Improving Semantic Mapping with Prior Object Dimensions Extracted from 3D Models

Abdessalem Achour, Hiba Al Assaad, Yohan Dupuis, Madeleine El Zaher

Improving Semantic Mapping with Prior Object Dimensions Extracted from 3D Models

Abdessalem Achour[1,3][0000−0002−0320−2703], Hiba Al Assaad[1][0000−0003−1545−7022], Yohan Dupuis[2][0000−0002−9725−2049], and Madeleine El Zaher[1][0000−0003−2841−0303]

[1] LINEACT CESI, Campus of Toulouse, 31670 Labège, France
[2] LINEACT CESI, Paris La Défense, 92074 Paris, France
[3] SMI doctoral school, HESAM University, 75013 Paris, France
{aachour, halassaad, ydupuis, melzaher}@cesi.fr

Abstract. Semantic mapping in mobile robotics has gained significant attention recently for its important role in equipping robots with a comprehensive understanding of their surroundings. This understanding involves enriching metric maps with semantic data, covering object categories, positions, models, relations, and spatial characteristics. This augmentation enables robots to interact with humans, navigate semantically using high-level instructions, and plan tasks efficiently. This study presents a novel real-time RGBD-based semantic mapping method designed for autonomous mobile robots. It focuses specifically on 2D semantic mapping in environments where prior knowledge of object models is available. Leveraging RGBD camera data, our method generates a primitive object representation using convex polygons, which is then refined by integrating prior knowledge. This integration involves utilizing predefined bounding boxes derived from real 3D object dimensions to cover real object surfaces. The evaluation, conducted in two distinct office environments (a simple and a complex setting) utilizing the MIR mobile robot, demonstrates the effectiveness of our approach. Comparative analysis showcases our method outperforming a similar state-of-the-art approach utilizing only RGBD data for mapping. Our approach accurately estimates occupancy zones of partially visible or occluded objects, resulting in a semantic map closely aligned with the ground truth.

Keywords: Semantic Mapping · Data Association · Prior Knowledge

1 INTRODUCTION

Mobile robots are increasingly finding applications across a wide range of settings, including homes, offices, healthcare facilities, and manufacturing environments. To execute their tasks effectively, these robots rely on having an accurate and up-to-date map of their surroundings. Traditional mapping techniques have primarily focused on metric or topological maps, which ensure safe navigation but often lack vital information about the environment, such as object categories and spatial relationships [1]. However, certain tasks demand a more pro-

found cognitive understanding of the environment, such as human-robot collaboration, semantic navigation [2], and object manipulation. Additionally, robots are increasingly transitioning from specialized, single-task machines to general-purpose systems that operate in diverse environments. To address these challenges, semantic mapping emerges as a promising approach, enriching maps with high-level semantic knowledge, including object categories, shapes, 3D models, and object relationships. This enrichment allows robots to effectively generalize knowledge, learn, and be transparent in their decision-making processes [3].

Our research delves into indoor semantic mapping within the framework of a digital twin—a virtual replica that mirrors real-world entities in real-time. The fast integration of this technology across various applications, such as manufacturing [4], and the numerous studies interested in its adoption in other domains like agriculture [5,6], motivates our focus. Our methodology involves continuously updating a semantic map within the digital twin, providing real-time information on object categories, positions, and occupancy zones. This integrated semantic map serves as a supervision and analysis tool, facilitating swift responses during challenges and significantly enhancing the capabilities of robots. Consequently, robots can adapt their tasks promptly to changing surroundings, have increased flexibility and real-time decision-making capabilities.

Existing literature outlines two approaches to semantic mapping: 2D and 3D. In 2D mapping, the aim is object localization and identification, while 3D mapping delves into the spatial characteristics of objects [7,8]. Choice depends on the intended application. For tasks like semantic navigation [2], a 2D map suffices, enabling navigation using high-level instructions like "Go to the kitchen" or "Transport the box to the garage". Conversely, object manipulation demands 3D models for spatial understanding [9]. Context also matters, in environments where 3D models of objects are available beforehand, prioritizing 2D mapping can be efficient. Real-time applications might favor 2D data for timely decision-making. However, scenarios requiring complex object manipulation, augmented reality applications, and surface mapping in fields like archaeology or geological exploration, necessitate 3D mapping for accuracy.

