



Project Acronym: **BIMERR**  
 Project Full Title: **BIM-based holistic tools for Energy-driven Renovation of existing Residences**  
 Grant Agreement: **820621**  
 Project Duration: **45 months**

## DELIVERABLE D5.4

### Innovative Scan-to-BIM tools for Automated BIM v2

Deliverable Status: **Final**  
 File Name: **BIMERR DELIVERABLE D5.4\_v0.5.docx**  
 Due Date: **30/06/2021 (M30)**  
 Submission Date: **29/06/2021**  
 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          |

#### **Disclaimer**

*BIMERR project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission (EC). EC is not liable for any use that may be made of the information contained therein.*

## AUTHORS LIST

| Leading Author (Editor)          |         |            |             |  |
|----------------------------------|---------|------------|-------------|--|
| Surname                          |         | First Name | Beneficiary | Contact email  |
| Valero                           |         | Enrique    | UEDIN       | <a href="mailto:e.valero@ed.ac.uk">e.valero@ed.ac.uk</a>   |
| Co-authors (in alphabetic order) |         |            |             |  |
| #                                | Surname | First Name | Beneficiary | Contact email  |
| 1                                | Bosché  | Frédéric   | UEDIN       | <a href="mailto:f.bosche@ed.ac.uk">f.bosche@ed.ac.uk</a>   |
| 2                                | Mohanty | Dibya      | UEDIN       | <a href="mailto:d.mohanty@ed.ac.uk">d.mohanty@ed.ac.uk</a> |
| 3                                | Ceklarz | Michal     | UEDIN       | <a href="mailto:s1762931@ed.ac.uk">s1762931@ed.ac.uk</a>   |
| 4                                | Tao     | Boan       | UEDIN       | b.tao@sms.ed.ac.uk   |
| 5                                | Fenz    | Stefan     | XYLEM       | fenz@xylem-technologies.com                                |
| 6                                | Rovas   | Dimitrios  | UCL         | d.rovas@ucl.ac.uk  |

## REVIEWERS LIST

| List of Reviewers (in alphabetic order) |              |            |             |  |
|---|--------------|------------|-------------|--|
| #                                       | Surname      | First Name | Beneficiary | Contact email  |
| 1                                       | Katsigarakis | Kyriakos   | UCL         | <a href="mailto:k.katsigarakis@ucl.ac.uk">k.katsigarakis@ucl.ac.uk</a> |
| 2                                       | Tsakiris     | Thanos     | CERTH       | <a href="mailto:atsakir@iti.gr">atsakir@iti.gr</a>                     |
| 3                                       | Lilis        | Georgios   | UCL         | <a href="mailto:g.lilis@ucl.ac.uk">g.lilis@ucl.ac.uk</a>               |
| 4                                       | Pantraki     | Evangelia  | CERTH       | <a href="mailto:epantrak@iti.gr">epantrak@iti.gr</a>                   |

## REVISION CONTROL

| Version | Author | Date       | Status                        |
|---------|--------|------------|-------------------------------|
| 0.01    | UEDIN  | 07.04.2021 | Setup                         |
| 0.1     | UEDIN  | 25.05.2021 | First draft                   |
| 0.2     | UEDIN  | 04.06.2021 | Conclusions + Final structure |
| 0.3     | UEDIN  | 07.06.2021 | For review                    |
| 0.4a    | CERTH  | 15.06.2021 | Review CERTH                  |
| 0.4b    | UCL    | 24.06.2021 | Review UCL                    |
| 0.5     | UEDIN  | 25.06.2021 | Final version                 |

## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <i>List of Figures</i> .....                                   | <b>6</b>  |
| <i>List of Tables</i> .....                                    | <b>7</b>  |
| <i>ACRONYMS</i> .....  | <b>8</b>  |
| <i>Executive Summary</i> .....                                 | <b>9</b>  |
| <b>1 Introduction</b> .....                                    | <b>10</b> |
| 1.1 Scope and Objectives of the Deliverable .....              | <b>10</b> |
| 1.2 Relation to other tasks/deliverables .....                 | <b>10</b> |
| 1.3 Changes to the First Version of the Scan-to-BIM Tool ..... | <b>13</b> |
| <b>2 Scan-to-BIM Overview</b> .....                            | <b>14</b> |
| <b>3 Scan-to-BIM Structural Sub-component</b> .....            | <b>15</b> |
| 3.1 Overview .....   | <b>15</b> |
| 3.2 Technology Stack and Implementation Tools .....            | <b>19</b> |
| 3.3 API/Software Documentation .....                           | <b>19</b> |
| 3.4 Assumptions and Restrictions .....                         | <b>19</b> |
| 3.5 Installation Instructions .....                            | <b>20</b> |
| 3.6 Usage Walkthrough .....                                    | <b>20</b> |
| 3.7 Licensing .....  | <b>21</b> |
| <b>4 Scan-to-BIM MEP Sub-component</b> .....                   | <b>22</b> |
| 4.1 Overview .....   | <b>22</b> |
| 4.2 Technology Stack and Implementation Tools .....            | <b>25</b> |
| 4.3 API/Software Documentation .....                           | <b>26</b> |
| 4.4 Assumptions and Restrictions .....                         | <b>27</b> |
| 4.5 Installation Instructions .....                            | <b>27</b> |
| 4.6 Usage Walkthrough .....                                    | <b>27</b> |
| <b>5 Scan-to-BIM Editor Sub-component</b> .....                | <b>30</b> |
| 5.1 Overview .....   | <b>30</b> |
| 5.2 Technology Stack and Implementation Tools .....            | <b>31</b> |

|          |  |           |
|----------|--|-----------|
| 5.3      | API/Software Documentation.....                | 32        |
| 5.4      | Assumptions and Restrictions.....              | 32        |
| 5.5      | Installation Instructions .....                | 32        |
| 5.6      | Usage walkthrough .....                        | 32        |
| 5.7      | Licensing .....                                | 33        |
| <b>6</b> | <b><i>Scan-to-BIM Interface .....</i></b>      | <b>34</b> |
| 6.1      | Overview.....                                  | 34        |
| 6.2      | Technology Stack and Implementation Tools..... | 34        |
| 6.3      | API/Software Documentation.....                | 35        |
| 6.4      | Assumptions and Restrictions.....              | 35        |
| 6.5      | Installation Instructions .....                | 35        |
| 6.6      | Usage walkthrough .....                        | 36        |
| 6.7      | Licensing .....                                | 38        |
| <b>7</b> | <b><i>Conclusion.....</i></b>                  | <b>39</b> |
|          | <b><i>Bibliography.....</i></b>                | <b>40</b> |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 1: Overview of the overall Scan-to-BIM Tool/process .....   | 14 |
| Figure 2: Overview of the Scan-to-BIM Structural process .....   | 15 |
| Figure 3: Coloured point cloud of a two-storey building (Kripis house) .....   | 15 |
| Figure 4: 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. ....   | 16 |
| Figure 5: Structural Component Detection for Kripis house: (a) slabs dividing the two storeys; (b) detected spaces (i.e. 1LSB representation); and (c) modelled walls for both storeys. ....   | 17 |
| Figure 6: Opening Detection: (a) Openings labelled as windows (blue) and doors (brown); and (b) BIM model of Kripis house, showing the walls with openings. ....   | 18 |
| Figure 7: IFC model: (a) 2LSBs obtained by the Scan-to-BIM Structural tool for the Kripis house model; (b) Structural components of the BIM model outputted by the Scan-to-BIM Structural sub-component and loaded on Solibri Anywhere (Solibri Inc., 2020)..... | 18 |
| Figure 8: Overview of the Scan-to-BIM MEP process .....  | 22 |
| Figure 9: Some MEP object detection results. ....  | 23 |
| Figure 10: 3D Reconstruction of scene from images.....   | 24 |
| Figure 11: Reprojection of detected bounding boxes to point cloud .....  | 25 |
| Figure 12: Web page for the object detection service.....  | 28 |
| Figure 13: Overview of the Scan-to-BIM Editor process .....  | 30 |
| Figure 14: Screenshots of the Scan-to-BIM Editor: (a) Main window; (b) Dialog for defining layers and materials to structural elements; (c) Dialog for assigning properties to MEP objects; (d) Classification of spaces into zones.....                         | 31 |
| Figure 15: Steps to produce an IFC model from the Scan-to-BIM Interface: (a) “Scan-to-BIM” is pressed in the Interface; (b) parameters are input for both Structural and MEP processes.....  | 37 |
| Figure 16: Steps for updating the local files containing the information of the BIMERR materials/properties database.....  | 37 |
| Figure 17: Steps for uploading an IFC file to the BIF: (a) Click on “Upload to BIF” in the main window; and (b) input the required parameters.....   | 38 |

