MoPAD: An Autonomous Robotic Platform for Automatic Extraction of Detailed Semantic Models of Buildings

A. Adán, B. Quintana, S. A. Prieto
3D Visual Computing and Robotics Lab. Universidad de Castilla-La Mancha.
Ciudad Real, Spain
{Antonio.Adan, Blanca.Quintana, Samuel.Prieto}@uclm.es

F. Bosché
Institute for Infrastructure and Environment, University of Edinburgh
Edinburgh, U.K.
f.bosche@ed.ac.uk

Abstract— This paper presents an integrated system that automatically provides detailed as-is semantic 3D models of buildings. The system is able to explore and reconstruct large scenes at a high level of detail, passing through five semantic levels, finally generating a detailed semantic model of the building. Our autonomous scanning platform collects raw data regarding the scene. At the first level of modelling, our autonomous scanning platform collects data regarding the scene and generates a point cloud that is later structured in a semantic point cloud model containing indoor, clutter and outlier point clouds. The second and third levels of semantic models consist of a simple B-rep representation and a model of basic building components, which includes the walls, ceiling, floor and columns, as well as their topology. Openings are then added, thus yielding our fourth semantic model. Finally, small components in buildings, such as sockets, switches, lights and others are recognised, resulting in the fifth semantic model. This approach has been tested on real data of building floors using our Mobile Platform for Autonomous Digitization (MoPAD). To the authors' knowledge this is the first work that, after obtaining 3D data with an autonomous mobile scanning platform, achieves such detailed modelling of building interiors. The performance of the method has been assessed quantitatively against ground truth on simulated and real environments. Two videos are available at the supplementary material of this paper.

Keywords. *Robotic platform, mobile mapping, 3D data processing, semantic modelling, building information model, scan-to-BIM.*

1 Previous work and contributions

1.1 *Semantic as-is building modelling*

Building Information Modelling (BIM) is the latest cutting-edge modelling technology and process in Architecture, Engineering & Construction. Within this new realm of BIM-based project delivery, the generation of precise as-is Building Information Models (BIMs) capturing the current state of facilities is useful to architects and engineers to design renovations and refurbishments, compare them with either as-designed models during construction for quality control, or compare them with prior as-is models during facility operation for asset monitoring.

A large portion of the effort put into the generation of such models, at least from a geometric and semantic point of view, has to date required significant human assistance, which comes with time-consuming human interactions and the risk of human errors.

As detailed in this section, many researchers have been working on the automatic creation of as-is BIM models with various levels of semantic content (references [1-54]). The degree of automation and detail in the methods proposed to date ranges from the automatic acquisition of data capturing the building's as-is state (typically in the form of 3D point coordinates, with additional colour, temperature, etc.) to the automatic detection and positioning of structural as well as small building components (e.g. switches).

Based on our experience, we propose to classify the modelling tasks in five consecutive levels, each of which yielding a model with increasing semantic detail. Figure 1 shows the semantic levels and the objects extracted at each level. Each level is detailed in the below.

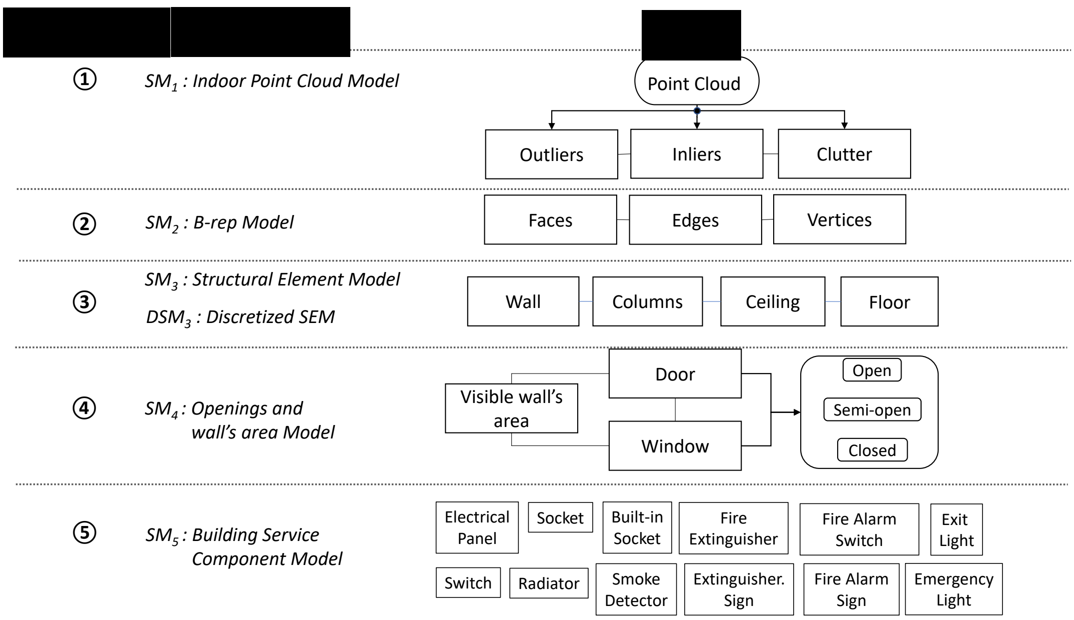


Figure 1. Levels of Semantic 3D modelling, along with objects extracted using our method.

(1) **Semantic Level 1: Automatic data acquisition of the building's as-is state.** The majority of mobile scanning platforms are commanded or guided by humans. In general, commanded mobile platforms (e.g. sensorised cars [1]) that perform permanent scans and digitise the environment cannot be considered as autonomous

systems. Completely autonomous systems are those that are able to perform navigation and 3D data acquisition without any initial knowledge of the scene and without human interaction. Such systems may use prior information to navigate an indoor or outdoor environment (e.g., service robots with information about tasks to be performed; UAVs with programmed flight paths for inspection or monitoring, etc.). During this stage, scan planning and next best scan (NBS) algorithms have to be tackled. Autonomous mobile scanning platforms, such as those in [2]–[8], aim to collect sufficient information and roughly represent building indoors [2]–[5], [7], [8] or building fabric [6]. At this level, the environment is 'modelled' with a 3D, often coloured point cloud, that however lacks semantic information. Recent proposals of such autonomous platforms can be found in [9] and [10] with ground robots (GR), [11] and [12] with unmanned aerial vehicles (UAV), and [13] with a combination of GRs and micro aerial vehicles (MVA).

(2) **Semantic Level 2: Simple geometric building model.** The goal here is to provide a simple geometric representation of the large unorganised point-cloud. This representation is usually created by means of a graph-structure that relates geometric primitives (i.e. vertex, edge, face, etc), all forming a B-rep representation. Examples at this level can be found in [14]–[16]. This is obviously a simplified polyhedral model of the building that still does not contain valuable semantic information.

(3) **Semantic Level 3: Recognition and labelling of primary Structural Elements (SEs) of the building.** This is a higher step in which a semantic meaning of the earlier simplified model is introduced. Some authors present 3D models in which the objects "wall", "ceiling", "floor" and "column" are semantically modelled ([17]–[26]). These objects are sometimes classified by means of context-based algorithms [23]. More detailed semantic models can be achieved by voxelising the 3D space and labelling each voxel, thus generating a discretised semantic model (DSM_3). Examples of such semantic voxel models can be found in [6], [27], [28].

(4) **Semantic Level 4: Recognition of openings within SEs of the building.** We refer here to the recognition of windows and doors ([18], [23], [25], [26], [29]–[44]), and details within SEs, such as skirting-boards, baseboards and moldings ([45]). Some approaches exclusively recognise open ([18], [29]–[37], [41]) or closed ([19], [29], [30], [33], [38]–[40], [42], [44], [46]) doors, and very few identify semi-closed doors [39]. Other methods detect concrete and tilted areas in images of walls during the construction process [47], or deal with pipes and other secondary objects in industrial facilities using 3D imaging technologies ([48], [49]).

(5) **Semantic Level 5: Recognition of small building service components (BSCs) on SEs.** BSCs refer to immovable objects on SEs, such as electrical components or indoor building signs. Very few methods achieve this level of detail in 3D semantic building models. Partial solutions that recognise luminaries ([50], [51]) or sockets ([52], [53]) have been published, with most of them framed in robot interaction applications. Bonanni et al. [54] identify and model a set of usual small components, such as electrical outlets, fire extinguishers, hydrant boxes and printers, on the walls of buildings using a human-robot collaboration approach.

Many approaches address the issue of recognition and positioning of movable furniture objects. In this case, the objective is their detection and modelling within the interior environment ([26], [55]–[59]). These objects, however, cannot be strictly considered part of the building and could be omitted in a 3D structural model.

1.2 *Contributions*

Figure 2 presents a graph showing the levels covered by each of the techniques referred to above, including ours. Each coloured oval represents one of the levels as defined in Figure 1. Note, in order to avoid unnecessary repetitions of the same reference, references that cover several levels in Figure 2 have been referred to in Section 1.1 just for the most significant level. Therefore, there is not a total correspondence between the works explicitly referred per level in Section 1.1. and those inside the ovals in Figure 2.

It is noteworthy that most of the approaches generate models with semantic detail contributing to one or two levels only, with just two of them covering levels 1 to 4. None of the methods that detect building service components (level 5) cover all levels. In this respect, we can state that our system uniquely integrates: (1) fully automated exploration and goal-driven data acquisition and (2) fully automated semantic modelling up to level 5.

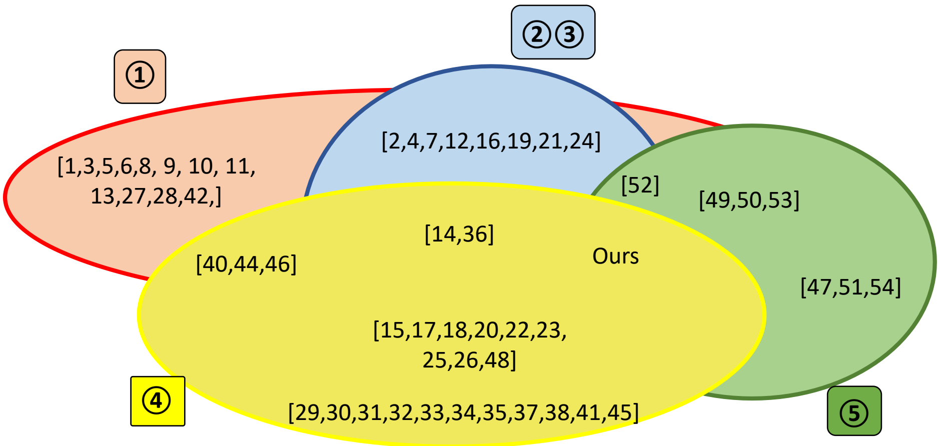


Figure 2. Diagram showing the levels reached by the reviewed approaches and ours.

Table 1 provides further details comparing the autonomous mobile mapping and modelling platforms published in the last five years that are used for digitization of buildings and are closely related to ours. The table also includes our system, MoPAD. Since MoPAD is an autonomous platform, commanded platforms are not considered in Table 1. The table reports: the publication year, the environment in which the platform works, the sensors used for digitization, the type of vehicle (V), the covered semantic models (*SM1* to *SM5*) and the output provided by the platform. It is noteworthy that most of the systems generate models that cover the first or second semantic levels, but none of them, except ours, tackles further modelling, including the modelling of BSCs.

In this paper we present an integrated system that autonomously explores and scans building interiors, and additionally produces detailed *SM5* models of those environments. The algorithms that address different steps of this system have already been published ([60], [61], [62], [63]). This article focuses on demonstrating the applicability of MoPAD in its entirety in real scenarios and discusses the limitations/disadvantages of the system. Section 2 is devoted to showing the navigation of the robot in a floor with multiple rooms. Section 3 summaries the different components of the system. Section 4 provides a detailed information of the experimentation, results and evaluation on representative cases study. Finally, Section 5 and Section 6 present, respectively, the conclusions and future work.

Table 1. Comparison with autonomous scanning platforms published in the last five years (V=Vehicle, GR= Ground Robot, MAV= Micro Air Vehicle, UAV= Unmanned Aerial Vehicle, LRF= Laser Range Finder, DSLR (Digital Single Lens Reflex)

Platform/Ref.	Year	Environment	Sensors	V	SM1	SM2	SM3	SM4	SM5	Output
Irma 3D/[4]	2014	Corridor and several rooms	3D laser scanner, thermal camera and RGB camera	GR	√	√				Point cloud and mesh model with thermal images.
PR2/ [9]	2014	A table top, two adjacent rooms and corridor with rooms	2D LRF and RGB-D camera	GR	√					3D point cloud
-/[7]	2015	Long corridor	RGB-D sensor	GR	√	√				3D point cloud
AscTec MAV /[11]	2016	Indoors and outdoors	Stereo camera	MAV	√					3D voxel model
CPS-VII/[13]	2017	Indoors and outdoors	3D laser scanner	GRs & MVA	√					3D point cloud
UAV I/[12]	2017	Indoors	Rotating laser module	UAV	√	√				3D voxel model and raw point cloud
GRoMI/[10]	2018	Scans taken in corridors and walkways	Two 2D LRF and DSLR camera	GR	√					3D point cloud
MoPAD (Ours)	2019	Indoors. Adjacent rooms and corridors	3D laser scanner and 2D LRF	GR	√	√	√	√	√	Detailed 3D model

2 Navigation of the robot in a floor with multiple rooms

MoPAD is able to navigate and scan a floor with several interconnected rooms. The overall navigation algorithm has been implemented by using a Dynamic Traversal Tree (DTT), in which each room is represented by a node. Parent nodes are connected with their respective child nodes as MoPAD moves from one room to another. Since MoPAD is designed to function without any prior information of the building and its layout, the tree starts empty and is expanded as new rooms are visited.