In the domain of digital twins with accessible 3D models, the need for extensive transmission of additional 3D data might be mitigated. As a result, our research is currently directed towards refining precise 2D mapping, aiming to leverage existing 3D models to streamline data exchange and potentially enhance operational effectiveness. Prior 2D mapping works relied solely on sensor data. For instance, Zhao et al. [10] proposed a solution that incorporates voice instructions to annotate an occupancy grid map with object labels. In contrast, Qi et al. [11] present a different approach, enabling the incorporation of object topological information into an occupancy grid map. This involves utilizing an object detection model and a triangulation algorithm, leveraging odometry and stereo vision data to identify objects within point clouds. Subsequently, they employ minimum bounding rectangles to represent the topological space based on labeled point clouds. Additionally, Zaenker et al. [9] utilize RGBD data to represent the occupancy zones of objects on the occupancy map using polygons.

Their approach involves extracting object point clouds and simultaneously labeling them using a detection model. They employ the Quickhull algorithm [12] to approximate the occupation zones from these labeled point clouds. Similarly, Dengler et al. [13] propose an RGBD-based solution. They employ a CNN-based detection model and a segmentation algorithm to identify object point clouds. Then, the occupation zones of objects are determined from the labeled points clouds. Each object's point cloud was projected onto the map plane, and the Quickhull algorithm was used to represent the object's occupation zone through polygons. Furthermore, they introduce a more accurate representation using oriented bounding boxes.

Beyond object recognition, studies have focused on semantic place categorization using sensor data, intending to enable autonomous robots to discern area-specific semantic labels akin to human perception, such as "office" or "kitchen" [14]. Recent methodologies leverage diverse technologies: Hiller et al. [15] utilize 2D laser sensor data, employing image patches from 2D occupancy maps as input for Convolutional Neural Networks (CNNs) to determine precise locations. Kaleci et al. [16] employ a 2D deep learning architecture, annotating occupancy grid maps using laser data, while Posada et al. [17] use CNNs to categorize omnidirectional camera images.

Clearly, the majority of prior works in 2D semantic mapping relied solely on sensor data. In contrast, the focal point of our approach lies in 2D mapping, introducing a major novelty: how to integrate prior knowledge into the semantic mapping process to enhance the quality of the created map ? This approach distinguishes our work from existing methodologies, as it strategically leverages prior knowledge to augment the mapping process. Specifically, in this paper, we propose an RGBD-based semantic mapping solution that utilizes real object models to improve the approximation of the 2D occupation zone of objects. Our solution enables the approximation of complete occupancy zones of objects, even when only partial object representations are available. It achieves this by utilizing predefined bounding boxes derived from the real dimensions of objects extracted from 3D models. The primary advantage of our approach lies in its ability to estimate the occupancy zones of partially visible or occluded objects, leading to a more accurate semantic map that closely aligns with ground truth. We assess the performance of our mapping solution in two distinct office settings and conduct a comparative analysis with an alternative semantic mapping method [13].

Our paper is organized as follows: In Section 2, we provide a brief description of the overall semantic mapping process, followed by a detailed description of our method for integrating prior object dimensions extracted from 3D models into the mapping process. In Section 3, we present our experimental results, including a discussion of the experimental setup, evaluation metrics, and results analysis. Finally, in Section 4, we summarize our findings and discuss potential avenues for improvement.

