOU (Intersection of Union) has been achieved. For Sockets and switches the mAP at 50%IOU is 0.86 and 0.89, respectively. The detailed performance of the trained network models is summarised in Table 4. Figure 13 shows some successful and failed detection results for a number of images.
- **3D Reconstruction (Structure-from-Motion):** Structure-from-Motion (SfM) is a process that relies on photogrammetric measurements to generate a 3D model of an object or environment (UNC LibGuides, n.d.). SfM reconstructs the 3D environment from an inputted set of

overlapping two-dimensional photographs taken from many locations and orientations of the scene. The SfM algorithm then detects discernible features (e.g. the corner of a door) within each photo and matches them across the images. The matches are inputted to a *bundle adjustment* algorithm that simultaneously positions the pictures relative to one another in 3D space, the optical characteristics of the camera(s) employed to acquire the pictures, and defines the 3D coordinates of the points corresponding to the features matched across the images. Figure 14 shows a result of 3D reconstruction of a room from a set of photogrammetric images.

Figure 13: Some MEP object detection results.

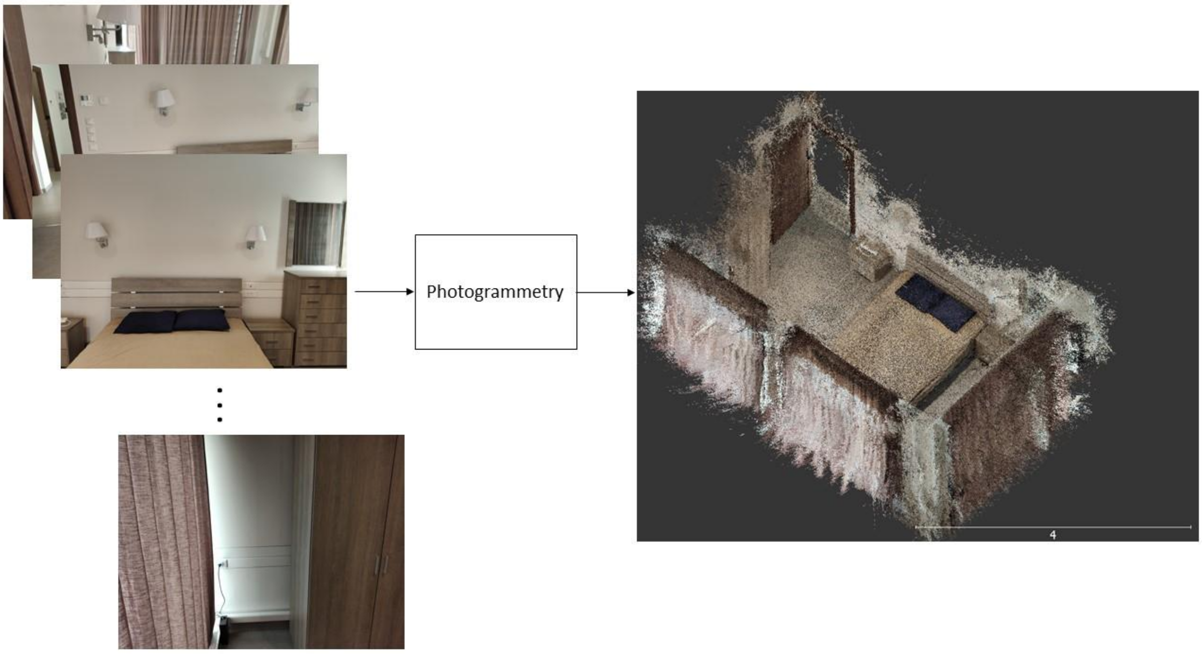


Figure 14: 3D Reconstruction of scene from images.

Table 4: MEP object detection performance.

MEP Object Category	No. of Images	mAP@50%	mAP@75%	mAP@50%:95%
Radiator	49	1.00	1.00	0.82
Socket	293	.864	.508	.479
Switch	293	.892	.590	.487

- **Locate MEP Objects in IFC Model:** This step takes the IFC model from Scan-to-BIM (Structural) component, the bounding box location from MEP detection and the photogrammetric 3D reconstruction of the scene as input. The photogrammetric reconstruction, once aligned with the Scan-to-BIM (Structural) model, provides the camera external calibration matrices and hence each pixel in image can be mapped to the corresponding point in the Scan-to-BIM (Structural) model (and the point cloud from which it was derived). *Note:* this step has not been implemented yet.
- **Model MEP Objects in IFC Model:** After mapping the bounding box to the Scan-to-BIM (Structural) model, the information is coded (i.e. added) into the IFC model file using the bounding box co-ordinates, the detected object category and information about the object on which the previously extracted information is mapped. *Note:* this step has not been implemented yet.