The navigation algorithm is summarised in Figure 3. The robot starts scanning the first room, which is the root node. Once the room has been scanned, the open doors of the current room are detected. The tree is then updated by adding to the current parent node the rooms corresponding to these open doors as child nodes. If there is any child node (i.e. room) that has not been visited yet, the robot goes to the unseen room passing through the nearest door and starts the scanning process again. Otherwise, the robot returns to the parent node. If that parent node is the root node and no more open door needs to be traversed, then the process ends.

Figure 4 (a) shows the scenario and the path followed by the robot to visit all the rooms. Figure 4 (b) illustrates the DTT' evolution. The robot scans room R1 and detects four open doors, which connect respectively with rooms R2, R3, R4 and R5. DTT is updated by adding the corresponding four child nodes. The robot then finds the nearest open door ($d_{1,2}$) and passes through it, entering room R2. The robot scans R2 and detects only one open door ($d_{2,6}$) connecting the unvisited room R6. After scanning R6, the method does not find any open door that connects an unseen room. Consequently, the robot backs to the parent node (R2). In R2 the same thing happens and the robot heads to R1. Now, from R1, the robot decides to enter and scans room R3 and the algorithm continues. This process is repeated until all rooms have been scanned.

Section 3 will now review the actions and data processes that MoPAD follows to generate the aforementioned semantic models of a single room.

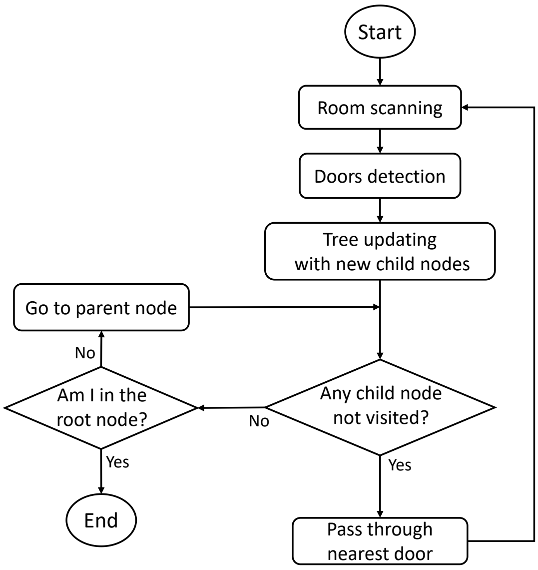
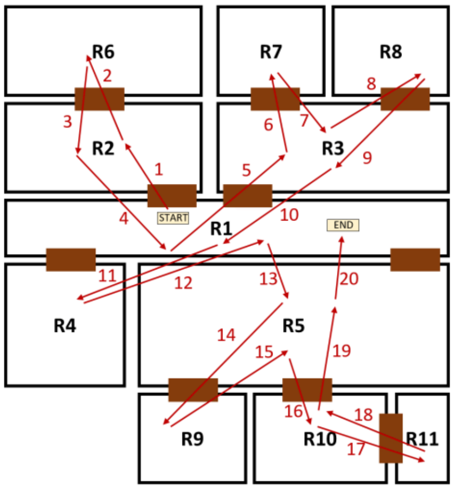
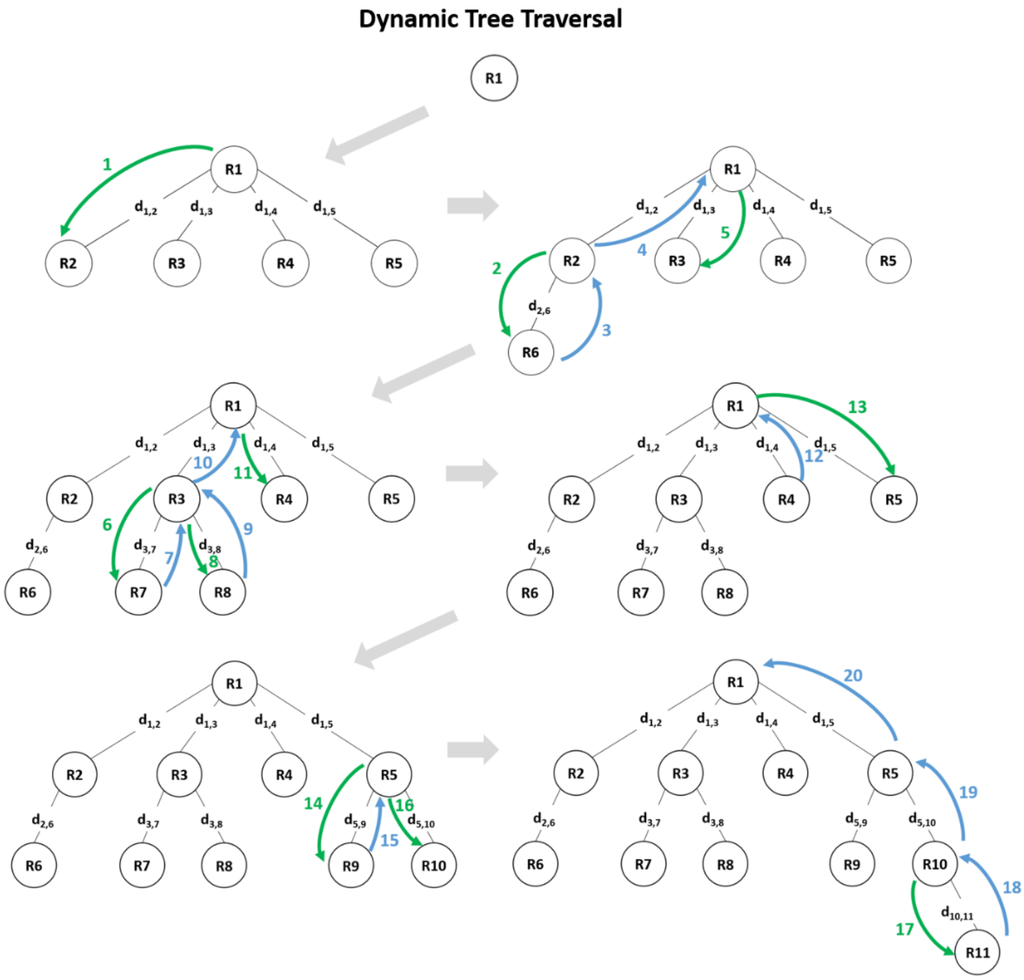


Figure 3. Flowchart of the MoPAD navigation algorithm.



a)



b)

Figure 4.a) Example of a building floor with multiple rooms and doors. The arrows indicate the path of the robot when passing through an open door. b) Evolution of the Dynamic Tree Traversal as MoPAD moves from one room to another. Forwards and backwards movements are represented by green and blue colours.

3 Progressive semantic modelling: from automated acquisition of 3D data to automated generation of semantically-rich as-is building model

3.1 *Automatic collection of complete point clouds: SM_1*

MoPAD is able to collect coloured point clouds of large environments. This is a sensorised mobile robot equipped with a long-range laser scanner, an RGB camera and two laser range finders for navigation. MoPAD moves autonomously from one position to another, which is calculated by a next best scan (NBS) algorithm.

The majority of the current mapping approaches accumulate as much data as possible and scan everything that lies inside ([27], [9]) or outside ([6], [64]) the building, no matter the meaning of the data. However, since the MoPAD's objective is to produce the model of a building, the NBS algorithm is based on collecting as much data as possible regarding Structural Elements (SEs), i.e. walls, ceiling, floor and columns.

Before taking a new scan, the region of interest (RoI) encompassing the scene is defined as the polyhedron that contains the accumulated point cloud (scans registered up to this time). This RoI is updated each time a new scan is taken. Thus, by not hypothesizing a

RoI, our approach contrasts with earlier NBS strategies that assumed RoIs *a priori* (bounding boxes or convex-hulls) ([27], [9], [65]). This strategy makes the approach more versatile for arbitrary-shape scenarios.

Figure 5 presents a flowchart that synthesizes the data acquisition stage. Let us assume that MoPAD has scanned a certain room A from n positions, leading to the accumulated point cloud $P_n(A)$. MoPAD first checks whether the scanning stopping criteria are satisfied. MoPAD stops scanning a room when at least one of thresholds ε_1 or ε_2 are overtaken, where ε_1 is the minimum percentage of SE that must be sensed (fixed at ε_1=90%), and ε_2 is the minimum increment of SE area with respect to the last scan (ε_2=1%).

The whole algorithm can be synthesized in the following points:

- If the stopping criteria are not satisfied, MoPAD moves to a new position, stops and scans. The main steps are:

(1) Calculation of NBS. The NBS location is calculated as the location inside the current discretised RoI from where the likelihood of collecting new visible SE regions is maximum. A raytracing process is carried out from each accessible MoPAD's location, within the RoI, and the next location is selected as the one providing the maximum likelihood of increase in visible SE area. The main steps of the NBS algorithm are:

a. Calculate the list of hypothetic next sensor positions that satisfy the security and accessibility requirements in the current room.
b. Calculate the *SE membership probability* of each voxel of the current discretised RoI, which is the probability of the voxel being located in a SE.
c. Calculate the *Certainty* of each SE from each hypothetic next sensor position, which is the sum of the *SE membership probabilities* of the voxels of the SE.
d. Calculate the *Certainty of the RoI* from each hypothetic next sensor position.
e. Find the NBS as the position that maximizes the *Certainty of the RoI*.

More information and details of the NBS algorithm can be found in [61].

(2) The robot moves towards the NBS location. To be able to autonomously navigate to the NBS location, the robot needs to be aware of its surroundings and locate itself within them. To do that, the robot obtains an obstacle map from the previously acquired point cloud, and matches it with the readings of a 2D laser range finder, by means of an Adaptative Monte-Carlo Localization algorithm (AMCL).

Once the robot knows its position on the map, the path planning algorithm (i.e. Navfn global planner running on ROS) generates the optimal path to the desired goal. The local planner, a Time Elastic Band (TEB) approach, computes all the velocity commands which are sent to the robot. The robot can thus precisely follow the calculated path and avoid obstacles not present on the original obstacle map. See [60] for further details on the navigation module of MoPAD.

(3) MoPAD takes a 360-scan and aligns the collected data, s(t), within the accumulated point cloud, S(t-1). The two main sub-processes here are: inliers/outliers detection and point cloud registration.

Inliers mean data which belong to the interior of the room, whereas outliers are points captured by the laser scanner that fall outside the room. These points appear owing to open doors and windowpanes. Inliers are detected as the points that lay within the perimeter defined by the points of the ceiling in a 2D plan view. Outliers are the points that fall outside that perimeter.

The registration stage is carried out in three steps. The set s(t) is firstly horizontally aligned according to the inclinometer of the 3D laser scanner. Secondly, the 2D (plan view) registration transformation matrix between s(t) and S(t-1) is estimated using the odometry of the mobile robot. Finally, this transformation is refined by applying the well-known Iterative Closest Point (ICP) technique. The accumulated point cloud, S(t), is then calculated.

(4) The RoI polyhedron is updated and its faces are classified into segments that contain or do not contain SE data.

(5) Discretisation of the space and labelling. The 3D space limited by the RoI is discretised in voxels which are labelled according to their visibility (occluded/non-occluded) and the SE condition (SE/non SE). Occupied voxels inside de RoI are labelled as clutter.

(6) Parameters ε_1 and ε_2 are calculated and the stopping criteria are checked again.

(7) If any of the stopping criteria is not verified go to (1).

- If at least one stopping criterion is verified, the system processes the total point cloud collected *P(A)* and obtains the semantic models SM_1 to SM_5 of room A. (see Sections 3.2 to 3.4). MoPAD then moves towards the exit door and takes two scans, one before and one after the doorframe that separates rooms A and B. Both scans are used subsequently to align the total point clouds *P(A)* and *P(B)* and therefore the respective room models. In addition, the second scan is considered to be the first point cloud of room B where the process starts again.