## LIST OF TABLES

|  |    |
|--|----|
| Table 1: User Requirements relevant to the Scan-to-BIM Tool and the level to which they shall be met. .... | 12 |
| Table 2: Tools and libraries used in the structural sub-component of Scan-to-BIM. ....                     | 19 |
| Table 3: MEP object detection performance. ....  | 24 |
| Table 4: Tools and libraries used in the MEP detection sub-component of Scan-to-BIM MEP tool ....          | 26 |
| Table 5: Tools and libraries used in Scan-to-BIM Editor. ....  | 32 |
| Table 6: Tools and libraries used in Scan-to-BIM Interface.....  | 35 |
| Table 7: Processing times for the generation of a BIM model of Kripis House.....                           | 36 |

## ACRONYMS

| Acronym | Meaning  |
|---------|--|
| 1LSB    | First Level Space Boundary   |
| 2LSB    | Second Level Space Boundary  |
| AC      | Air Conditioning   |
| API     | Application Programming Interface  |
| ARIBFA  | Augmented-Reality Enabled In-situ Building Feature Annotation tool           |
| 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   |
| COCO    | Common Objects in Context  |
| EU      | European Union   |
| GUI     | Graphical User Interface   |
| HVAC    | Heating, Ventilation and Air Conditioning                                    |
| IFC     | Industry Foundation Classes  |
| IOU     | Intersection Over Union  |
| mAP     | Mean Average Precision   |
| MEP     | Mechanical, Electrical and Plumbing  |
| PG      | Photogrammetry   |
| PWMA    | Process & Workflow Modelling and Automation toolkit                          |
| R-CNN   | Regional Convolutional Neural Network  |
| RenoDSS | BIMERR Renovation Decision Support System                                    |
| SfM     | Structure from Motion  |
| SPF     | STEP Physical file   |
| T       | Task   |
| TLS     | Terrestrial Laser Scanning   |
| UC      | Use Case   |
| UEDIN   | The University of Edinburgh  |
| UK      | United Kingdom   |
| UR      | User Requirement   |
| WP      | Work Package   |



## EXECUTIVE SUMMARY

The BIMERR Deliverable D5.4 “Innovative Scan-to-BIM Tools for Automated BIM Generation v2” documents the BIMERR Scan-to-BIM tools and concludes the second 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 start 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. This tool is composed of three 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; 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; and the *Scan-to-BIM Editor*, which populates BIM models with information that cannot be retrieved by means of computer vision techniques but is required for renovation (e.g. materials of structural components or power of HVAC systems).

The present documentation of the BIMERR Scan-to-BIM Tool along with its sub-components 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. In this second release, the BIMERR Scan-to-BIM Tool implements all the envisaged functionalities and is presented as three tools (i.e. pieces of software): a first one to produce an initial BIM model from point clouds delivered by laser scanners; a second one to create a 3D reconstruction of the spaces from pictures of the building and subsequently identify and model MEP entities, which will be included in the BIM model; and a third one to edit the model and populate it with additional information required for the renovation process. Results from the pre-validation sites, especially the Kripis house, are shown in the deliverable.

## 1 INTRODUCTION

### 1.1 SCOPE AND OBJECTIVES OF THE DELIVERABLE

This deliverable reports on the work conducted up to M30, especially emphasising the work performed between M20 and M30, 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.

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

- **Scan-to-BIM Structural sub-component:** 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 tool employs automated processes only.
- **Scan-to-BIM MEP sub-component:** the data processing tool that takes as input the scan data (principally visible light pictures) and outputs Mechanical, Electrical and Plumbing (MEP) components (e.g. radiators, switches, lights) that are integrated in the structural model to produce an enriched BIM model. The tools employ automated processes only.
- **Scan-to-BIM Editor sub-component:** the developed tool to populate BIM models with material information and MEP properties, as well as cluster spaces into zones.

The three sub-components introduced above have been subsequently integrated in an umbrella component with Graphical User Interface (GUI): the Scan-to-BIM Interface. Therefore, from the GUI, the user is able to: call the Scan-to-BIM Structural and Scan-to-BIM MEP sub-components listed above; complete the BIM model with material information and additional properties; and communicate with the BIMERR Interoperability Framework (BIF) to upload the produced IFC file.

### 1.2 RELATION TO OTHER TASKS/DELIVERABLES

This deliverable reports the second version (v2) of the Scan-to-BIM Tool, improving the sub-components presented in the Deliverable D5.3 published in M20.

#### 1.2.1 Relation to other BIMERR Tools

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

- The **Augmented-Reality Enabled In-situ Building Feature Annotation (ARIBFA)** tool to complete/correct the BIM model on site (WP5; Deliverables D5.9 and D5.10), and
- the **Renovation Decision Support System (RenoDSS)** tool for assessing the energy performance of an existing building and designing its energy-focused refurbishment (WP7; Deliverables D7.1 and D7.2).

Other BIMERR Tools (e.g. Building Information Collection Application (BICA), Process & Workflow Modelling and Automation toolkit (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, and visible light 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).

### **1.2.2 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             | 100%        |
| 12  | Architect  | Architect shall be able to have an up-to-date representation of the building to be renovated                          | High          | Yes             | 100%        |
| 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 | To be confirmed | N/A         |
| 148 | Surveyor   | Surveyors shall be able to use open formats for delivering as-is data   | High          | Yes             | 100%        |



### 1.3 CHANGES TO THE FIRST VERSION OF THE SCAN-TO-BIM TOOL

In the Deliverable 5.3, submitted in M20, a number of functionalities were remarked to be included in the second version of the Scan-to-BIM Tool. In this section, the functionalities added (or modifications made) to the initial version of the tool are summarised:

- Scan-to-BIM Structural sub-component: The tool is now able to deliver IFC models automatically generated from point clouds of complex environments, after all the algorithms have been included in the same piece of software. Additionally, 2<sup>nd</sup> Level Space Boundaries of more complex environments (i.e. containing surfaces with a number of corresponding boundaries) are now automatically created by the Scan-to-BIM Tool. This is especially challenging in the case of buildings with more than one storey with non-rectangular spaces, as is the case of the Kripis house.
- Scan-to-BIM MEP sub-component: The web service for MEP object detection is now deployed on Eleanor Cloud Service and can be accessed via URL. The re-projection of detected MEP components in image to the point cloud and codification of the same in IFC file is now complete. All the components are integrated and are available as two pieces of executables.
- IFC models can now be modified using the newly produced Scan-to-BIM Editor sub-component. That tool allows the user to annotate models with material information for walls, slabs, windows, and doors, and assign properties to MEP components. Furthermore, the Editor enables grouping spaces of the model into zones.
- The three sub-components listed above have been integrated in an umbrella component with GUI, which can be used to: 1) execute the Scan-to-BIM processes (Structural, MEP or both); 2) connect to a database containing material and MEP information and download its content to be used for annotation (see Deliverables D7.1 and D7.2 for further information); 3) perform the Editor tasks; and 4) connect to the BIF, via the platform's API, to upload the generated IFC file.



## 2 SCAN-TO-BIM OVERVIEW

The aim of the proposed Scan-to-BIM Tool, which has been conceived as a unified software solution with GUI, 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/properties databases).

Scan-to-BIM, as illustrated in Figure 1, 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 three 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.
- **Scan-to-BIM Editor sub-component:** this sub-component enables the user to take as input an IFC model and manually annotate properties to entities - properties that cannot be automatically identified by means of computer vision.

As mentioned in the Section 1.3, the three tools defining the core of Scan-to-BIM have been integrated under an umbrella component. This component, called **Scan-to-BIM Interface**, provides a GUI to (1) facilitate the interaction of the user with the proposed solution for the generation of IFC files, (2) connect to the BIMERR material/components, and (3) communicate with the BIF to subsequently upload the produced BIM model.

The above sub-components are presented in Sections 2, 3, and 4, respectively.