5.2 Technology Stack and Implementation Tools

The Scan-to-BIM (MEP) sub-component has four main sub-parts:

- Detection of MEP objects in image
- Photogrammetric 3D Reconstruction of the scene
- Mapping the detected bounding box to the point cloud
- Codification of the information to the IFC file

Of those sub-components, only the first two have been implemented and are considered here. The detection of MEP objects in images uses deep learning developed using Tensorflow (Abadi et al., 2016) and is written in Python. The model has been trained using the cloud computing service Google Colab, which offers cloud based Jupyter Notebooks with a NVIDIA Tesla K80 GPU. This component will be made available independently as a cloud-based web service. However, this will be embedded as a part of the complete solution. For photogrammetric 3D reconstruction of the scene from the images, a closed source but free software VisualSFM (Wu, 2011) software is being investigated. Currently the software being used for this process is Agisoft Metashape (Agisoft Metashape, 2020). The third component will be the same C# tool (based on the xBIM library) used in the Scan-to-BIM (Structural) component to generate IFC files. Table 5 summarises the tools and libraries used for the implementation of the Scan-to-BIM (MEP) sub-component (at this stage).

Table 5: Tools and libraries used in the MEP detection subcomponent of Scan-to-BIM(MEP) tool

Library/Tool	Version	Licence
Tensorflow	1.15.2	*Apache 2*
Python	3.7.4	*BSD License*
LabelImg	1.8.0	*MIT License*
Google Colab	-	-
Agisoft Metashape	1.6.3	*Proprietary*
Flask	1.1.2	*BSD License*

5.3 API/Software Documentation

All the algorithms will be merged under a unique project and included as a single tool in the Scan+BIM Platform. However, individual components can also be run separately if needed by users.

- **detectMEPObject:** This component is a web service developed using Flask framework. This can be accessed via a url. Currently the web service is complete and runs on a localhost. We are currently in the process of deploying on a server so that it can be accessed from anywhere. This web service allows to upload an image as input and it outputs the detected object name with the corresponding bounding box.

5.4 Assumptions and Restrictions

The MEP component of the Scan-to-BIM tool has few assumptions and restrictions as follows:

- The MEP objects that have been considered are only radiators, sockets, and switches and the algorithm has been trained with a small size dataset. Training the algorithm with more data, better performance and generalisation can be achieved.
- Due to prominent difference in the structure of sockets and switches in UK (Type G Sockets) and EU (Type C, E, and F Sockets), a model has been trained for each of them separately. However, due to similar structure of radiators in both UK and EU, a model has been trained on the combined dataset and achieves better generalization.

5.5 Installation Instructions:

The Scan-to-BIM (MEP) tool, when delivered integrated to the Scan+BIM Platform, will be configured during the compilation of the overall solution.

5.6 Usage Walkthrough

For detection of MEP Object in an image the user needs to access the web page. As shown in Figure 15 the page contains a Browse button. By clicking the *Browse* button, the desired image can be uploaded. Then *Upload* button can be clicked to get the predicted output as shown in Figure 15. This also saves the detected object class and bounding box co-ordinates to a text file.

Note: Even though the web service is ready, it only runs on the localhost now. And the url for the web page will be available after deploying the web service to a cloud-based web server.

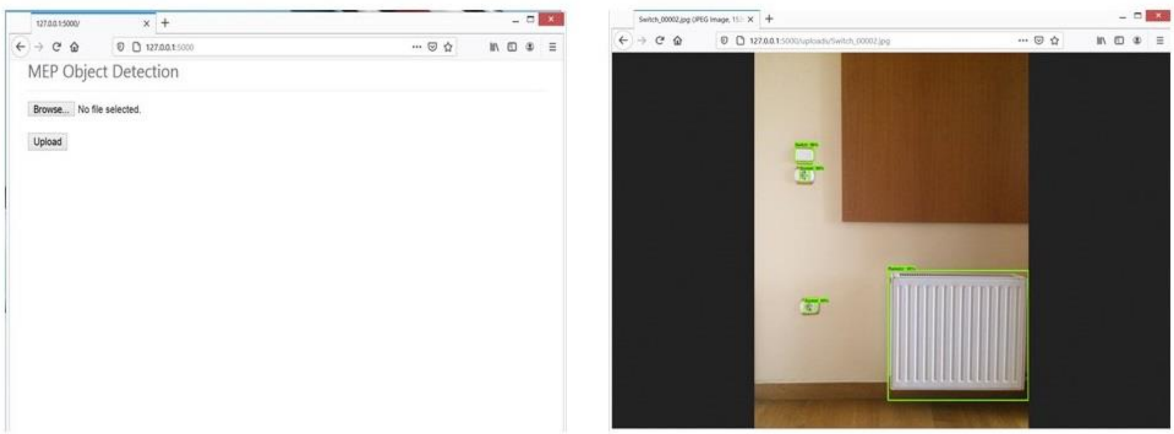


Figure 15: Web page for the object detection service.

3D reconstruction of scene from images is achieved by using Agisoft Metashape software. To load photos, select *Add Photos* command from the *Workflow* menu and browse the folder containing the images to select files to be processed. Unwanted images can be removed by right clicking the image and choosing *Remove* or by clicking the *X symbol* on the toolbar. Then, to align the photos, select *Align Photos* command from the *Workflow* menu and choose the desired accuracy level from the dialog box. Next, select *Build Dense Cloud* from *Workflow* menu and choose the desired quality from the dialog box to generate the dense cloud. To export the point cloud, select *Export Points* command from the *File* menu and browse the destination folder, choose file type, print file name and click *Save* button.