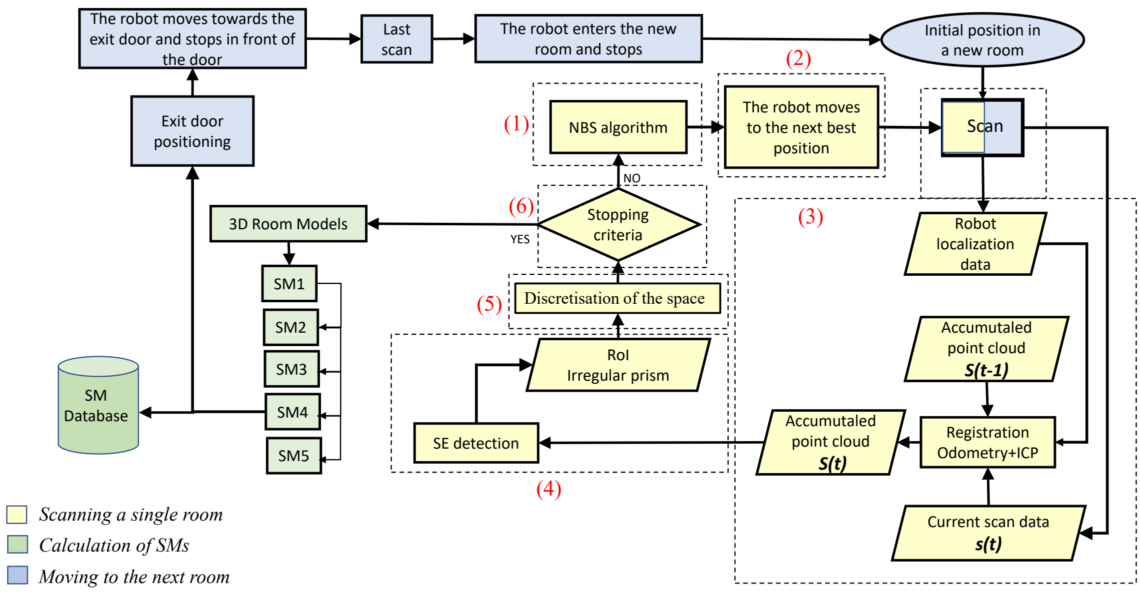


Figure 5. Flowchart of the scanning and modelling approach. Actions for the scanning of a single room are shown in yellow. The ID according to the explanation provided is also introduced. The calculation of the semantic models are in green, and the robot actions to leave the room and start scanning a new room are in blue.

Figure 6 illustrates the main steps from the data acquisition of a single room. In each row, there is a flowchart (left), a top view of the scene with the data in different colours (centre) and a 3D view of the results (right). Colours are used to highlight data points, regions or sub-processes.

The input of the algorithm is the point cloud collected with a 3D scanner. In the first row, the point clouds of the two consecutive scans at times *t-1* and *t* painted in green and blue are registered (step (3)). The row below concerns step (4): Obtaining the RoI. Blue bars delimit the current RoI and magenta bars represent the zones that have been recognised as SE. The 3D view of the RoI with the data points superimposed in magenta is shown (only walls are shown for a better visualization; not ceilings or floors). The creation of the voxel space is in the next row (step (5)). We present a top view of the discretised space and 3D view of the voxels with different labels. Finally, the last row illustrates the next position of the scanner calculated with our NBS algorithm. The output of this process is the SM_1 model of the room.

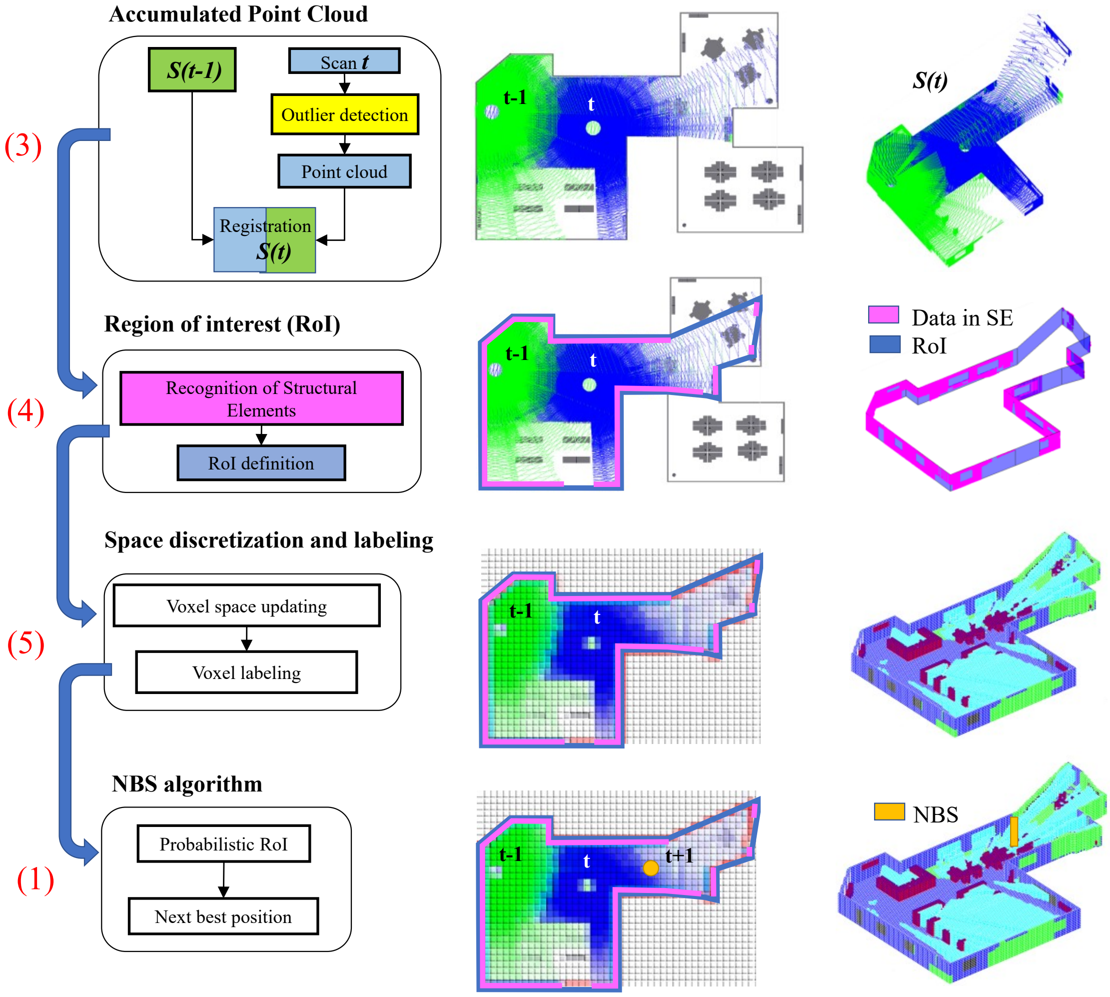
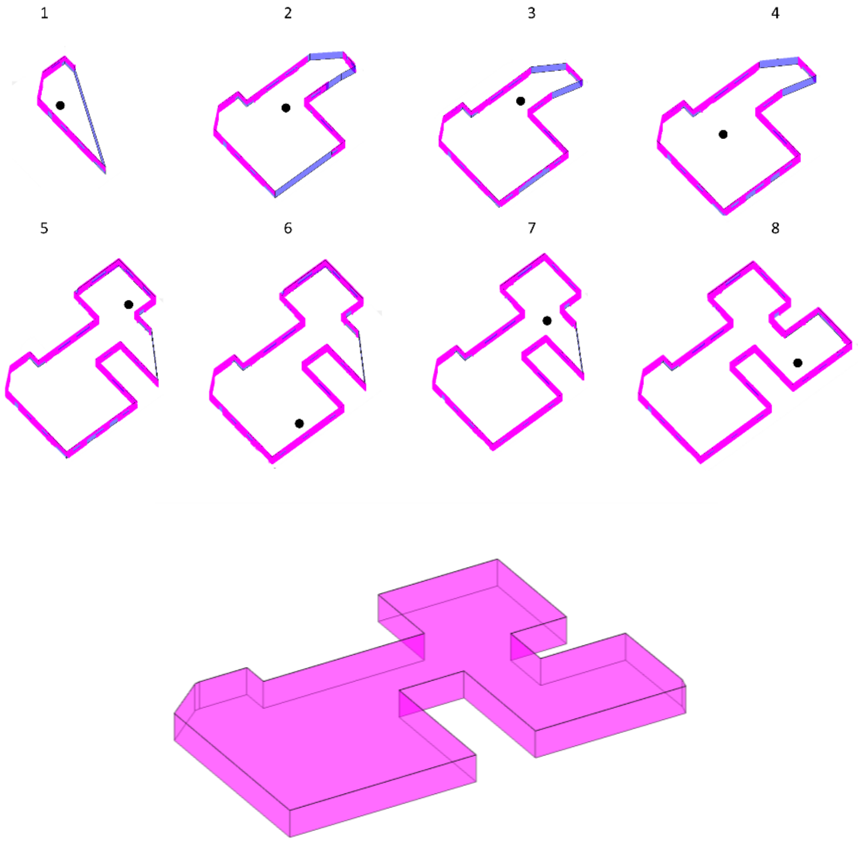


Figure 6. Autonomous data acquisition of a single room.

3.2 *Detection and Modelling of Structural Components: SM2 and SM3/DSM3*

Our scan planning algorithm has the advantage that an irregular polyhedron that encloses the indoor point cloud is updated after each new scan. This polyhedron denominated as RoI above, defines the indoor boundaries and provides by itself a still raw B-rep representation. The last polyhedron obtained from the scanning process is, therefore, our second semantic model SM_2, which is composed of faces, edges and vertices. Figure 7 shows an example of the evolution of this polyhedron until the scene has been completely scanned, along with the final model SM_2.



(a)

(b)

Figure 7. (a) Evolution of the B-rep for eight successive scanner locations (only the wall faces are shown for clarity); and (b) the final model SM_2.

Figure 8(a) shows a flow chart that explains how SM_3 models are obtained as well as an illustrative example. SM_3 models contain information concerning the main SEs of a building (wall, ceiling and floor).

The extraction of the points belonging to the floor and ceiling of the room is carried out first. This is easily done by detecting two maximums in the Z-histogram of the data. We assume here that ceilings and floors are planar and horizontal regions. However, since our modelling process is applied room by room, our approach is at least able to detect ceilings of different heights in different rooms.

The segmentation of the points belonging to each wall is conducted next. The point cloud is first projected onto the XY plane and is then discretised, thus generating a 2D image *I*. The edges of the polygon that encloses the data in *I* are the walls.

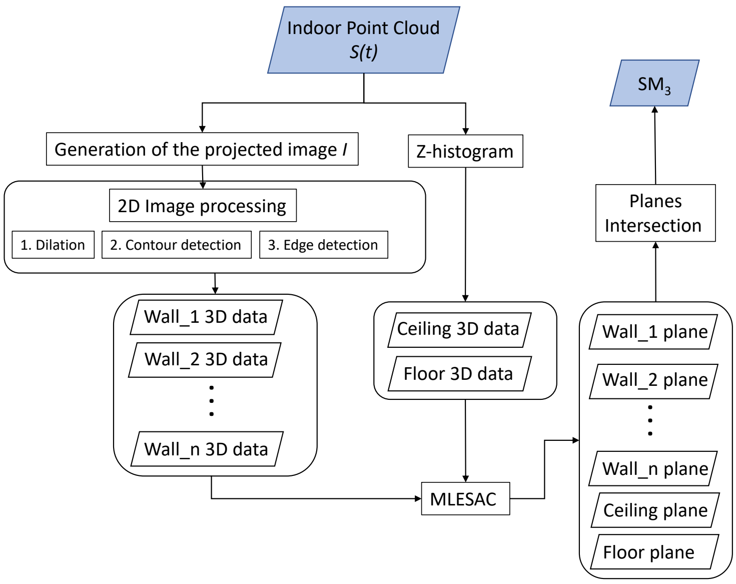
All these data segments (walls, ceiling and floor) are fitted to planes using the MLESAC technique [66], and coherently intersected. For a room A, the SM_3 model is therefore composed of planar patches, labelled semantically ($wall_{1A}$, $wall_{2A}$,…$ceiling_A$, $floor_A$), along with their topology.

The set of points delimited by SM_3 is then used to extract a semantic voxelized model. This is a 3D grid of voxels where a voxel is like a small cube which does or does not contain at least one data point. In this discretised structure, the faces of SM_3 become faces composed of voxels. Besides, the remaining voxels are labelled with a semantic meaning according to its visibility condition. The result of all this is a discretised semantic model DSM_3 composed of labelled voxels as follows.

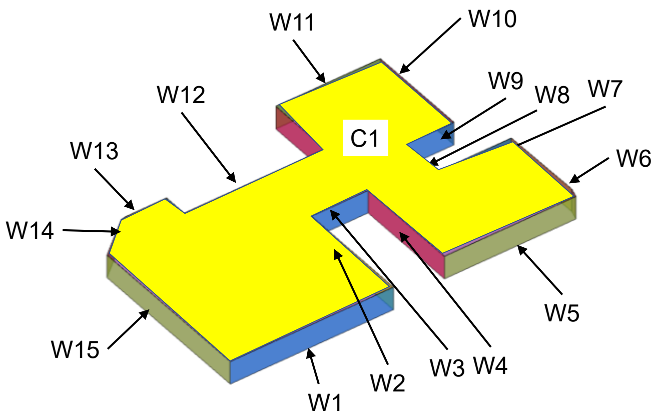
Voxels that are part of SEs are labelled as: (1) *opening*, if they lie in an opening area; (2) *structure*, if they are visible, i.e. contain scanned points, and (3) *occluded-structure*, if they are not visible from any of the scanning locations. The remaining voxels are: (4) *clutter*, if they contain points that do not belong to any SE, (5) *empty* for the seen voxels that do not contain any scanned point and (6) for all the remaining unseen voxels.

Figure 8(b) presents a SM_3 model with 15 walls, 1 ceiling and 1 floor. The corresponding dual discretised semantic model DSM_3 is shown in Figure 8(c). The reader can find complete information of this process in [61].