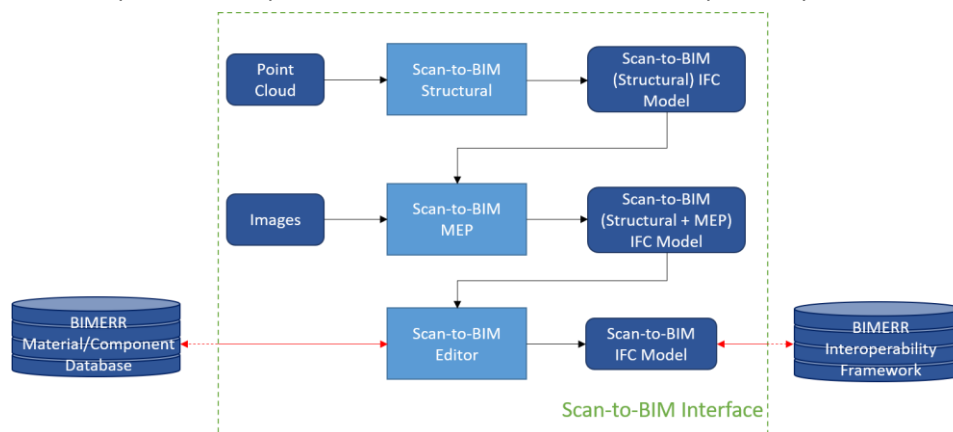


Figure 1: Overview of the overall Scan-to-BIM Tool/process

### 3 SCAN-TO-BIM STRUCTURAL SUB-COMPONENT

#### 3.1 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 2, the tool is divided into a number of tasks (or algorithms), which analyse the building from general to specific.

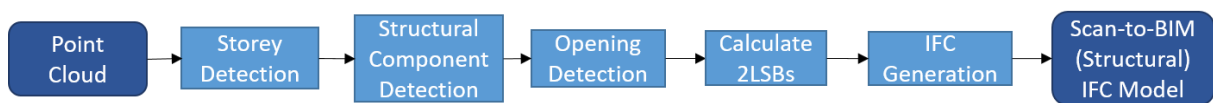


Figure 2: Overview of the Scan-to-BIM Structural process

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 3) is the input to the Scan-to-BIM Structural sub-component process. The process goes through the successive steps:

- **Storey detection:** 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 4.



Figure 3: Coloured point cloud of a two-storey building (Kripis house)

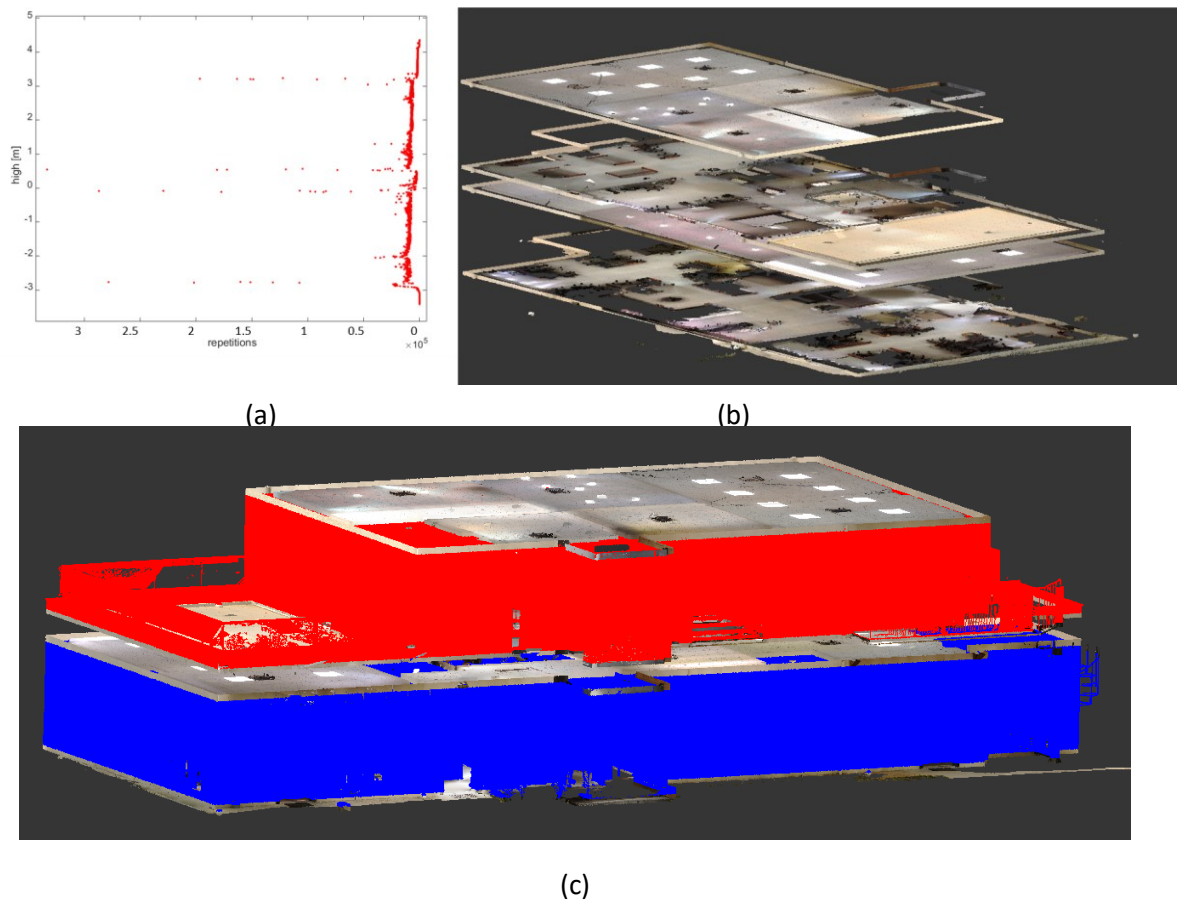


Figure 4: Storey Detection: (a) Histogram showing density of points along the vertical axis; (b) segmented floors and ceilings of a building (right); and (c) overall point cloud segmented into storeys.

- **Structural component detection:** Once the global point cloud is segmented into storeys, each individual cloud representing a storey of the building is processed to define 1) the surfaces of the walls visible to the scanner and 2) the slabs dividing the building into storeys.