Exemplar Dataset

Initially, a set of images were collected from different UK and EU residences and University of Edinburgh campus to investigate and design the Scan-to-BIM (MEP) component. The dataset will be available upon request at Edinburgh Datashare. At the moment, our team are contacting the collaborators who provided us with images, following the General Data Protection Regulation (GDPR), to make as many of the photographs public. This dataset has been used to train the Faster R-CNN network to detect MEP objects in images.

A photogrammetric set of images from one of the pre-validation sites, Kripis House, has been acquired and pre-processed by CERTH. This set of images has been used to validate the trained model and for 3D reconstruction of the scene.

6. CONCLUSION AND PLAN FOR V2

The BIMERR Enhanced Scan-to-BIM Tools are to be employed at the outset of the BIMERR renovation process. The tools use as input digital documentation data of a residential building, that includes both laser scanning and photography (with significant overlap to support photogrammetric reconstruction). The tools then aim to almost fully automatically produce a Building Information Model of the building, that includes structural components (walls, floors, openings) and MEP components (e.g. radiators, sockets and switches). The BIM model is outputted by the tool in IFC-SPF format and contains information that is required by the RenoDSS Component primarily, as well as the ARIBFA Component. As documented in this deliverable (D5.3), the core functionalities of the BIMERR Enhanced Scan-to-BIM Tools are delivered by 3 main sub-components, namely the *Scan+BIM Software framework* that provides means to load, visualise and author BIM models (in IFC format) and digital documentation data (e.g. in E57 format); the *Scan-to-BIM (Structural) sub-component* to create the structural BIM model from the point cloud data; and the *Scan-to-BIM (MEP) sub-component* to enhance the previous BIM model with MEP components. The initial version of the BIMERR Enhanced Scan-to-BIM Tools is not complete with most remaining work being for Scan-to-BIM (MEP) sub-component. In contrast, the Scan-to-BIM (Structural) sub-component is well progressed and has already been tested with data from the Kripis House, one of the pre-validation sites.

The final release of the BIMERR Enhanced Scan-to-BIM Tools will be provided at M30 and shall include all missing functionalities for the Scan-to-BIM tools and shall be presented as an unified solution integrated within the Scan+BIM software framework (possibly aside from the 3D Reconstruction stage that is currently conducted using an existing third-party software package). It will also have been tested using data from both pre-validation sites.

The more relevant functionalities to be included in the final release of Scan-to-BIM encompass the following:

- Generate IFC models by using the semantic information extracted from point clouds of complex environments.
- Create 2nd Level Space Boundaries for complex environments, like those in the pre-validation sites.
- Map detected MEP object bounding boxes to the point cloud and codify the information to IFC model.
- Integrate extractStorey, calculateOpenings, IFCGenerator2LSB and detectMEPObject into the Scan+BIM platform.
- Establish communication with BIF for uploading the delivered IFC model.

BIBLIOGRAPHY

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation, pages 265-283.

Agisoft Metashape (2020). AgiSoft Metashape Professional Edition (Version 1.6.3) [Software]. Retrieved from http://www.agisoft.com/downloads/installer/.

CloudCompare (2020). CloudCompare (version 2.11) [GPL software]. Retrieved from https://www.cloudcompare.org/

Hetch, H. and Jaud, S. (2019). TUM OpenInfraPlatform: The Open-Source BIM Visualisation Software. In31. Forum Bauinformatik, Berlin, Germany, 2019

Huber, D. (2011) `The ASTM E57 file format for 3D imaging data exchange'. Proceedings of SPIE - The International Society for Optical Engineering. DOI: https://doi.org/10.1117/12.876555.

International Organization for Standardization. (2018). ISO 16739-1:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema. ISO, November 2018.

Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Dolĺar, P. and Zit-nick, C. L. (2014). Microsoft COCO: common objects in context. arXive-prints, arXiv:1405.0312 [cs.CV].

Lockley, S., Benghi, C., Černý M. (2017). Xbim.Essentials: a library for interoperable building information applications. Journal of Open Source Software, 2(20), 473, https://doi.org/10.21105/joss.00473.

Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In NIPS.

Solibri, Inc. (2020). Solibri Anywhere (version 9.10). https://www.solibri.com/.

UNC LibGuides (n.d.). Using Agisoft Metashape. Retrieved from https://guides.lib.unc.edu/metashape.

Valero, E., Mohanty, D.D. and Bosché, F. (2020). Development of an Open Source Scan&BIM Platform. Submitted to ISARC 2020.

Wu, C. (2011). VisualSFM: A Visual Structure from Motion System. Retrieved from http://www.cs.washington.edu/homes/ccwu/vsfm/.

Zoph, B., Brain, G., Vasudevan, V., Shlens, J., and Le, Q. V. Google Brain. (2018). Learning Transferable Architectures for Scalable Image Recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8697-8710.