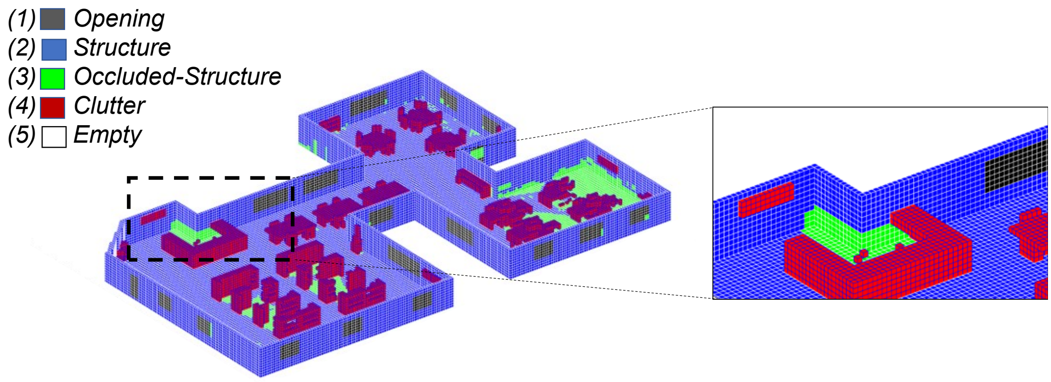
DSM_3 is used to support the identification of the NBS location. Specifically, it is used to calculate the hypothetic visible areas, measured in voxels units, from each accessible MoPAD's location by using a raytracing process.



(a)



(b)



(c)

Figure 8. (a) Process of generating the model SM_3. (b) An example of model SM_3 with the walls (W), floor (F) and ceiling (C) detected. (c) The dual discretised model DSM_3 with all labelled voxels.

3.3 *Detection and Modelling of Openings: SM_4*

Our method detects openings (doors and windows) and then calculates the opening angle λ of each door. The technique is unique in that it integrates the information regarding both the geometry (i.e. XYZ coordinates) and colour (i.e. RGB or HSV) provided by our integrated scanning system. Extended information of this method can be found in [62]. Opening detection is carried out in two stages, using both the labelled voxels for the walls obtained in DSM_3 and their corresponding 4D orthoimages J_{CD}. Each pixel of an orthoimage contains colour (RGB or HSV) and depth information (i.e. orthonormal distance from the 3D points to the wall plane). See Figure 9(a) for examples of wall 4D orthoimages.

The recognition algorithm is divided into two steps: wall area detection and opening detection.

1) Wall area detection. The segmentation of the visible parts of the wall (not occluded by other objects that may be hung on the wall) is conducted by finding clusters of coherent colour seeds on the wall and then carrying out a segmentation by colour. As a result of this process, the visible area of the wall is separated from the rest of the wall and the openings on it are sought in the remaining parts of the wall.

2) Opening detection. The algorithm is based on analysing the discontinuities in the 4D RGB-D space and on the knowledge of the visible area of the wall. We process the colour and depth components of J_{CD} (i.e. J_C and J_D) separately and recombine the results subsequently. Assuming that door and window frames are rectangular, we detect straight lines in J_C and J_D. These lines represent the discontinuities as regards the colour and depth of the wall (if the door has a protruding doorframe, the discontinuity in the depth dimension should result in line detections; if the door is a different colour from that of the wall, the discontinuity in the colour dimensions should also result in line detections). Note that the lines detected may only contain *parts* of the contours of hypothetical doors due to potential occlusions.

All possible rectangles defined by two pairs of horizontal and vertical lines are then tested, and we retain only those rectangles whose size falls within the range of typical opening sizes. This yields a highly reduced set of rectangles $\{r\}$. Finally, each rectangle r is recognised as an actual opening if it fulfils a set of conditions regarding properties of colour and depth consistency, degree of door frame occlusion and location consistency within the wall. We typically classify a door leaf by means of its opening angle λ. Doors are labelled as: *open,* if $90^o \leq \lambda$; *semi-open*, if $\varepsilon \leq \lambda < 90^o$ (where ε=5 º); and *closed* if $0^o \leq \lambda < \varepsilon^o$.

Figure 9 shows the detection of three doors in a wall. In the first example, there are several objects that occlude parts of both doors. The occlusion percentage is around 38%. Additionally, the colour of the wall is not uniform owing to the lower part of the wall being tiled. The second example presents a wooden double-door with a similar colour to the wall and the third example shows the detection of a *semi-open* door.

The recognized openings are ultimately added to SM_3 to make up SM_4. Some examples of integrated models are presented in Section 4 that reports experimental results.

(a) (b)

Figure 9. Examples of opening detection. (a) 4D-Orthoimage of a wall. (b) Rectangles that enclose the detected doors.

3.4 *Detection and Modelling of Small Building Service Components: SM_5*

Beside the principal architectural components included in the earlier models, other secondary but permanent components, such as building service components, located on walls and ceilings can be recognised in the coloured point clouds, and integrated to SM_4, thereby obtaining SM_5.

The main problem to achieve this semantic level, when using the data provided by an autonomous platform, lies in the quality of the data (accuracy and density). Although the angular resolution of our laser scanner provides 10's of millions of points per 360°-scan, few data is acquired from these objects that are often small, and as a consequence. For example, an object "electrical switch" is commonly captured as an 15x15 pixel orthoimage. In addition, the data accumulated in the orthoimage usually come from different robot positions, which blurs and distorts the final image of the object. If the scanning resolution were increased, the quality of the image would increase, but at the same time the large amount of data might collapse the system, making it slow and impractical. This is one of the issues that will be dealt with in future improvements of our system.

With such poor-quality data, local feature -based recognition methods are ineffective. We conducted a set of experiments using SIFT ([67]) and SURF ([68]) algorithms using a database composed of 31 different models of BSCs. The algorithms were applied to 53 orthoimages extracted from coloured point clouds of 5 scenarios. The recognition results were disappointing, with success rates below 5%.

Instead of using local descriptors, we propose a global feature -based strategy that separately processes the geometry (i.e. XYZ coordinates) and colour (i.e. RGB) of each SE, with the aim of finding candidates. The results obtained in the 3D and colour domains are subsequently combined in a consensus phase. The approach is summarised below, but the reader can find complete information in [63].

Our approach assumes a database that contains the models of the objects that could be recognised and located in a building. This database includes the colour and the depth image models of each expected object. We also make the reasonable assumption that these small components lie on SEs.

Each one of the SEs is characterized by a 6D orthoimage J_{CD} , where a pixel contains the position (XYZ) and the colour (RGB) of the SE points projected onto to the SE plane (in our case, points at a distance of 20 cm). After removing the existing openings (if any) in J_{CD}, the new orthoimage $\hat{J}_{CD}$ is decomposed into two sub-images: a depth image $\hat{J}_D$ and a colour image $\hat{J}_C$.

$\hat{J}_D$ is used to detect protruding objects in the depth image of the SE plane. A high-frequency filter is applied to emphasize object boundaries in the image. An image matching algorithm using normalized cross-correlation is then utilised to recognise the objects in the detected protruding areas.

$\hat{J}_C$ is used to detect objects with colour discontinuities. Areas with colour discontinuities are taken as candidates and bounded in small boxes. The candidate boxes are then classified by using a perceptron algorithm. A feature pattern composed of global colour descriptors in HSV and Lab colour spaces is used for the training of the neural network. Since the saturation and the relative colour measures, *a* and *b*, are less sensitive to image distortion and blur, the pattern is based on these colour components. The three images corresponding to components *S*, *a* and *b*, are reduced to three classes. The pattern is composed by twelve values: the respective values of the first and second class, and their relative areas.

Although some objects might be detected by means of both geometry and colour, some others may be recognised in only one of the images $\hat{J}_C$ or $\hat{J}_D$. For each candidate O, a consensus algorithm is implemented by means of its *Recognition Coherence Matrix* $R(O)$, which contains the respective *Recognition Coherence Levels* α of O in the SE. Each entry α in R signifies an instance of the object in the scene and measures the coherence between a detection of O in $\hat{J}_D$ (or none) and in $\hat{J}_C$ (or none). α 's value is in the range [0,1] and is calculated using the formula in Equation (1). α assesses the overlap between two detection bounding boxes B_C^i and B_D^j , obtained in $\hat{J}_C$ and $\hat{J}_D$ respectively and mormalises the overlap over B_{CD}^{ij} , which the bounding box that encloses B_C^i and B_D^j . In Equation (1) $\langle . \rangle$ signifies the number of pixels. The second row of Equation (1) corresponds to the no-overlapping cases. Finally, when only one bounding box is detected, $\alpha = 0.5$.

$$\alpha = \begin{cases} \dfrac{\langle B_C^i \rangle \cup \langle B_D^j \rangle}{\langle B_{CD}^{ij} \rangle} & if\ B_C^i \cap B_D^j \neq \emptyset \\ 0 & if\ B_C^i \cap B_D^j = \emptyset \\ 0.5 & if\ \nexists B_C^i or \nexists B_D^j \end{cases} \tag{1}$$

The *Recognition Coherence Matrix* $R(O)$ is used to identify several instances of the object O inside an SE. To do this, the highest value in $R(O)$, i.e. highest *Recognition Coherence Level*, is selected and is considered to correspond to a recognised instance of O, and the corresponding row and column of R are then removed. This process is iterated until R is null or until the number of selected cells is equal to the expected number of instances of O in the SE.

Figure 10 illustrates this process with an example in which several instances of the same object are or are not detected in both $\hat{J}_D$ and $\hat{J}_C$. The first row in Figure 10 shows the instances recognized in $\hat{J}_D$ and $\hat{J}_C$. In some cases, the respective recognition algorithms yield more than one instance per object. Thus, the object type 1 has four and three instances in $\hat{J}_C$ and $\hat{J}_D$ respectively, and the object type 2 has two instances in both orthoimages. One instance of the object type 3 is found in $\hat{J}_D$, and one instance of the object 4 are recognized in $\hat{J}_C$. The figure then shows the four *Recognition Coherence Matrices* and the *Recognition Coherence Levels* obtained for each instance after consensus. Figure 11 illustrates some details of the detection of BSCs in walls.

Finally, the instances of a recognized object are added to the earlier model SM_4, thus generating the last semantic model SM_5. Further examples of recognition of BSCs are presented in the next section.

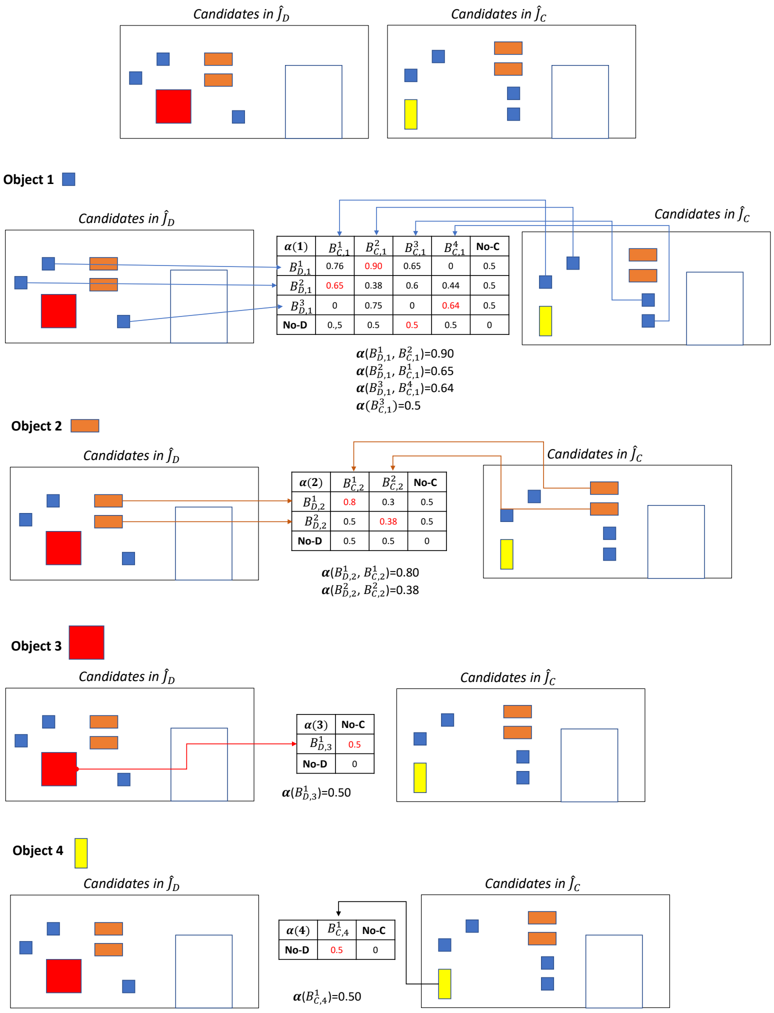


Figure 10. Example of recognition of multiple instances of an object on a wall.