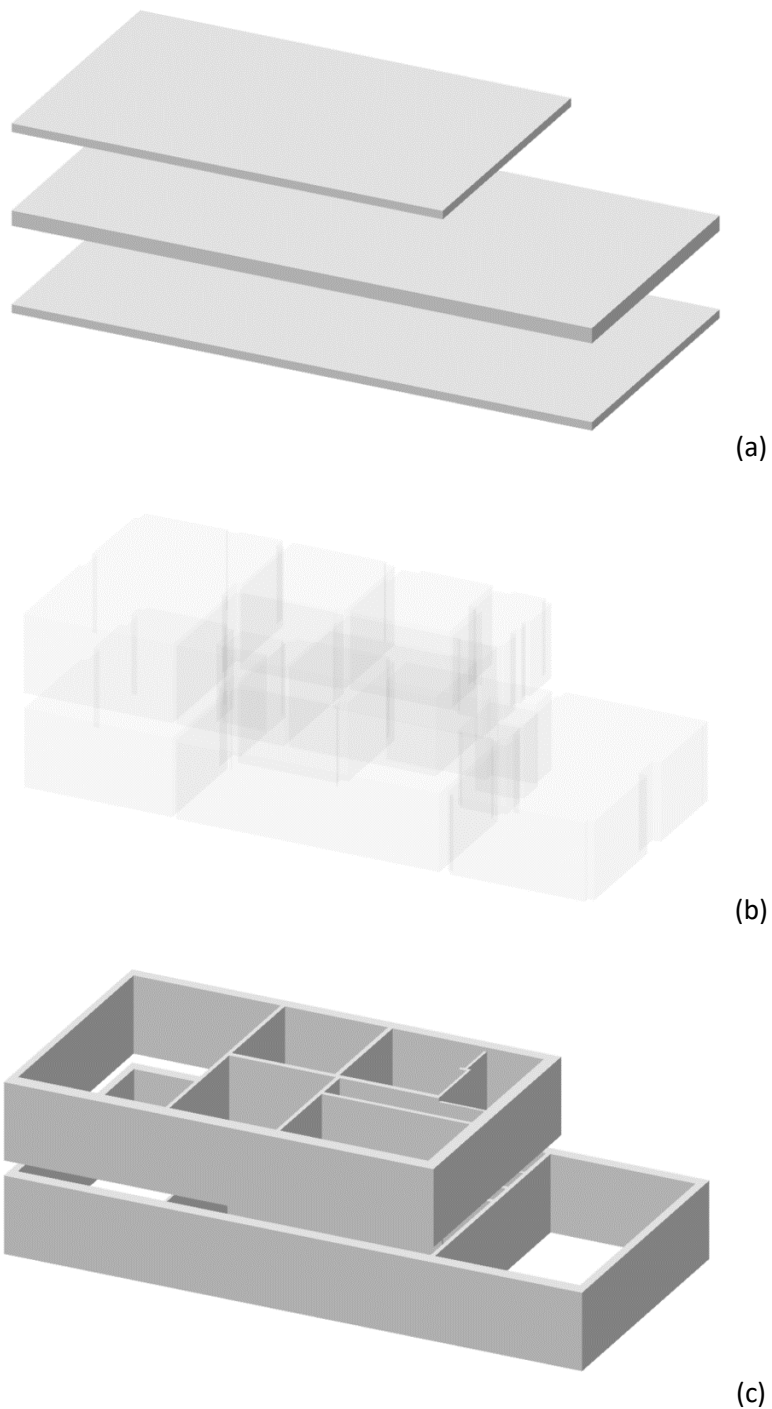
Initially, points with z-values corresponding to those of the floors and ceilings (see *Storey detection*), together with the outer surfaces of exterior walls, are used to define the slabs (Figure 5 (a)). Then, the points corresponding to the ceiling of each storey are extracted and subsequently clustered (to define spaces). Note that, the lack of points inside the wall elements facilitates the identification of clusters, and therefore, 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), which define the Boundary Representation (i.e. 1LSB) for each space, as shown in Figure 5(b).

Finally, the two sides of each wall are matched by searching sets of parallel vertical planes with opposite normal vectors. If a wall polygon has no matching plane, then, an artificial surface is



created to fully define the wall. The outcome of this process is the set of 3D walls for the given storey, as shown in Figure 5(c).



**Figure 5: Structural Component Detection for Kripis house: (a) slabs dividing the two storeys; (b) detected spaces (i.e. 1LSB representation); and (c) modelled walls for both storeys.**

- **Opening Detection:** After all the spaces and walls are defined (i.e. the 1<sup>st</sup> Level Space Boundaries (1LSBs)), opening elements (e.g. doors and windows) are identified for walls by processing point clouds neighbouring both sides of each wall. Bounding boxes are calculated for all empty volumes and openings are detected based on the overlap between pairs of bounding boxes on both sides of the walls. Figure 6 illustrates the results for opening detection in Kripis house.

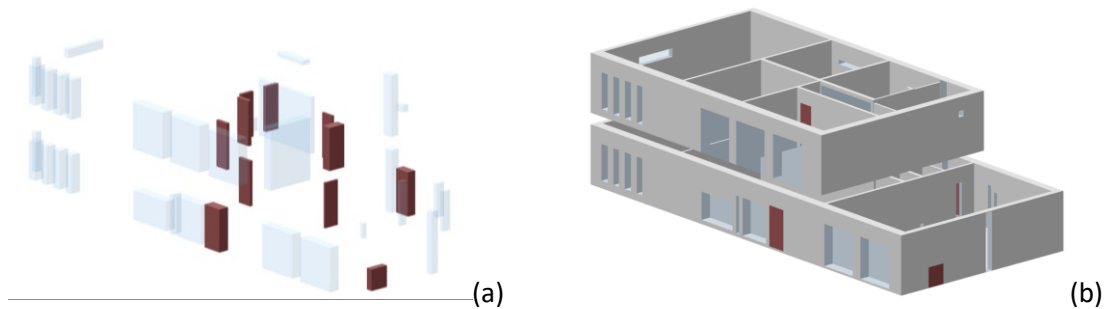


Figure 6: Opening Detection: (a) Openings labelled as windows (blue) and doors (brown); and (b) BIM model of Kripis house, showing the walls with openings.

- **Calculate 2<sup>nd</sup> Level Space Boundaries (2LSBs):** Once all the structural elements have been defined and relationships between certain entities (e.g. a wall and its openings) are established, geometric and relational information is used to define the 2LSBs, which is required for energy-related calculations. Figure 7(a) shows 2LSBs calculated for the Kripis house.
- **IFC Generation:** After structural entities are parametrised, and 2LSBs are defined, this information is codified into an IFC-SPF file, following the specifications defined by buildingSmart in the standard ISO 16739 (International Organization for Standardization, 2018). Figure 7(b) illustrates the structural part of the BIM model delivered by Scan-to-BIM Structural after being loaded into Solibri Anywhere (Solibri Inc. 2020), an IFC viewer widely used in industry.

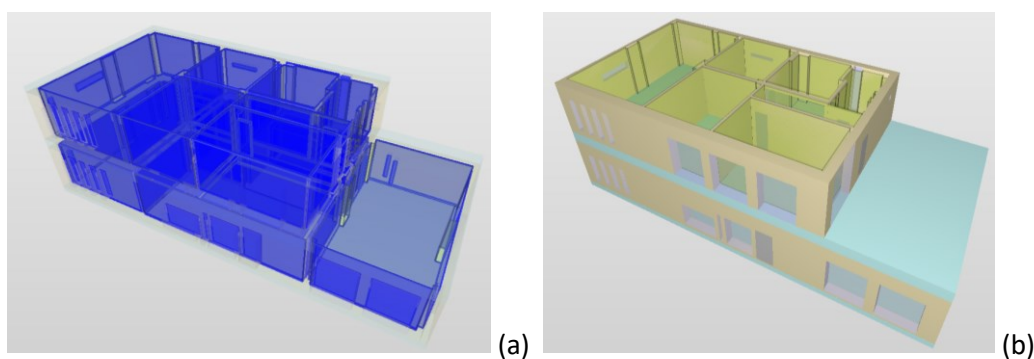


Figure 7: IFC model: (a) 2LSBs obtained by the Scan-to-BIM Structural tool for the Kripis house model; (b) Structural components of the BIM model outputted by the Scan-to-BIM Structural sub-component and loaded on Solibri Anywhere (Solibri Inc., 2020).

### 3.2 TECHNOLOGY STACK AND IMPLEMENTATION TOOLS

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

- Identification and parametrisation of structural components, and further analysis of the relationships amongst these.
- Codification of the obtained information into IFC-SPF files, following the IFC4/IFC4x1 version.

The Scan-to-BIM Structural sub-component will be run as a standalone project, being invoked from the command prompt or from the Scan-to-BIM Interface. The first subpart (i.e. identification of structural elements) 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 subpart, written in C#, reads the pseudo-IFC file and produces IFC files. Libraries used in the above-mentioned projects are enumerated in Table 2.

**Table 2: Tools and libraries used in the structural sub-component of Scan-to-BIM.**

| Library                | Version | Licence     |
|------------------------|---------|-------------|
| <b>Eigen</b>           | 3.3.7   | MPL2        |
| <b>OpenCV</b>          | 4.1.2   | Apache 2    |
| <b>PCL</b>             | 1.9.1   | BSD Licence |
| <b>Xbim.Essentials</b> | 5.1.297 | CDDL        |

### 3.3 API/SOFTWARE DOCUMENTATION

All the processes (i.e. algorithms) enumerated in Section 3.1 have been merged under a unique project and included as a single tool in the Scan-to-BIM Interface. However, Scan-to-BIM Structural can be also run from the command prompt.

The program takes a point cloud of a building as a main input and delivers an IFC file, including all the structural elements identified by the algorithm. Binaries can be found at <https://datashare.ed.ac.uk/handle/10283/3938> (s2b folder).

### 3.4 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.
- It currently supports .ply format for the input point cloud data, although work is on-going on supporting the E57 format as well.