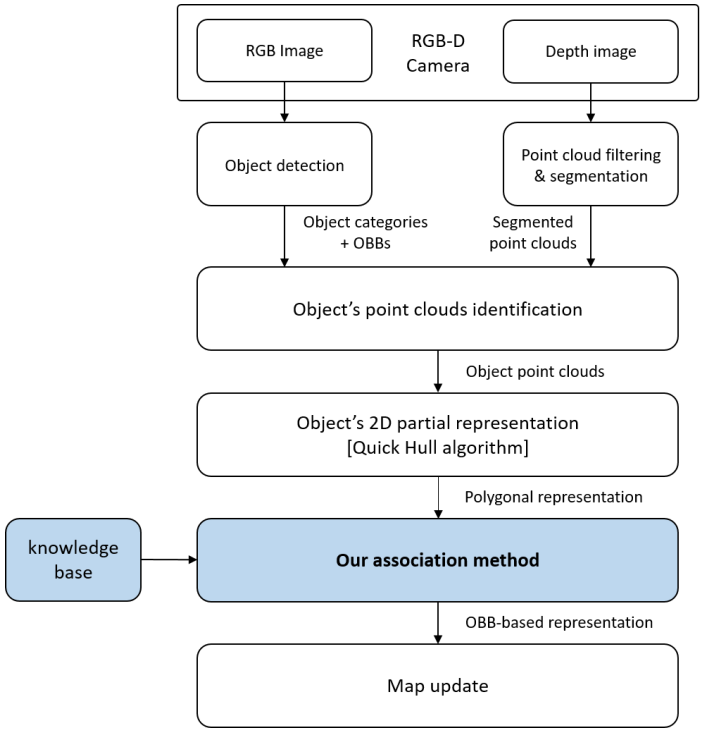
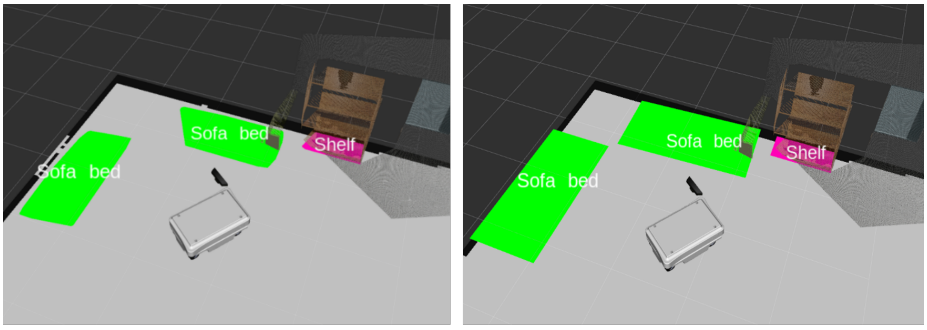


Fig. 1. Enhancing semantic object representation: Integration of our method within the semantic mapping process, based on the framework proposed in [13]. Modules highlighted in blue indicate the incorporation of prior knowledge for improved semantic object representation.

2 METHOD DESCRIPTION

In this research, we propose a real-time semantic mapping solution tailored for autonomous mobile robots, leveraging RGBD data. This paper specifically focuses on a novel association approach designed to generate 2D representations of semantic objects using both RGBD data and prior knowledge. The objective of our work is to deploy a semantic map onto a digital twin. This semantic map serves various purposes, primarily enhancing the capabilities of mobile robots and enabling them to perform more complex tasks. What sets our work apart is the utilization of prior knowledge provided by the digital twin, including information about objects existing in the environment and their numerical models. Additionally, our method is designed to be real-time, embeddable, and resource-efficient for deployment on a mobile robot, fulfilling the dual purpose of mapping the environment and updating the digital twin in real-time.

The overall process is briefly summarized in the following section, followed by a detailed description of our approach.



(a) Representation with Convex Polygons (b) Representation with Rectangular Bounding Boxes

Fig. 2. Semantic object occupancy zone representation

2.1 The overall semantic mapping process

There are two major semantic mapping approaches in mobile robotics: mono-robot approaches and collaborative approaches. In mono-robot approaches, a single robot is used for semantic mapping, whereas collaborative approaches involve multiple agents, which can be robots or other entities, working together in various ways to create the semantic map. This paper is interested in mono-robot semantic mapping. Readers interested in collaborative semantic mapping can refer to our review paper [1], where we conducted an in-depth study of both approaches.