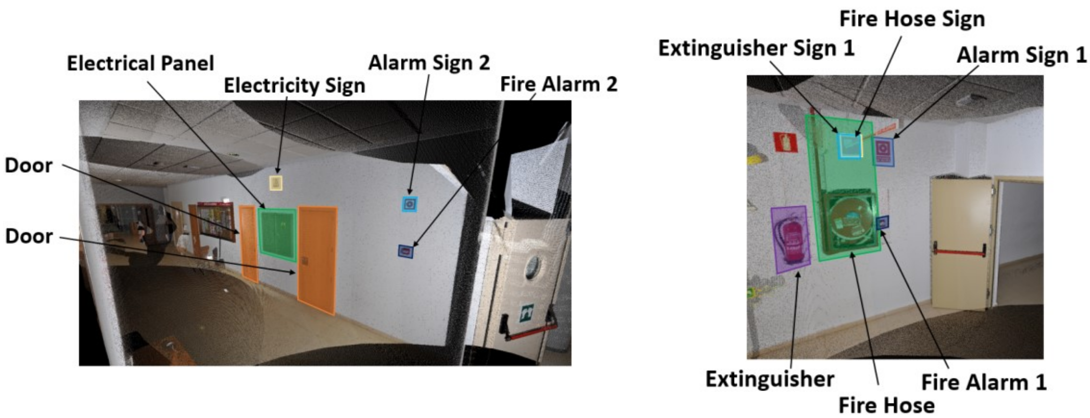


Figure 11. Representative real scenes with the output of the opening and BSCs detection approaches. The positions of the detected doors and the BScs are highlighted in different colours.

4 Experimental Results

4.1 *MoPAD*

Experimentation on real environments has been carried out with our mobile platform in representative cases study. The coloured point clouds are collected with a Riegl VZ400 laser scanner and a RGB-DSLR (Digital Single Lens Reflex) camera on board MoPAD (Figure 12 (a)). The camera and scanner are mounted on a motorized pan-tilt unit that enables the system to acquire full 360° x 180° acquisitions . We note that a thermal camera has also been added to obtain 3D thermal scans, but this additional data is not considered in the work reported here. The platform carries two 2D laser range finders (LRF1 and LRF2). LRF1 is placed close to the VZ400 and its captured data must be aligned to the obstacle map (OM), which has previously been obtained from a slice of the point cloud. Both sets of points must match for successful robot positioning. LRF2 is placed at the base of the robot and localizes the robot while it navigates from one scanning position to the next. Two computers command the Riegl VZ400. The whole robot system calculates the NBS and undertake all 3D processing stages onboard.

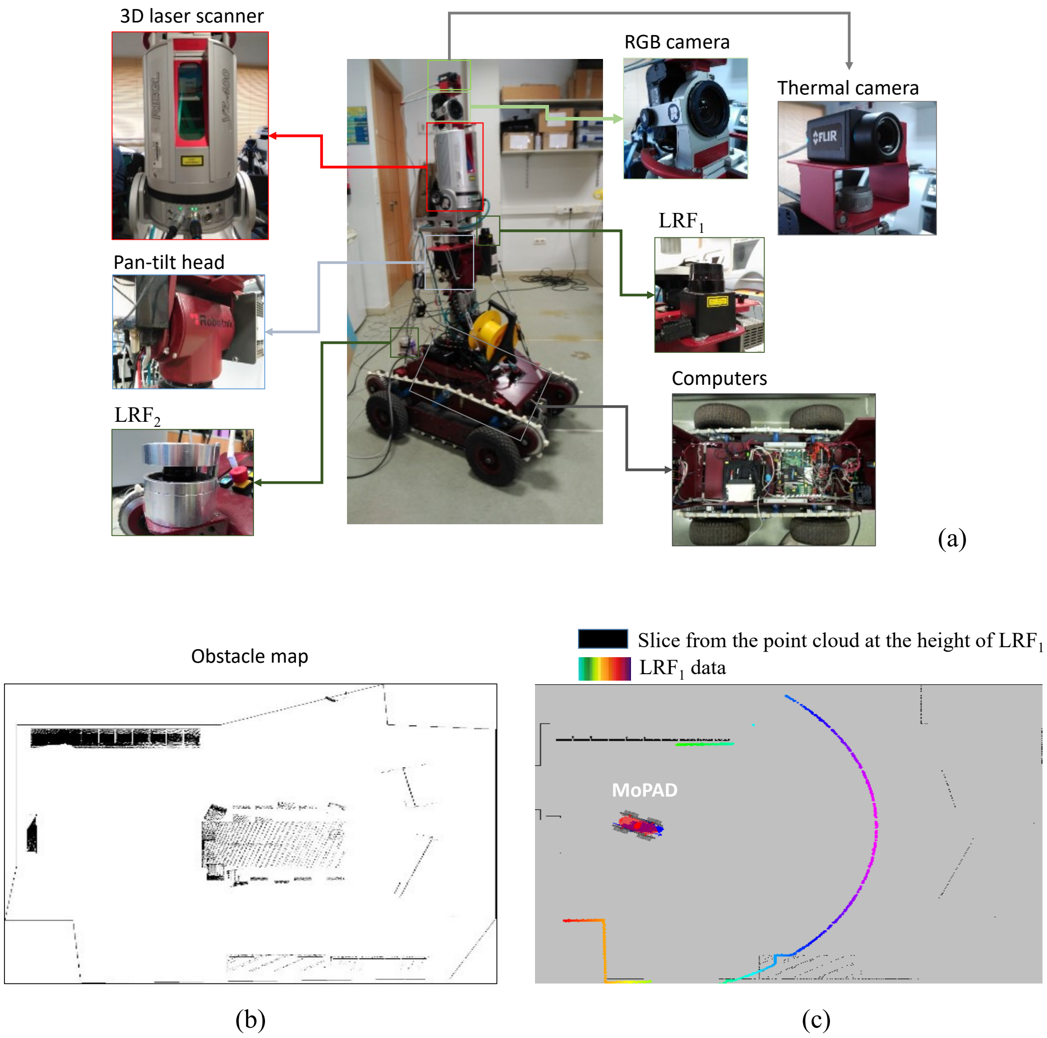


Figure 12. (a) Mobile Platform for Autonomous Digitization (MoPAD). (b) Obstacle map calculated from the point cloud below the scanner. (c) Positioning and tracking of the robot. In black is a slice of points at the height of LRF_1 and in different colours are the data obtained from LRF_1.

4.2 *Experimentation on a real environment*

A part of the basement of the Industrial Engineering School building at Castilla La Mancha University (Spain) was chosen as a good example to test our system on real environments. Figure 13 shows several photos of the scenario tested. The scenario, of 19.5 m x 44 m in size, is composed of 4 rooms with 32 walls, 4 ceilings, 4 floors, 26 columns, 32 doors and 49 small objects. The ground truth 3D model for evaluation of accuracy and errors was produced manually by using a terrestrial laser scanner with manual acquisition and registration processes.

Figure 13. Photos of different rooms of the scenario tested.

- *Level 1*. Following our scan planning algorithm, the floor was scanned from 14 different locations producing a total of 46 million points corresponding to SEs (SM_1 model). The average processing times were: 27.7 sec for registration, 29.2 sec for point cloud pre-processing (inliers/outliers and segmentation of floor and ceiling), 28.1 sec for the calculation of the RoI and 127.8 sec for segmentation, labelling and NBS algorithm. Figure 14 (a) shows shots of the MoPAD at the scanning positions and the current obstacle maps. The scan positions and the corresponding aligned point clouds are shown underneath. A video showing the whole process is available as supplementary material.

- *Levels 2 and 3*. After finishing the data acquisition stage, the simplified 3D model is a polyhedron with 38 patches, 90 edges and 60 vertices organised in a relational graph structure (SM_2). In level 3, the patches are labelled as walls, ceilings, floors or columns, thus extending the semantic content of the 3D model (SM_3). We found 38 SEs (30 walls, 4 floors and 4 ceilings) and 25 columns. The recognition rates (true positive cases) were: SEs 95% and columns 96%. The occlusion percentage was 9.4 %. Figure 14(b) left illustrates the model SM_3.

  The accuracy of the SEs detected in SM_3 was obtained by comparing our simplified 3D model with the ground truth model. This comparison was made by measuring the difference between the length of the vertical and horizontal edges in our model and those in the ground truth model (denoted as d_v and d_h). An orientation error angle Φ was also calculated for each patch of the model. Figure 15(a) illustrates these dimensions. The average values for the differences between the vertical and the horizontal edges were $d_v = 2.8$ cm and $d_h = 2.3$ cm, with maximum values of $d_{v\_max} = 4.5$ cm and $d_{h\_max} = 11$ cm. The average and maximum values of the orientation error were $\Phi = 0.17º$ and $\Phi_{max} = 1.55º$, respectively.

- *Level 4*. The method detected 28 doors (3 open and 25 closed doors) and the recognition rates (true positive cases) for doors was 87.5%. The right of Figure 14(b) shows the model SM_4.

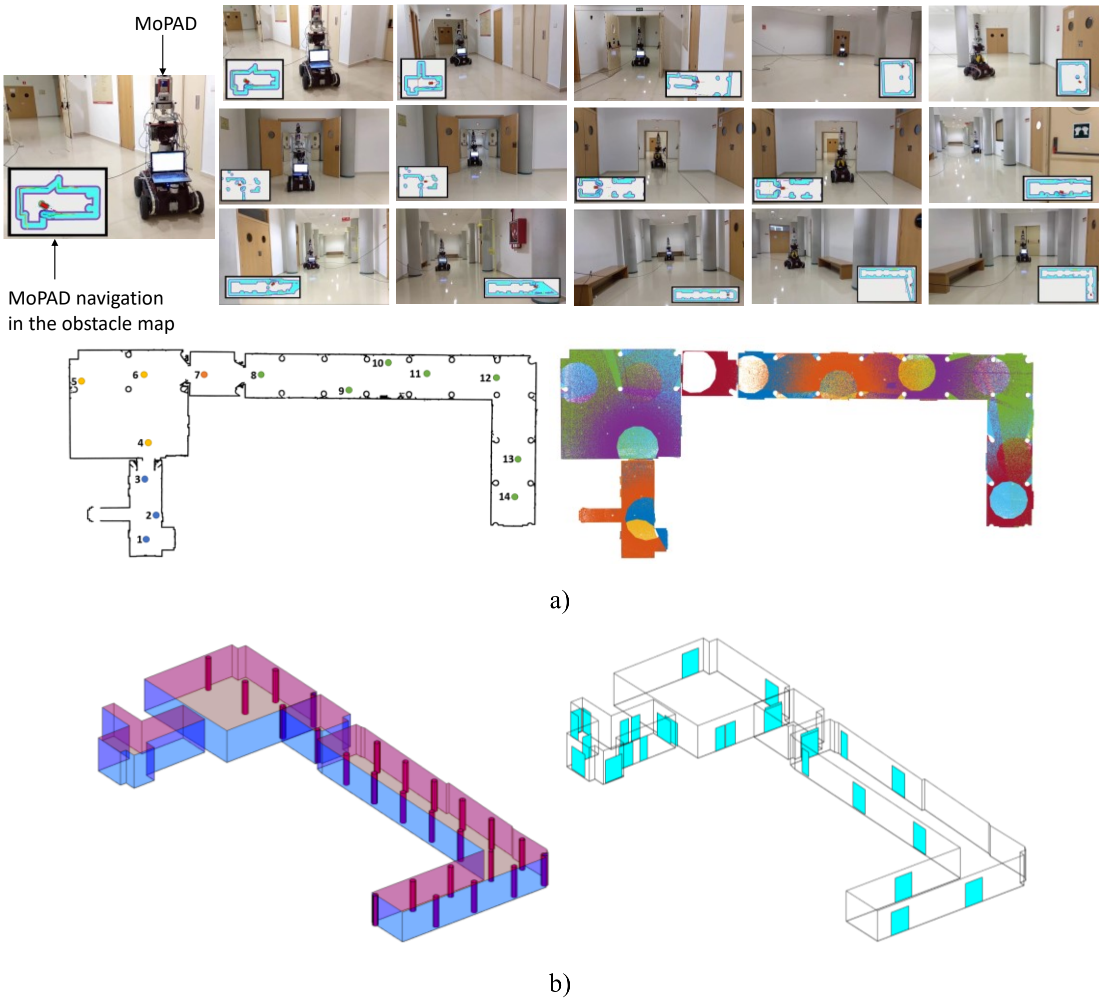


Figure 14. (a) up) The MoPAD platform scanning the indoor environment. MoPAD navigates from one position to another using the obstacle map and the navigation stack module. Down) Scanning positions and top view of *SM1*. Aligned point clouds in different colours are shown. (b) Details of model *SM3* (walls, ceiling, floor and columns) and *SM4* (openings).

Precision and *Recall* metrics were used to evaluate the pose and size of the recognised openings. We computed the overlap between the areas of the ground truth and the recognised opening, and evaluated the true-positive (t_p) , false-positive (f_p) and false negative (f_n) detected areas as illustrated in Figure 15(b). The average values were Precision=0.98 and Recall=0.94. The absolute and relative errors, e_{abs} and e_r, were also calculated as shown in equations (2) and (3). Their average values were found to be $e_{abs} = 0.31$ m2 and $e_r = 0.08$, respectively.

$$e_{abs} = f_p + f_n \tag{2}$$

$$e_r = \frac{f_p + f_n}{t_p + f_n} \tag{3}$$

- *Level 5*. A contribution of our work compared to others is the inclusion of BSCs into the semantic 3D model of the building. In this study case, 49 different objects were grouped into seven classes: Socket/Switch, Fire-Alarm Switch, Fire Sign, Exit Sign, Electric Sign, Exit Light, Door Sign. A total of 43 out of 49 objects were successfully detected on the walls of the scene.