- Storeys are expected to be stacked up along the z-axis.
- Both floors and ceilings (i.e. slabs) are assumed to be horizontal and walls are assumed to be vertical.
- In the case of suspended ceilings, since these cannot be distinguished from slabs in TLS data, they have been considered as surfaces belonging to slabs.
- Although openings on slabs (e.g. staircases) have not been identified, work is on-going.
- Balconies and other objects on the façades of buildings are not currently being analysed.

### 3.5 INSTALLATION INSTRUCTIONS

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

### 3.6 USAGE WALKTHROUGH

The standalone project can be 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 *s2bStruct.exe*, which corresponds to the Scan-to-BIM structural sub-component, the user must type the name of the .exe file of the tool followed by the input arguments. These arguments include: (1) the path to the file containing the point cloud, (2) voxel size for wall segmentation in metres, (3) resolution of the cloud in centimetres, (4) voxel size for opening detection in metres, and if MEP objects have been identified, (5) the path to the file containing MEP-related information, which has been delivered by the Scan-to-BIM MEP sub-component. An example is given in the following.

```
>> s2bStruct.exe "C:/Path/to/your/file/pointCloud.ply" 0.04 1 0.05 "C:/Path/to/your/file/pseudo-MEP.txt"
```

As previously mentioned, the Scan-to-BIM Structural sub-component has been embedded into the Scan-to-BIM Interface (See Section 6). For running the tool, the user only needs to click on the "Scan-to-BIM" icon and a dialog window will appear to ask for the required input arguments (as illustrated in Figure 15, within Section 6).

#### 3.6.1 Exemplar Dataset

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 were 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 the Scan-to-BIM Structural sub-component ([https://bscw.fit.fraunhofer.de/sec/bscw.cgi/d52510650/Smart\\_Home\\_cleanOut\\_1cm.ply](https://bscw.fit.fraunhofer.de/sec/bscw.cgi/d52510650/Smart_Home_cleanOut_1cm.ply)).

### 3.7 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>).

## 4 SCAN-TO-BIM MEP SUB-COMPONENT

### 4.1 OVERVIEW

The objective of Scan-to-BIM MEP sub-component is the detection of the MEP components of buildings in images of the environment, and subsequently adding these entities to the IFC-SPF files generated by the Scan-to-BIM Structural sub-component. The process includes several sub-processes as shown in Figure 8.

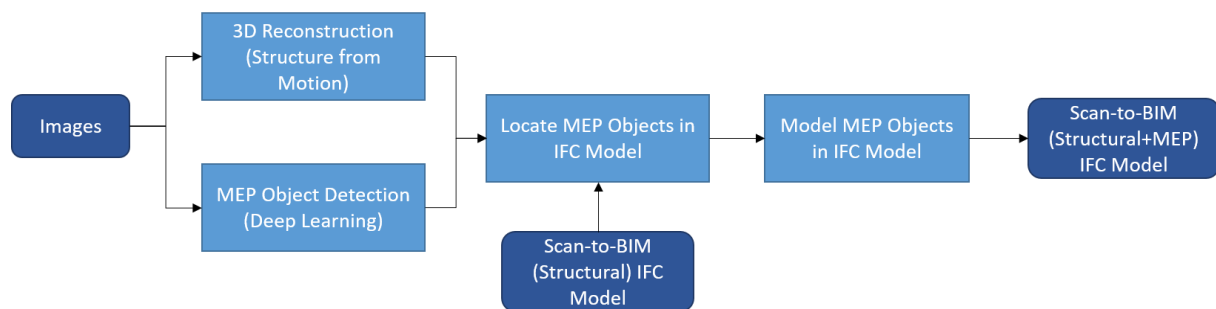


Figure 8: Overview of the Scan-to-BIM MEP process

The process encompasses the following steps:

- MEP Object Detection:** The first step of the Scan-to-BIM MEP process is the detection of different MEP components (e.g. radiators, AC units, 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 and AC units, mean average precision (mAP) of 1 and .838 respectively at 50%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 3. Figure 9 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, 2020). 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. This step outputs a reconstructed point cloud of the scene and the camera parameters which are the input for the next step. Figure 10 shows a result of 3D reconstruction of a room from a set of photogrammetric images. In case a 360° image from the laser scanner is available, there is no need for a photogrammetric set of images and reconstruction of the scene from these images, as the 360° image is already aligned with the point cloud.



**Figure 9: Some MEP object detection results.**



Figure 10: 3D Reconstruction of scene from images.

Table 3: 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        |
| Aircon              | 2062          | .838    | .791    | .670        |

- Locate MEP Objects in IFC Models:** 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. This step involves manually locating at least 4 points in the point cloud for corresponding pixels in any one image from the photogrammetry set to find the transformation required for establishing 2D-3D correspondence (i.e. re-projection). This operation is applicable to all the images in the photogrammetric set and hence needs to be done only for one image per a photogrammetric image set. Once this transformation and camera external calibration matrices are available,



each pixel in the image can be mapped to the corresponding point in the Scan-to-BIM Structural model. Figure 11 shows the localization of MEP components in the point cloud.



Figure 11: Reprojection of detected bounding boxes to point cloud

- **Model MEP Objects in IFC Models:** 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 coordinates, the detected object category and information about the object on which the previously extracted information is mapped.

## 4.2 TECHNOLOGY STACK AND IMPLEMENTATION TOOLS

The Scan-to-BIM MEP sub-component has five main sub-parts:

- Detection of MEP objects in images and representation as bounding boxes
- Photogrammetric 3D Reconstruction of the scene
- Re-projection of the detected bounding boxes onto the point cloud (as illustrated in Figure 11), resulting in four 3D points per MEP object
- Projection of the obtained 3D points onto the nearest neighbouring walls/slabs polygons to define the volume of each entity.
- Codification of the information to the IFC file

The detection of MEP objects in images uses deep learning, has been 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. The MEP object detection sub-component is now deployed as a web service on Eleanor Cloud Service provided by University of Edinburgh. For photogrammetric 3D reconstruction of the scene from the images, Agisoft Metashape (Agisoft Metashape, 2020) is used. The third component is the same C# tool (based on the xBIM library) used in the Scan-to-BIM Structural sub-component to generate IFC files. Table 4 summarises the tools and libraries used for the implementation of the Scan-to-BIM MEP sub-component.

**Table 4: Tools and libraries used in the MEP detection sub-component of Scan-to-BIM MEP tool**

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

### 4.3 API/SOFTWARE DOCUMENTATION

All the algorithms will be merged under a unique project callable from the Scan-to-BIM Interface. However, the Scan-to-BIM MEP component can also be run separately if needed by users.

- **MEP Object Detection Web Service:** The web service is deployed on Eleanor Cloud Service of University of Edinburgh. This web service allows to upload an image as input and returns the detected object name with the corresponding bounding box. This can be accessed both via web page and API from code.
- **detectMEP:** This component is an executable which takes the photogrammetric image set, reconstructed point cloud and the scaling factor as input and returns the real world coordinates for the MEP objects in the scene.
- **findScale:** This component is a pre-requisite for the *detectMEP* component. This is an executable which takes the coordinates of, at least, 4 pixels of any image from the photogrammetric set and corresponding world coordinates and the camera parameter for the same image as input. Then it returns the scaling factor required for re-projection of pixel coordinates to world coordinates.

The binaries for *detectMEP* and *findScale* are available at <https://datashare.ed.ac.uk/handle/10283/3938> (*mep* folder).

#### 4.4 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, AC units, 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.

#### 4.5 INSTALLATION INSTRUCTIONS

No complex installation required other than installing the Agisoft Metashape software ([Installation guide](#)) which is required for photogrammetric reconstruction. The binaries are available on the CyberBuild Datashare collection.