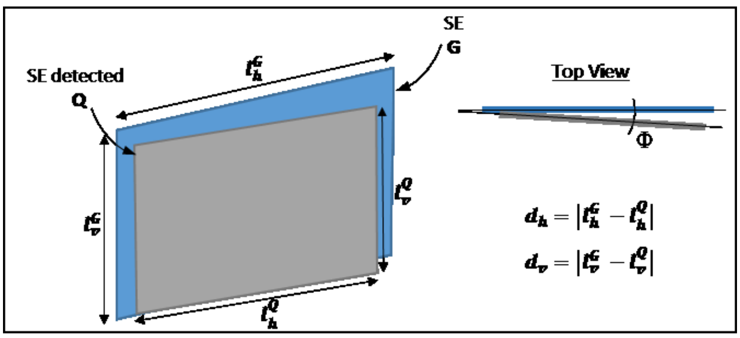
  Table 2 shows the percentages of true positive, false positive and false negative for each class. The worst percentage corresponds to sockets and switches owing to the smaller size and lack of texture of these objects. Indeed, the selected scanner resolution led to SE orthoimages generated with a resolution of 1cm/pixel, and therefore the size of orthoimages of switches was typically 15 x 15 pixels.

  We measured the localisation accuracy by calculating the horizontal and vertical errors between the centres of the recognised and the ground truth objects on the wall plane (Figure 15 (c)). The mean horizontal and vertical errors were 1.3 cm and 2.6 cm, respectively.

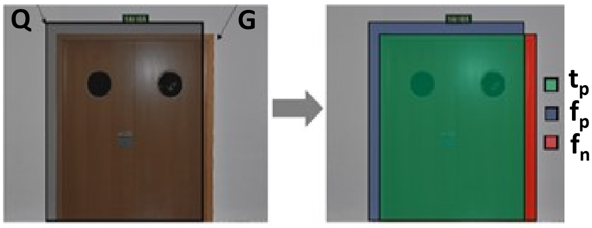
Figure 16 presents details of the detection of BSCs and the *SM5* model. A 3D flythrough video of this model is also available at the supplementary material.

Table 2. Detection of BSC. Recognition results. (N=number of objects, TP=true positive, FP=false positive, FN=false negative)

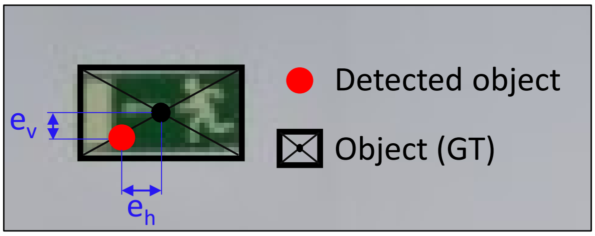
	Class	N	TP (%)	FP (%)	FN (%)	TP with α=0.5	TP with α>0.9
#1	Socket/Switch	14	28.6	57.1	14.3	100	0
#2	Fire-Alarm Switch	2	100	0	0	0	100
#3	Fire Sign	7	85.7	14.3	14.3	100	0
#4	Exit Sign	6	66.7	33.3	0	100	0
#5	Electric Sign	2	50.0	50.0	0	100	0
#6	Emergency Light	7	71.4	28.6	0	20	80
#7	Door Sign	11	63.6	9.1	27.3	71	28



(a)

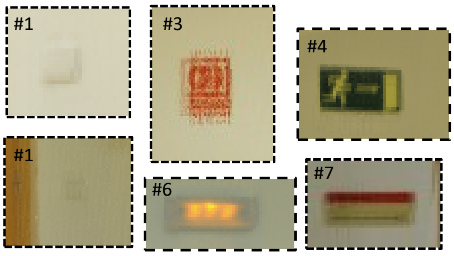


(b)

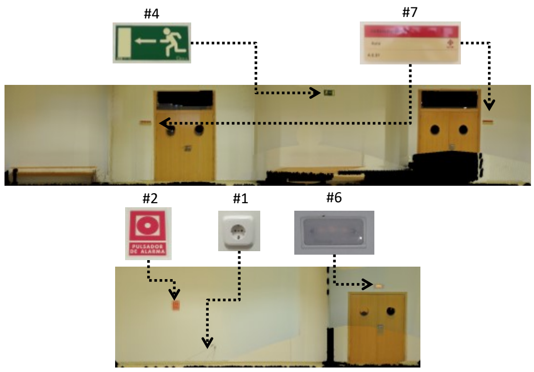


(c)

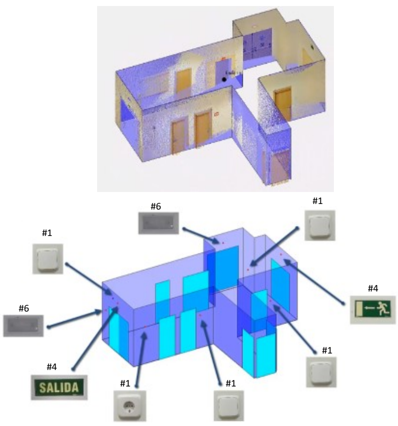
Figure 15. (a) Illustration of accuracy measurements of the SEs detected: d_v, d_h and ϕ. (b) Definition of t_p, f_p and f_n areas in the opening detection approach. (c) Positioning errors of BSCs.



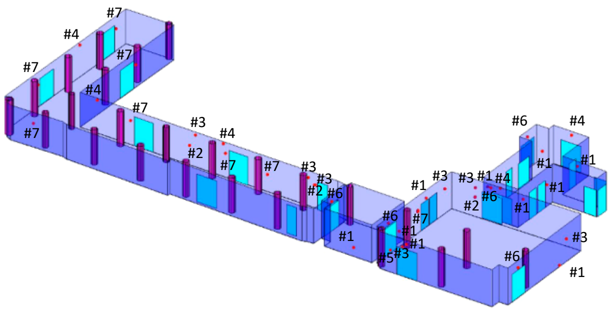
(a)



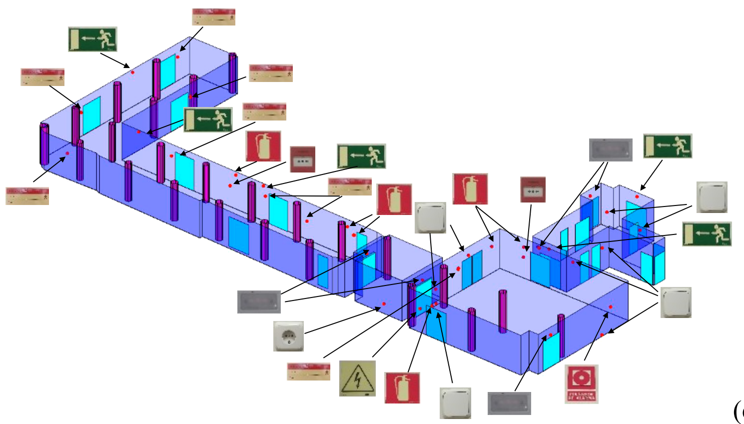
(b)



(c)



(d)



(e)

Figure 16. a) Blurred orthoimages of several BSCs. (b) Examples of BSCs recognised and classified in SEs. (c) *SM₅* of room #1. (d) and (e) The *SM₅* model. Red spots represent the objects detected.

4.3 *Experimentation on a simulated environment*

We have analysed the results obtained in a simulated case study consisting of the first floor of a typical office building. The advantage of simulated data is that the ground truth model regarding and the location of each object at each semantic level is exact.

The simulation of our automated scanning platform was carried out with Blensor [69]. Blensor can simulate the acquisition of coloured 3D data obtained from different LIDAR devices. In our case, the simulated Riegl VZ-400 laser scanner obtains data that are automatically aligned in a universal reference system. Blensor also makes it possible to inject noise into the data, signifying that our method can be tested in realistic simulation conditions.

The simulated scenario is depicted in Figure 17. The scene is a building floor of 22.9 m x 19.4 m in size, composed of 5 rooms, with 24 walls, 8 doors, 10 windows and 116 BSCs. Figure 18. shows the final result obtained after passing through all semantic levels. Discussions of the results obtained for each level are provided in the following paragraphs.

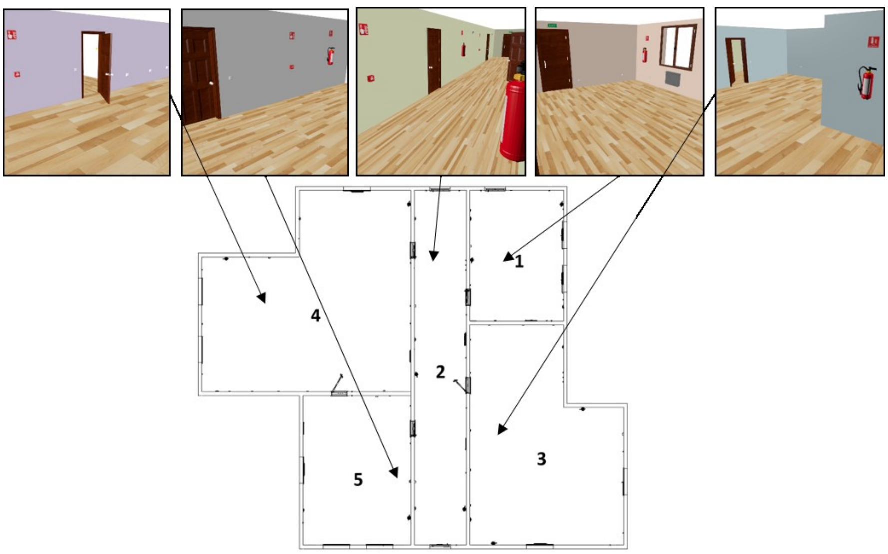


Figure 17. The case study. The building plan and details of some BSCs.

- *Level 1*. The complete scenario required 30 scans, with a total of 313 million points. The total point cloud was segmented into three data types: inliers, outliers and clutter (SM_1). Inliers are data which belong to SEs of the scene. Outliers are points collected from outside the room (owing to the fact that the laser scanner beam passes through open windows and doors) or incorrect data originating from shiny and reflective surfaces.

- *Levels 2 and 3*. The simplified 3D model is a polyhedron with 34 patches, 72 edges and 48 vertices organised in a relational graph structure (SM_2). In level 3, the faces are labelled as walls, ceilings, floors or columns, thus extending the semantic content of the 3D model (SM_3). We found 24 walls, 5 floors and 5 ceilings. There were no false negative or false positive cases.

  The accuracy of the SEs detected was: $d_v = 3.1$ cm, $d_h = 4.8$ cm, $d_{v\_max} = 3.1$ cm and $d_{h\_max} = 15.0$ cm. The average and maximum values of the orientation error were $\Phi = 0.03$º and $\Phi_{max} = 0.41$º, respectively.

- *Level 4*. Our method was able to detect all the existing openings: a total of 13 doors and 10 windows. The opening state of the doors was also successfully calculated, with 0 open-doors, 9 closed-doors and 4 semi-open doors correctly labelled.

  The evaluation using the Precision and Recall metric was: Precision=0.99, Recall=0.92, $f_{n\_max} = 0.49\, m^2$ and $f_{p\_max} = 0.02\, m^2$, respectively. The average values of the absolute and relative errors were $e_{abs} = 0.15\, m^2$ and $e_r = 0.08$, respectively.

- *Level 5*. A database was created with some of the most frequent objects found on the walls of a standard building. The scenario had 116 small BSCs, and our approach detected 101 true objects (87.1%), 6 false objects (5.1%) and was not able to recognise 15 objects (12.9%). Horizontal and vertical location errors were 2.4 mm and 2.3 mm, respectively. Table 3 shows more details of TP, FN and FP per object class. The last two columns contain those percentages of TP in which the average *Recognition Coherence Level* α is higher than 0.9 and equal to 0.5 respectively.

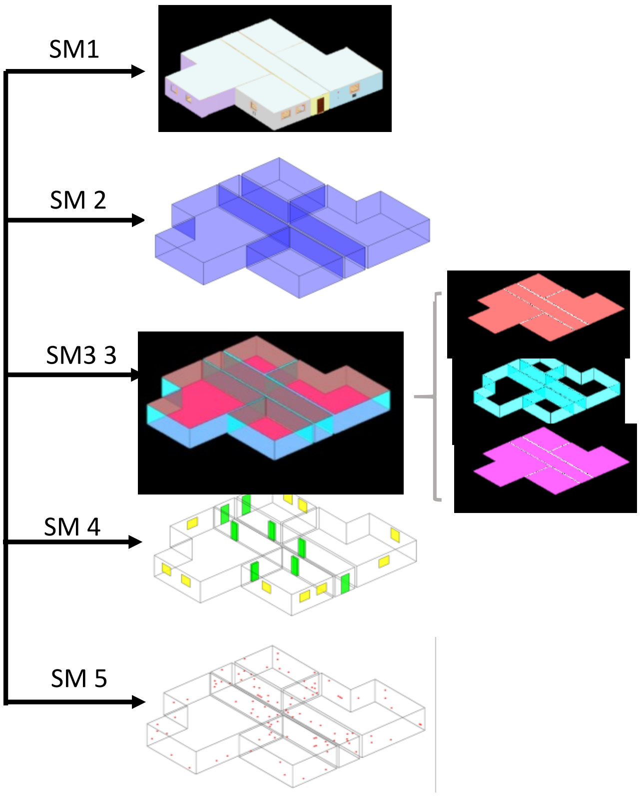
  Low values correspond to objects that are mainly recognised by either colour or by geometry (e.g. built-in socket, extinguisher sign), whereas high values are those that are identified in both orthoimages $\hat{J}_D$ and $\hat{J}_C$ (e.g. socket x2, switch).