#### 4.6 USAGE WALKTHROUGH

##### 4.6.1 MEP Object Detection Web Service

The web service can be used as an independent component. The user can access the component deployed as web service via a webpage or from code through an API using the URL (<http://129.215.193.54:5000/>). As shown in Figure 12, the page contains a *Browse* button. By clicking the *Browse* button, the desired image can be uploaded. Then the *Upload* button can be clicked to get the predicted output as shown in Figure 12. When used from a code using an API, the web service returns the detection output in JSON format. To access the web service through code (for example, Python) the following script can be used.

```
import requests
image = {"file": open(PATH_TO_INPUT_IMAGE, "rb")}
results = requests.post(url, files=image)
print(results.text)
```

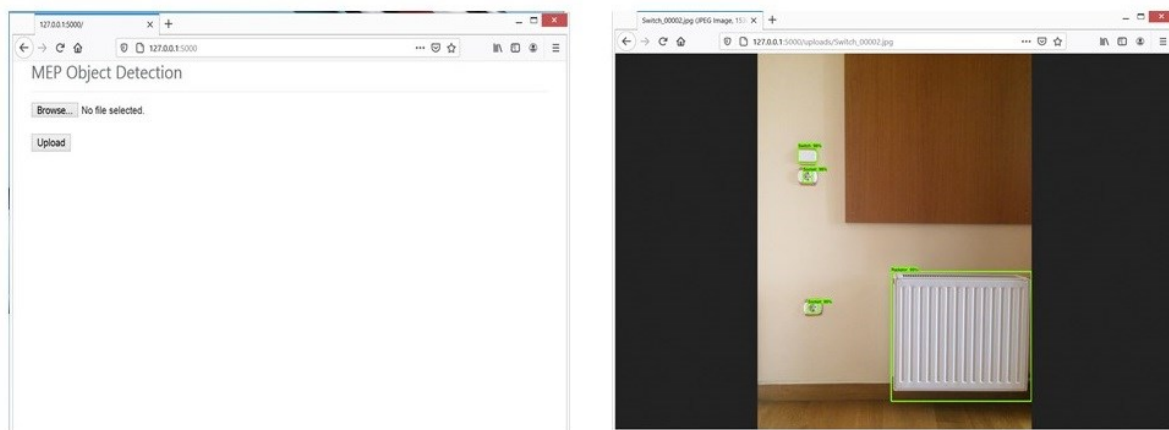


Figure 12: Web page for the object detection service.

#### 4.6.2 MEP Object Detection and Re-projection

The whole pipeline for detecting MEP components in images and re-projecting them to the point cloud involves 3 steps. The 3D reconstruction of the scene from images, finding the scaling factor required for re-projection and re-projection of the detected MEP components in images to the point cloud.

The 3D reconstruction of a 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. To export the camera parameters select *Export Camera* command from *File* menu and then, in the browser, select the *Save as type* option to .nvm from the drop down list and save it as camera .nvm.

Once the camera parameters (camera.nvm) and the reconstructed point cloud are available, the user needs to find the scaling factor that is required for the re-projection of detected objects in image to point cloud. For this step, the user needs to map minimum 4 points in one image from the photogrammetry set to the point cloud manually. These 4 pixels’ coordinates and world coordinates along with the image name need to be saved into a text file (pixel-to-world-map.txt) and then passed to the *findScale* executable as an argument. The *findScale* returns a text file (scale.txt) with the value of scaling factor. If all images in the photogrammetric set are acquired using the same camera, then the scaling factor needs only be calculated once with any one image from the set. It can then be applied to all other images automatically.

With the scaling factor available, the *detectMEP* executable can be used to detect MEP components in images and re-project them to point cloud. The executable needs the photogrammetry image set

directory (*Images*), camera parameters (*camera.nvm*) and scaling factor (*scale.txt*) in the same directory. It can be executed from command prompt, and it returns a text file *pseudo-MEP.txt* with all the detected MEP objects, their corresponding world-coordinates and camera position.

To execute *detectMEP* the user needs to navigate to the directory (Path/to/detectMEP.exe) in the command prompt and type the following command:

```
>>detectMEP.exe
```

#### **4.6.3 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. 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 and re-projection of detected MEP objects in the images to the point cloud.

## 5 SCAN-TO-BIM EDITOR SUB-COMPONENT

### 5.1 OVERVIEW

Scan-to-BIM Editor was initially conceived as a supplement to *Scan-to-BIM Structural*, aiming to deliver a more complete IFC file, including information about layers and materials of the structural components of the building. In summary, to manually provide information to a BIM model (as shown in Figure 13), which cannot be extracted by means of techniques related to computer vision.

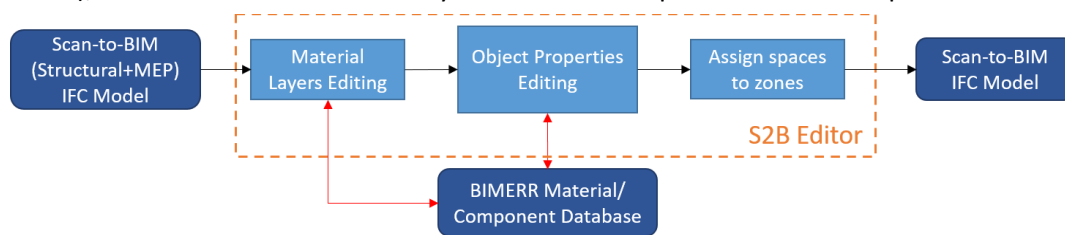
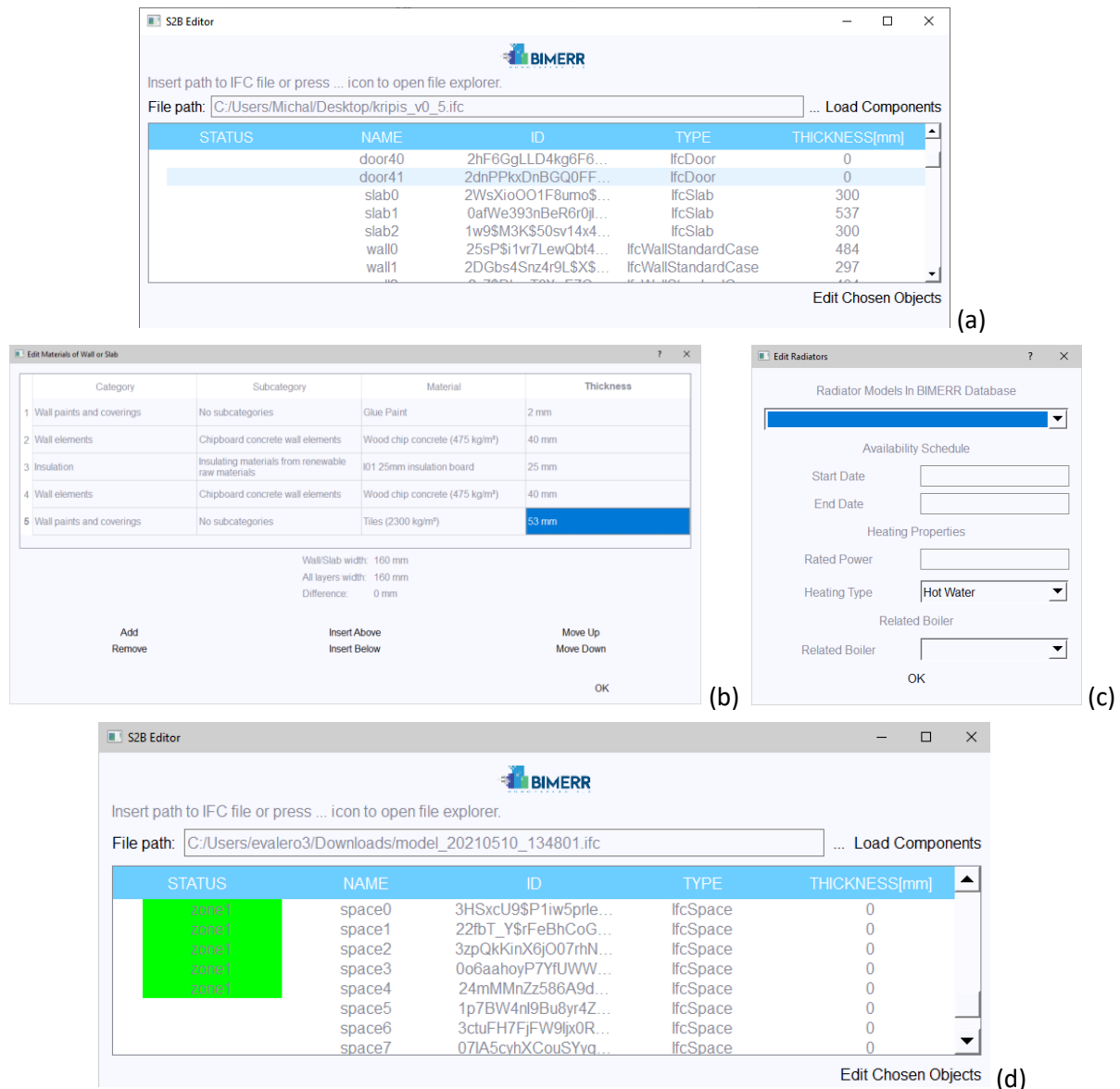


Figure 13: Overview of the Scan-to-BIM Editor process

Whilst this tool was presented as part of Scan-to-BIM Structural in the Deliverable 5.3, it is considered as a sub-component in this document, since additional functionalities have been developed as detailed in the following.

- **Addition of information related to materials:** After loading the IFC file into the software, the tool can populate -through a GUI (see Figure 14(a))- each object with information relative to the layers defining the entity (i.e. walls and slabs). For each layer, thickness and material information are added as illustrated in Figure 14(b). A similar process can be followed to add information about door and window entities. Note that information related to materials should ideally match that of the materials stored in the BIMERR Material Database (see Deliverables D7.1 and D7.2 for further information on the content of the database as well as connection/communication) to enable a smooth integration with RenoDSS. For this, an approach, detailed in Section 6, has been defined.
- **Addition of information related to MEP entities:** Properties of MEP objects, particularly some types of radiators and air conditioning units, can be also added to the IFC file as shown in Figure 14(c). And, importantly, relationships between objects and their parent entity (e.g. boiler) can be established at this point.
- **Classification of spaces into zones:** Another process to be performed manually, due to the variety and complexity of the configuration of spaces (i.e. rooms) in apartments (i.e. zones), is the assignment of spaces to zones. In the Scan-to-BIM Editor tool, spaces can be annotated with existing zones (see Figure 14(d)) or new zones can be created.



**Figure 14: Screenshots of the Scan-to-BIM Editor: (a) Main window; (b) Dialog for defining layers and materials to structural elements; (c) Dialog for assigning properties to MEP objects; (d) Classification of spaces into zones.**

## 5.2 TECHNOLOGY STACK AND IMPLEMENTATION TOOLS

This tool has been written in C++/C# and uses the Qt library for the GUI. It reads BIM models (IFC-STEP format) and subsequently delivers modified IFC files, after the user manually includes additional (i.e. non-visual) information. Libraries used in the Scan-to-BIM Editor are enumerated in Table 5.



**Table 5: Tools and libraries used in Scan-to-BIM Editor.**

| Library         | Version | Licence              |
|-----------------|---------|----------------------|
| Qt              | 5.12.2  | LGPL version 3       |
| Xbim.Essentials | 5.1.297 | CDDL                 |
| Newtonsoft.Json | 12.0.3  | MIT Licence          |
| .NET-core       | 3.1.15  | Apache License 2.0 A |

### 5.3 API/SOFTWARE DOCUMENTATION

This component is a standalone software solution with GUI. The Scan-to-BIM Editor takes an IFC file as a main input and delivers an enriched IFC file, with additional information on materials of structural components and properties of MEP entities. Binaries can be found at <https://datashare.ed.ac.uk/handle/10283/3938>.

### 5.4 ASSUMPTIONS AND RESTRICTIONS

There are several assumptions made in the current version of the tool:

- The program can only edit IFC models in the IFC4/IFC4x1 schema.
- Walls are modelled using *ifcWallStandardCase* entity and if this assumption is violated the program will most likely fail in calculating the real wall thickness. However, this could be modified according to the users' needs.
- *IfcUnitaryEquipment* is used for modelling air conditioning units, and *ifcBoiler* and *ifcSpaceHeater* for heating systems.
- Editor assumes accuracy of up to 1mm and if objects dimensions are specified with higher accuracy these dimensions are rounded up.
- When multiple editions of the model are performed, and some changes are overwritten, this may lead to leaving redundant entities in the model (e.g. dummy material sets).

### 5.5 INSTALLATION INSTRUCTIONS

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

### 5.6 USAGE WALKTHROUGH

The main functionality of this tool is the annotation of IFC files with information on materials and properties of structural and MEP entities. A video illustrating these processes, as well as the classification of spaces into zones, can be found at: <https://bscw.fit.fraunhofer.de/sec/bscw.cgi/d53580035/editProject.mp4>.



## 5.7 LICENSING

The Scan-to-BIM Editor 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>).

## 6 SCAN-TO-BIM INTERFACE

### 6.1 OVERVIEW

As presented in previous sections, three sub-components or tools have been developed to semi-automatically produce BIM models using point clouds and photographs as inputs. These were initially designed as independent pieces of software without connection between them. However, as illustrated in Figure 1, the three tools define the overall Scan-to-BIM process: the IFC model containing the structural entities, which is the output of the Scan-to-BIM Structural sub-component, is an input to the *MEP sub-component*; and the IFC file obtained after adding the MEP objects will be modified by the *Editor*, by adding the supplementary information. Therefore, an umbrella component is required to merge the Scan-to-BIM sub-components and facilitate the users' interaction with the Scan-to-BIM Tool. Additionally, communication with a database to obtain information on construction material and MEP properties is required, as well as a connection to the BIF to upload the obtained IFC file and make it available to the rest of BIMERR tools.

The functionalities provided by the Scan-to-BIM Interface are the following:

- **Connect to Material/MEP database:** Information related to materials must match that of the materials stored in the BIMERR Material Database (see Deliverables D7.1 and D7.2) to enable a smooth integration with RenoDSS. Therefore, material name and ID are retrieved by the Scan-to-BIM Interface from a database through an API. With the database name and ID assigned to objects in the edited model, the RenoDSS Tool can easily import corresponding thermal properties also contained in their database to perform energy simulations.
- **Run Scan-to-BIM Structural + Scan-to-BIM MEP sub-components:** The Scan-to-BIM Interface can execute the algorithms leading to the generation of a BIM model, using a point cloud and photographs of the building as inputs.
- **Run Scan-to-BIM Editor:** This tool completes the IFC file with information that cannot be extracted by means of techniques related to computer vision. This includes (1) the addition of information related to materials, (2) the addition of information related to MEP entities, and (3) the classification of spaces into zones.
- **Upload the IFC file to the BIF:** Once the required additional information has been added to the IFC file, the Editor connects to the BIF and uploads the final version of the BIM model.

### 6.2 TECHNOLOGY STACK AND IMPLEMENTATION TOOLS

This tool has been written in C++ and uses the Qt library for the GUI. Importantly, it uses OpenSSL, as a complement to Qt, for secure connections to third party tools. Libraries used in the Scan-to-BIM Interface are enumerated in Table 6.

**Table 6: Tools and libraries used in Scan-to-BIM Interface.**

| Library | Version | Licence            |
|---------|---------|--------------------|
| Qt      | 5.15.2  | LGPL version 3     |
| OpenSSL | 1.1.1   | Apache License 2.0 |

### 6.3 API/SOFTWARE DOCUMENTATION

As mentioned in Section 6.1, the Scan-to-BIM Interface is connected to a database containing material and properties information in order to deliver BIM models compatible with RenoDSS. The tool connects to the database using an API. For a proper communication with the database, the Interface requires an access key which is built in the code. Additionally, the IP of the machine running the Interface needs to be registered by the database owners.

Once the IFC has been completed, it can be submitted to the BIF. For this, the user must provide, using the dialog of the Interface, the URL to the project, the access token specific for the Data Collection Job set up in BIF, and the Project ID delivered by Keycloak after creating and configuring the BIMERR project (see Deliverable 8.4 for further information on this).

In addition to the mentioned processes, the Scan-to-BIM Interface can call Scan-to-BIM Structural + MEP sub-components to produce IFC files from point clouds and photographs. The executables for these tools need to be stored locally, in the folder where the executable for the Interface is located. Binaries can be found at <https://datashare.ed.ac.uk/handle/10283/3938>.

### 6.4 ASSUMPTIONS AND RESTRICTIONS

The Scan-to-BIM Interface connects to the materials/components database and create a local copy when the “Update BIMERR database” button is pressed (see Figure 15 (a)). The date of the last update is shown in the main window. It is assumed that updating the local copy of the database every few days is frequent enough for the software to work correctly.

### 6.5 INSTALLATION INSTRUCTIONS

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

## 6.6 USAGE WALKTHROUGH

Following the order of the Scan-to-BIM pipeline, illustrated in Figure 1, the first process required to produce an IFC model is Scan-to-BIM Structural (+MEP). As previously mentioned, these processes can be run from the Interface (*interface.exe*), as shown in Figure 15 (a).