Figure 18 shows the five semantic 3D models obtained and the SM_5 with some details.

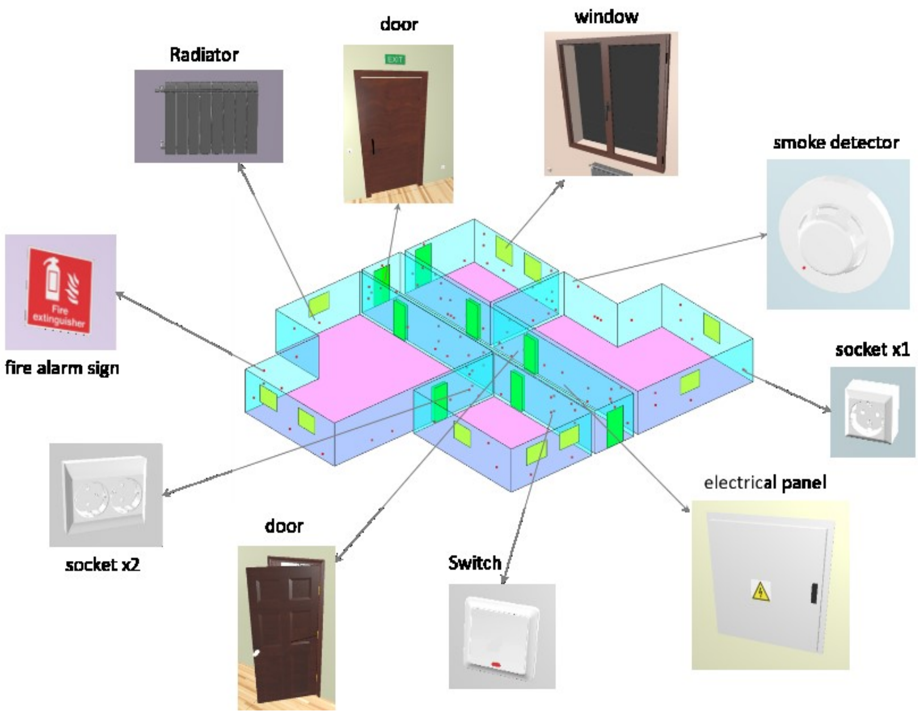
Table 3. Recognition results for building service components in simulated data.

Object	N	TP (%)	FN (%)	FP (%)	TP with α>0.9	TP with α=0.5
Electrical Panel	4	75.0	25.0	0	33.3	66.7
Socket x1	20	100	0	0	30.0	70
Socket x2	6	100	0	0	100	0
Socket x4	11	81.8	18.1	9.1	77.8	22.2
Built-in Socket	6	50.0	50.0	0	0	100
Switch	16	87.5	12.5	12.5	78.5	21.5

Fire Extinguisher	9	88.8	11.1	0	37.5	62.5
Radiator	6	83.3	16.6	0	20.0	80
Fire Alarm Switch	8	87.5	12.5	12.5	42.9	57.1
Smoke Detector	10	90.0	10.0	0	66.7	33.3
Exit Light	3	100	0	0	66.7	33.3
Extinguisher Sign	9	77.7	22.2	22.2	0	100
Fire Alarm Switch Sign	8	87.5	12.5	0	0	100



(a)



(b)

Figure 18. (a) The five semantic 3D models obtained. In level 5, the centroids of the BSCs detected are plotted in red, and the openings are removed for clarity. (b) The *SM₅* model. Details with the location and images of some of the small components detected.

4.4 *Discussion*

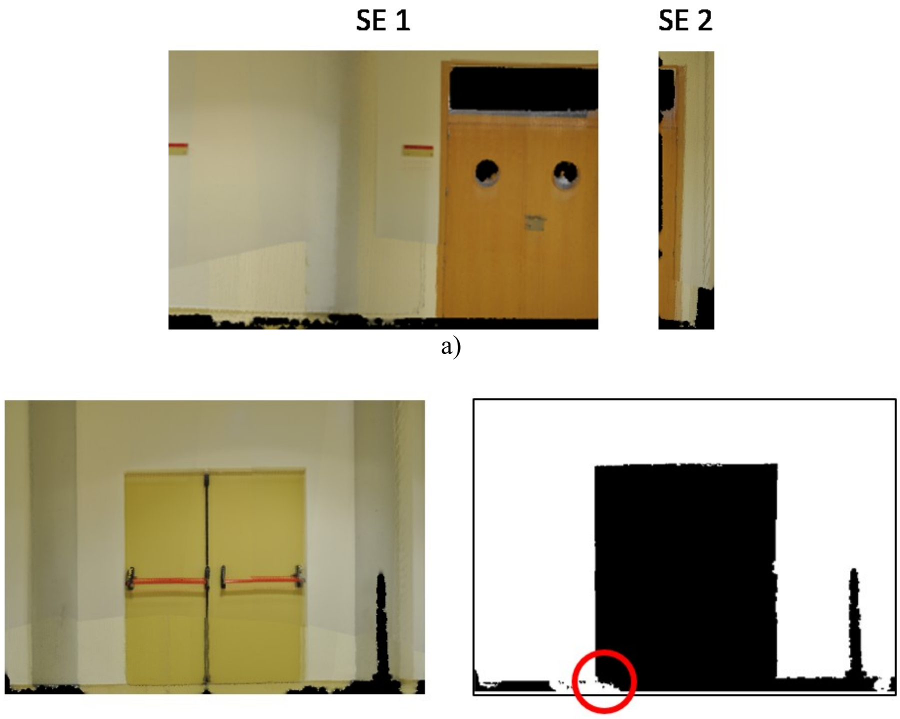
This section discusses the performance of MoPAD in the real and simulated environments.

Differences in Level 1 are in the density/quality of the data and the final point cloud. In simulation, the building indoor model is composed of perfect flat SEs with homogeneous colours. However, in reality, the SEs are not perfect flat surfaces and there exist slight variations in the colour, which depend on several factors (illumination of the scene, shadows, etc).

Although a small amount of noise is programmed in the simulated data acquisition process, more accurate and dense point clouds are obtained (resolution of 4 pixel/cm^2) than in the real case (resolution of 1 pixel/cm^2). Besides, the simulation software automatically performs precise registration between different scans. In the real case, as mentioned in Section 3.1, the data are registered using the robot's odometry and the ICP algorithm. It can be stated that, in general, the input data in simulation are more precise in geometry and colour than the real input data. This is important to highlight because this affects the results for the other levels, particularly level 5.

For levels 2 and 3, the system provides very good results in both cases. Regarding the detection of SEs, the real case achieves percentages higher than 95% while no errors are recorded in simulation. The errors d_v and d_h are, in both cases, below 5 cm, which is a reasonable result, although more accurate modelling may be required depending on the application. Finally, errors in Φ are below 1º, which also demonstrates that the system works successfully in both environments.

For level 4, the results in simulation were excellent, with all existing openings successfully detected. In the case of the real data, although the recognition rate (true positive cases) was high (near 90%), the algorithm failed in some cases. The main causes of these failures are the presence of occluded zones around doors or from slight colour variation between door and wall. Figure 19 shows two examples in which the detection of doors fails. In the first case (Figure 19 (a)), the SE detection algorithm fails, because a structural element is incorrectly split into several smaller structural elements. In the second case (Figure 19 (b)), the wall and door have similar colours, which led to a small region of the door being labelled as wall area, causing the failure.



b)

Figure 19. Examples of doors not detected in level 4.

For level 5, the results are different in both environments. Although the percentages in the detection of objects on the walls were quite similar in both cases, in the real case the classification algorithm failed more than in the simulated case. The true positive percentages per object were lower and the false positive percentages were higher. As already discussed in Section 3.4, the main reason of this is the lower quality of the orthoimages in the real case, in which small objects appear as blurred small orthoimage patches. On the contrary, the simulated point clouds provides higher quality orthoimages which yields better recognition results.

5 Conclusions

Many of the current automatically-generated 3D models representing large environments describe the scene only at a geometric level, lacking semantic detail. In the field of building modelling, representations are typically simplified B-rep models in which the principal information concerns walls, ceiling, floor and openings. In this paper, we present a method to automate the creation of high-level semantic models of buildings using autonomous (non-commanded) mobile robots. After reviewing the state of the art in this area, it can be said that, to date, none of the comparable autonomous systems achieves this level of semantic detail.

Our MoPAD platform autonomously digitises building interiors by scanning from locations that are calculated by an original NBS algorithm. The system processes the collected data with the objective of finding essential parts of the architecture as well as significant secondary components (i.e. BSCs) lying on walls and ceilings. When the scan stopping criteria are verified, MoPAD creates a SM_5 model of the current room, identifies the exit door, enters the adjacent room and starts again the scanning process. The whole system has been tested under real conditions yielding promising results. The accuracy of 3D models generated from the proposed technique has been evaluated encouragingly against ground truth models in real and simulated environments.

6 Future Work

Despite the progress made to date, many challenges remain to be addressed. Our method still has several environmental restrictions and limitations at each semantic level. At level 1, the NBS algorithm should take into account the quality and density of the data in the SEs. This would improve the quality of orthoimage J_{CD} and therefore the object recognition results. The metric used to select the NBS location and overall point cloud quality could also be extended to include quality of data for BSCs (as opposed to SEs only, right now).

At levels 2 and 3, one primary aspect to be improved concerns the detection of non-planar structural elements, such as curve walls and irregular ceilings/floors with different heights, within the same room. At level 4, solutions for non-rectangular doors and more complex cases, including severe occlusion, must be addressed. With regard to level 5, new recognition algorithms, including deep learning techniques, should certainly be considered to achieve higher detection performances. Our efforts will also be focused towards dealing with a wider range of BSCs.

In order to increase the degree of autonomy of our mobile platform, the current research is also focussed on how our MoPAD passes from one room to another when the door is closed or semi-closed. Our opening recognition algorithm can detect the state of a door but, so far, the mobile robot can enter the adjacent room only when the door is open. In the case when the door is closed, the MoPAD should interact with the door handle and push/pull the door until the door is completely open. This entails supplying a manipulator to MoPAD and programming a precise manipulator-door interaction. This is a robotic issue that we are currently working on.

To date, our system has been tested on a building story composed of a few rooms. We are encouraged to test our system in more extended and difficult environments. Particularly, we are planning to automatically extract as-is SM_5 models of multi-story buildings. The big problem here is that MoPAD should recognize the lift doors, enter the lift and leave, all in a safely manner. Such degree of autonomy requires many other complex actions to be dealt with in the future.

Acknowledgment

This work has been supported by the Spanish Economy and Competitiveness Ministry [DPI2016-76380-R project] and by the University of Castilla-La Mancha [PREDUCLM16/23 human resources grant].

References

[1] I. Toschi, P. Rodríguez-Gonzálvez, F. Remondino, S. Minto, S. Orlandini, and A. Fuller, "Accuracy Evaluation of a Mobile Mapping System With Advanced Statistical Methods," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2015, DOI: 10.5194/isprsarchives-XL-5-W4-245-2015.

[2] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 3–4, pp. 181–198, 2003, DOI: 10.1016/j.robot.2003.09.004.

[3] R. Grabowski, P. Khosla, and H. Choset, "Autonomous exploration via regions of interest," *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, DOI: 10.1109/IROS.2003.1248887.

[4] D. Borrmann *et al.*, "A mobile robot based system for fully automated thermal 3D mapping," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 425–440, 2014, DOI: 10.1016/j.aei.2014.06.002.

[5] M. Strand and R. Dillmann, "Using an attributed 2D-grid for next-best-view planning on 3D environment data for an autonomous robot," *Proceedings of the IEEE International Conference on Information and Automation*, 2008, DOI: 10.1109/ICINFA.2008.4608017.

[6] P. S. Blaer and P. K. Allen, "Data acquisition and view planning for 3-D modeling tasks," *International Conference on Intelligent Robots and Systems*, pp. 417–422, 2007, DOI: 10.1109/IROS.2007.4399581.

[7] B. Charrow *et al.*, "Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping," in *Robotics: Science and Systems XI*, 2015.

[8] L. Iocchi and S. Pellegrini, "Building 3d Maps With Semantic Elements Integrating 2D Laser, Stereo Vision And IMU On A Mobile Robot," in *Proceedings of the 2nd ISPRS International Workshop on 3D-ARCH*, 2007.

[9] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148–164, 2014, DOI: 10.1016/j.jvcir.2013.07.006.

[10] P. Kim, J. Chen, and Y. K. Cho, "SLAM-driven robotic mapping and registration of 3D point clouds," *Automation in Construction*, vol. 89, no. May, pp. 38–48, May 2018, DOI: 10.1016/j.autcon.2018.01.009.

[11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, pp. 1–16, 2016, DOI: 10.1007/s10514-016-9610-0.

[12] Z. Meng *et al.*, "A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017, DOI: 10.1109/LRA.2017.2655144.

[13] R. Kurazume, S. Oshima, S. Nagakura, Y. Jeong, and Y. Iwashita, "Automatic large-scale three dimensional modeling using cooperative multiple robots," *Computer Vision and Image Understanding*, vol. 157, pp. 25–42, 2016, DOI: 10.1016/j.cviu.2016.05.008.

[14] J. Jung *et al.*, "Productive modeling for development of as-built BIM of existing indoor structures," *Automation in Construction*, vol. 42, pp. 68–77, 2014, DOI: 10.1016/j.autcon.2014.02.021.

[15] C. Wang, Y. K. Cho, and C. Kim, "Automatic BIM component extraction from point clouds of existing buildings for sustainability applications," *Automation in Construction*, vol. 56, pp. 1–13, 2015, DOI: 10.1016/j.autcon.2015.04.001.

[16] C. Mura, O. Mattausch, A. Jaspe Villanueva, E. Gobbetti, and R. Pajarola, "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts," *Computers & Graphics*, vol. 44, pp. 20–32, Nov. 2014, DOI: 10.1016/j.cag.2014.07.005.

[17] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Computers & Graphics*, vol. 54, pp. 94–103, 2016, DOI: 10.1016/j.cag.2015.07.008.

[18] L. Díaz-Vilariño, K. Khoshelham, J. Martínez-Sánchez, and P. Arias, "3D Modeling of Building Indoor Spaces and Closed Doors from Imagery and Point Clouds," *Sensors*, vol. 15, no. 2, pp. 3491–3512, 2015, DOI: 10.3390/s150203491.

[19] A. Budroni and J. Boehm, "Automated 3D Reconstruction of Interiors from Point Clouds," *International Journal of Architectural Computing*, vol. 8, no. 1, pp. 55–74, 2010, DOI: 10.1260/1478-0771.8.1.55.

[20] M. Previtali, L. Barazzetti, R. Brumana, and M. Scaioni, "Towards automatic indoor reconstruction of cluttered building rooms from point clouds," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014, DOI: 10.5194/isprsannals-II-5-281-2014.

[21] J. Xiao and Y. Furukawa, "Reconstructing the World's Museums," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 243–258, 2014, DOI: 10.1007/s11263-014-0711-y.

[22] A. Adan and D. Huber, "3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter," *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 275–281, 2011, DOI: 10.1109/3DIMPVT.2011.42.

[23] X. Xiong, A. Adan, B. Akinci, and D. Huber, "Automatic creation of semantically rich 3D building models from laser scanner data," *Automation in Construction*, vol. 31, pp. 325–337, 2013, DOI: 10.1016/j.autcon.2012.10.006.

[24] D. Hähnel, W. Burgard, and S. Thurn, "Learning compact 3D models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems*, 2003, DOI: 10.1016/S0921-8890(03)00007-1.

[25] S. Ikehata, H. Yan, and F. Yasutaka, "Structured Indoor Modeling," *The IEEE*

International Conference on Computer Vision (ICCV)2, pp. 1323–1331, 2015, DOI: 10.1109/ICCV.2015.156.

[26] I. Armeni *et al.*, “3D Semantic Parsing of Large-Scale Indoor Spaces,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1534–1543, 2016, DOI: 10.1109/CVPR.2016.170.

[27] E. Lopez-Damian, G. Etcheverry, L. E. Sucar, and J. Lopez-Estrada, “Probabilistic view planner for 3D modelling indoor environments,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4021–4026, 2009, DOI: 10.1109/IROS.2009.5354375.

[28] I. Maurović, M. Đakulović, and I. Petrović, “Autonomous Exploration of Large Unknown Indoor Environments for Dense 3D Model Building,” in *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, 2014, no. Section III, pp. 10188–10193.

[29] X. Yang and Y. Tian, “Robust door detection in unfamiliar environments by combining edge and corner features,” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pp. 57–64, 2010, DOI: 10.1109/CVPRW.2010.5543830.

[30] M. M. Shalaby, M. A. M. Salem, A. Khamis, and F. Melgani, “Geometric model for vision-based door detection,” *Proceedings of 2014 9th IEEE International Conference on Computer Engineering and Systems, ICCES 2014*, pp. 41–46, 2015, DOI: 10.1109/ICCES.2014.7030925.

[31] A. Andreopoulos and J. K. Tsotsos, “Active vision for door localization and door opening using playbot: A computer controlled wheelchair for people with mobility impairments,” *Proceedings of the 5th Canadian Conference on Computer and Robot Vision, CRV 2008*, pp. 3–10, 2008, DOI: 10.1109/CRV.2008.23.

[32] W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang, and G. Fu, “Door recognition and deep learning algorithm for visual based robot navigation,” *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pp. 1793–1798, 2014, DOI: 10.1109/ROBIO.2014.7090595.

[33] R. Sekkal, F. Pasteau, M. Babel, B. Brun, and I. Leplumey, “Simple monocular door detection and tracking,” *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, pp. 3929–3933, 2013, DOI: 10.1109/ICIP.2013.6738809.

[34] S. Kim, H. Cheong, D. H. Kim, and S. K. Park, “Context-based object recognition for door detection,” in *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics, ICAR 2011*, 2011, pp. 155–160.

[35] A. H. Adiwahono, Y. Chua, K. P. Tee, and B. Liu, “Automated door opening scheme for non-holonomic mobile manipulator,” *International Conference on Control, Automation and Systems*, no. Iccas, pp. 839–844, 2013, DOI: 10.1109/ICCAS.2013.6704030.

[36] L. C. Goron, L. Tamas, and G. Lazea, “Classification within indoor environments using 3D perception,” *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on*, pp. 400–405, 2012, DOI: 10.1109/AQTR.2012.6237743.

[37] S. M. Z. Borgsen, M. Schöpfer, L. Ziegler, and S. Wachsmuth, “Automated door

detection with a 3D-sensor," *Proceedings - Conference on Computer and Robot Vision, CRV 2014*, pp. 276–282, 2014, DOI: 10.1109/CRV.2014.44.

[38] K. M. Varadarajan and M. Vincze, "3D room modeling and doorway detection from indoor stereo imagery using feature guided piecewise depth diffusion," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 2758–2765, 2010, DOI: 10.1109/IROS.2010.5651525.

[39] T. H. Yuan, F. H. Hashim, W. M. D. W. Zaki, and A. B. Huddin, "An Automated 3D Scanning Algorithm using Depth Cameras for Door Detection," *2015 International Electronics Symposium (IES)*, pp. 58–61, 2015, DOI: 10.1109/ELECSYM.2015.7380814.

[40] M. Derry and B. Argall, "Automated doorway detection for assistive shared-control wheelchairs," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1254–1259, 2013, DOI: 10.1109/ICRA.2013.6630732.

[41] N. Banerjee *et al.*, "Human-Supervised Control of the ATLAS Humanoid Robot for Traversing Doors," *IEEE-RAS International Conference on Humanoid Robots*, pp. 722–729, 2015, DOI: 10.1109/HUMANOIDS.2015.7363442.

[42] J. Xu, K. Kim, L. Zhang, and D. Khosla, "3D Perception for Autonomous Robot Exploration," *International Symposium on Visual Computing*, pp. 888–900, 2015, DOI: 10.1007/978-3-319-27857-5 79.

[43] A. Budroni and J. Böhm, "Automatic 3D Modelling of Indoor Manhattan-World Scenes From Laser Data," vol. XXXVIII, 2010.

[44] B. Kakillioglu, K. Ozcan, and S. Velipasalar, "Doorway detection for autonomous indoor navigation of unmanned vehicles," *IEEE International Conference on Image Processing (ICIP)*, pp. 3837–3841, 2016, DOI: 10.1109/ICIP.2016.7533078.

[45] E. Valero, A. Adan, D. Huber, and C. Cerrada, "Detection, Modeling, and Classification of Moldings for Automated Reverse Engineering of Buildings from 3D Data," in *International Symposium on Automation and Robotics in Construction (ISARC)*, 2011.

[46] D. Dai *et al.*, "Detecting, locating and crossing a door for a wide indoor surveillance robot," *2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013*, no. December, pp. 1740–1746, 2013, DOI: 10.1109/ROBIO.2013.6739719.

[47] H. Hamledari, B. Mccabe, and S. Davari, "Automated computer vision-based detection of components of under-construction indoor partitions," 2017, DOI: 10.1016/j.autcon.2016.11.009.

[48] T. Czerniawski, M. Nahangi, C. Haas, and S. Walbridge, "Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor," *Automation in Construction*, vol. 71, pp. 346–358, Nov. 2016, DOI: 10.1016/j.autcon.2016.08.011.

[49] Y. Turkan, F. Bosché, C. T. Haas, and R. Haas, "Tracking of secondary and temporary objects in structural concrete work - ProQuest," *Construction Innovation*, vol. 14, no. 2, pp. 145–167, Aug. 2014, DOI: http://dx.doi.org/10.1108/CI-12-2012-0063.

[50] L. Díaz-Vilariño, H. González-Jorge, J. Martínez-Sánchez, and H. Lorenzo, “Automatic LiDAR-based lighting inventory in buildings,” *Measurement: Journal of the International Measurement Confederation*, vol. 73, pp. 544–550, 2015, DOI: 10.1016/j.measurement.2015.06.009.

[51] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, “Automatic detection of road tunnel luminaires using a mobile LiDAR system,” *Measurement: Journal of the International Measurement Confederation*, vol. 47, no. 1, pp. 569–575, 2014, DOI: 10.1016/j.measurement.2013.09.044.

[52] V. Eruhimov and W. Meeussen, “Outlet detection and pose estimation for robot continuous operation,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 2941–2946, 2011, DOI: 10.1109/IROS.2011.6048578.

[53] U. Krispel, H. L. Evers, M. Tamke, R. Viehauser, and D. W. Fellner, “Automatic texture and orthophoto generation from registered panoramic views,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 40, no. 5W4, pp. 131–137, 2015, DOI: 10.5194/isprsarchives-XL-5-W4-131-2015.

[54] T. M. Bonanni, A. Pennisi, D. Bloisi, L. Iocchi, and D. Nardi, “Human-Robot Collaboration for Semantic Labeling of the Environment,” in *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration*, 2013, no. July 2015, pp. 1–25.

[55] E. Valero, A. Adán, and F. Bosché, “Semantic 3D Reconstruction of Furnished Interiors Using Laser Scanning and RFID Technology,” *Journal of Computing in Civil Engineering*, vol. 30, no. 4, p. 4015053, Sep. 2016, DOI: 10.1061/(ASCE)CP.1943-5487.0000525.

[56] M. Gunther, T. Wiemann, S. Albrecht, and J. Hertzberg, “Building semantic object maps from sparse and noisy 3D data,” *IEEE International Conference on Intelligent Robots and Systems*, no. 287752, pp. 2228–2233, 2013, DOI: 10.1109/IROS.2013.6696668.

[57] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D Point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008, DOI: 10.1016/j.robot.2008.08.005.

[58] N. Blodow *et al.*, “Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments,” in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 4263–4270.

[59] A. Nüchter and J. Hertzberg, “Towards semantic maps for mobile robots,” *Robotics and Autonomous Systems*, 2008, DOI: 10.1016/j.robot.2008.08.001.

[60] S. A. Prieto, B. Quintana, A. Adán, and A. S. Vázquez, “As-is building-structure reconstruction from a probabilistic next best scan approach,” *Robotics and Autonomous Systems*, vol. 94, pp. 186–207, 2017, DOI: 10.1016/j.robot.2017.04.016.

[61] B. Quintana, S. A. Prieto, A. Adán, and A. S. Vázquez, “Semantic Scan Planning for Indoor Structural Elements of Buildings,” *Advanced Engineering Informatics*, 2016, DOI: 10.1016/j.aei.2016.08.003.

[62] B. Quintana, S. A. Prieto, A. Adán, and F. Bosché, “Door detection in 3D coloured point clouds of indoor environments,” *Automation in Construction*, vol. 85, pp.

146–166, 2018, DOI: 10.1016/j.autcon.2017.10.016.

[63] A. Adán, B. Quintana, S. A. Prieto, and F. Bosché, “Scan-to-BIM for ‘secondary’ building components,” *Advanced Engineering Informatics*, vol. 37, pp. 119–138, 2018, DOI: 10.1016/J.AEI.2018.05.001.

[64] D. Wolf, a. Howard, and G. S. Sukhatme, “Towards geometric 3D mapping of outdoor environments using mobile robots,” *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, DOI: 10.1109/IROS.2005.1545152.

[65] C. Stachniss and W. Burgard, “Exploring unknown environments with mobile robots using coverage maps,” *Proc. of the International Conference on Artificial Intelligence (IJCAI)*, pp. 1127–1132, 2003, DOI: 1630659.1630821.

[66] P. H. S. Torr and A. Zisserman, “MLESAC: A New Robust Estimator with Application to Estimating Image Geometry,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000, DOI: 10.1006/cviu.1999.0832.

[67] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157 vol.2.

[68] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006, DOI: 10.1007/11744023_32.

[69] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, “BlenSor: Blender sensor simulation toolbox,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6939 LNCS, no. PART 2, pp. 199–208, 2011, DOI: 10.1007/978-3-642-24031-7_20.