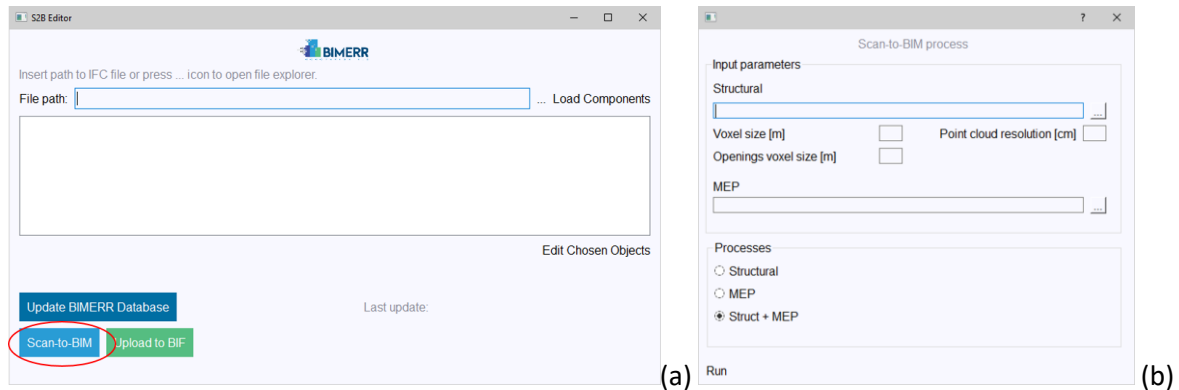
Once “Scan-to-BIM” button is pressed in the Interface main window, a dialog is shown (Figure 15 (b)). Three different processes can be executed from this dialog:

- **MEP**, which can run *findScale* and *detectMEP* in order to produce a pseudo-MEP file as mentioned in Section 4.6.2.
- **Structural**, to generate an IFC file containing the structural components of the building. For this process, the path to the initial point cloud is needed, as well as other parameters required for the analysis of the point cloud (e.g. voxel size and resolution).
- **Struct+MEP**, to produce a BIM model including both structural elements and MEP objects. In this case, in addition to the input arguments mentioned in the previous process, the path to the pseudo-MEP file is required.

After including the parameters, in both Structural or Struct+MEP, the BIM generation process is “Run” and an IFC file is generated. Note that this process can take a considerable time, depending on the size of the point cloud and the configuration (i.e. size of walls and openings and occlusions) of the building. For example, in the case of Kripis house, the generation of an IFC file containing the structural components of the building was performed in 59’, using a machine with an Intel Core i7-8665U processor, and 32GB of installed RAM. The whole process can be divided in sub-processes as detailed in Table 7.

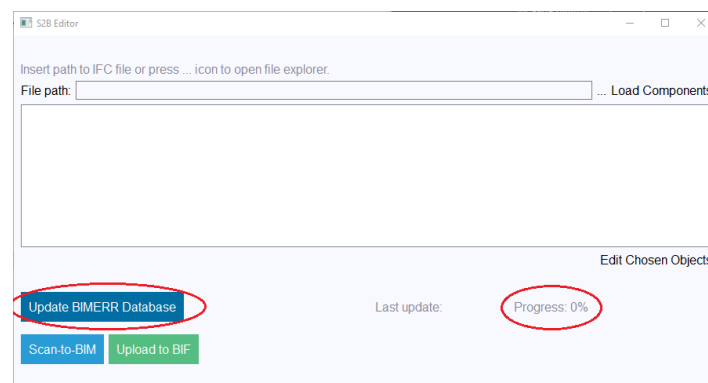
Table 7: Processing times for the generation of a BIM model of Kripis House

| Sub-process                                  | Time [min] |
|--|------------|
| Loading point cloud [16M points]             | 0:04       |
| Segmentation of cloud in storeys             | 0:01       |
| Extraction of faces and generation of spaces | 2:03       |
| Definition of walls                          | <0:01      |
| Definition of 1LSBs for spaces               | <0:01      |
| Definition of 2LSBs for walls                | 0:02       |
| Definition of slabs                          | 0:01       |
| Definition of 2LSBs for slabs                | 0:05       |
| Assigning points per wall                    | 0:08       |
| Opening detection                            | 57:11      |
| Generation of IFC file                       | 0:02       |



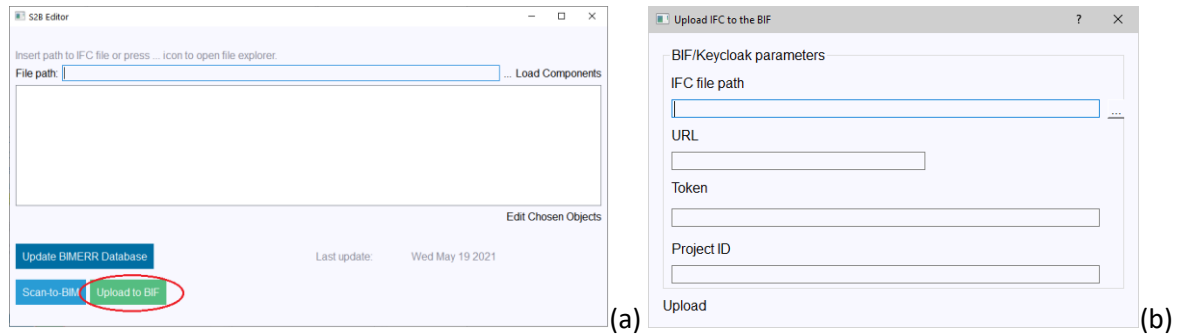
**Figure 15: Steps to produce an IFC model from the Scan-to-BIM Interface: (a) “Scan-to-BIM” is pressed in the Interface; (b) parameters are input for both Structural and MEP processes.**

Then, and before modifying the IFC model with material and properties, the information from the RenoDSS database is acquired/updated, by clicking on the button “Update BIMERR Database”, as illustrated in Figure 16. A message showing the progress of the downloading process will be shown on the GUI, as well as the date of the latest update.



**Figure 16: Steps for updating the local files containing the information of the BIMERR materials/properties database.**

Finally, once the IFC file is complete and includes all the information relative to materials, properties and zones, the user can upload the final version of the model to the BIF. As illustrated in Figure 17, the user must first press on the “Upload to BIF” button in the main window, and subsequently provide the requested information about URL, access token specific to the project Data Collection Job, as set up in BIF and the Project ID obtained from Keycloak after creating the project (see Deliverable 8.4).



**Figure 17: Steps for uploading an IFC file to the BIF: (a) Click on “Upload to BIF” in the main window; and (b) input the required parameters.**

## 6.7 LICENSING

The Scan-to-BIM Interface 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>).

## 7 CONCLUSION

The BIMERR Scan-to-BIM Tool is to be employed at the outset of the BIMERR renovation process. The tool uses as input digital documentation data of a residential building, that includes both laser scanning and photography (with significant overlap to support photogrammetric reconstruction). The tool then aims to almost fully-automatically produce a BIM model of the building, that includes structural components (walls, floors, openings) and MEP components (e.g. HVAC, sockets, and switches). The BIM model is outputted by the tool in IFC-SPF format and contains information that is required by the RenoDSS tool primarily, as well as the ARIBFA tool.

As documented in this deliverable (D5.4), the core functionalities of the BIMERR Scan-to-BIM Tool are delivered by 3 main sub-components, namely the *Scan-to-BIM Structural sub-component* to create the structural BIM model from the point cloud data; the *Scan-to-BIM MEP sub-component* to enhance the previous BIM model with MEP components; and the *Scan-to-BIM Editor*, which can (1) modify the generated BIM model by adding information about materials (to structural components) and properties (to MEP objects) as well as (2) cluster the spaces into zones. These three tools have been integrated into the Scan-to-BIM Interface, which facilitates the interaction of users with the tool and provides communication with the material/components database and the BIMERR Interoperability Framework to upload the generated BIM models. The Scan-to-BIM Tool has been tested with data from the Kripis House, one of the pre-validation sites.

## 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 12<sup>th</sup> 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. In 31. 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., Dollar, P. and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. arXiv-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 (2020). 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. 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.