Existing literature presents several approaches for semantic mapping. Our method, depicted in Figure 1, is proposed based on the RGBD-based approach introduced in [13]. This approach assumes the presence of an established occupancy grid map and knowledge of the global pose of the robot. The semantic mapping process involves employing a detection model to identify objects within the RGB image, combined with a rapid segmentation algorithm for segmenting objects within the depth image. An association step establishes links between each object detection and its corresponding segment in the point cloud, creating object point clouds.

To represent the spatial extent of objects, our method projects object point clouds onto the ground plane and uses the Quickhull algorithm [12] to generate a convex polygonal representation of the object. We choose this algorithm for its computational simplicity and efficiency in approximating convex shapes from point sets. It produces streamlined object representations with minimal computational overhead, making it suitable for real-time mapping applications. Figure 2.a illustrates object representations using convex polygons.

It's important to note that the generated polygonal representation captures only the observable portion of the object, as the robot's RGBD sensor detects partial aspects of objects within its operational range and field of view during navigation. To address this limitation, we incorporate prior knowledge of object dimensions from a pre-existing dataset of CAD models to construct a comple-

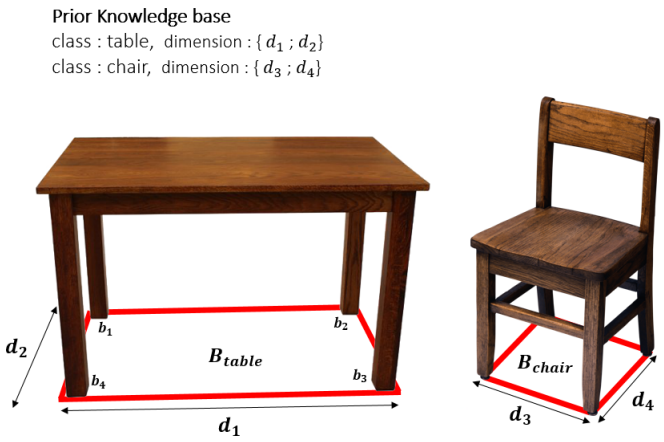


Fig. 3. Example of the prior knowledge base of objects existing in the environment.

mentary rectangular bounding box representation, as shown in Figure 2.b. This integration of prior information significantly enhances the accuracy and consistency of the resulting map. Moreover, the mapping process is incremental, allowing the assimilation of new information to fill in missing details or update object shapes and positions.

In the following, we assume that the polygon representing the object at each time point has already been constructed using the Quickhull algorithm. We then focus on the association method for the generation of the bounding box representation.

2.2 Prior knowledge

In our work, prior knowledge consists of a list of objects found in the environment. Each object is described by the dimensions length and width, represented respectively by l and w, of the rectangular bounding box enclosing its 2D projection to the ground. Indeed, bounding boxes are commonly used in many mapping approaches to represent the zone occupied by objects. They allow the representation of a wide variety of objects in the real world, ranging from simple geometric shapes to complex structures. Even curved objects can often be enclosed within a bounding box that is large enough to contain it without excessive waste of space. Additionally, bounding boxes reduce the complexity of object representation, as they are defined by a few simple parameters such as the center point, orientation, and size.

In our case, the environment being considered is an office environment, so objects such as tables, chairs, desks, etc. may be present. We have adopted a strategy of employing a single model for each object category. This choice is driven by our core objective, which is to validate our mapping technique. While the consideration of multiple models does not currently influence our mapping

process directly, it does introduce an additional challenge: the precise association of detections with the appropriate shape in our knowledge base. While this aspect is not our current primary focus, it remains a potential avenue for exploration in our future work.

For instance, when considering the "table" class, there is only one model present in the environment, and consequently, only one set of dimensions $\{l, w\}$ is stored in our prior knowledge base. Using these known dimensions, a predefined bounding box, denoted as $B_o = \{b_1, b_2, b_3, b_4\}$, with vertices defined in a clockwise direction, will be created to represent the occupancy of the object. Figure 3 shows an example of the prior knowledge and predefined bounding boxes for two objects: "table" and "chair".

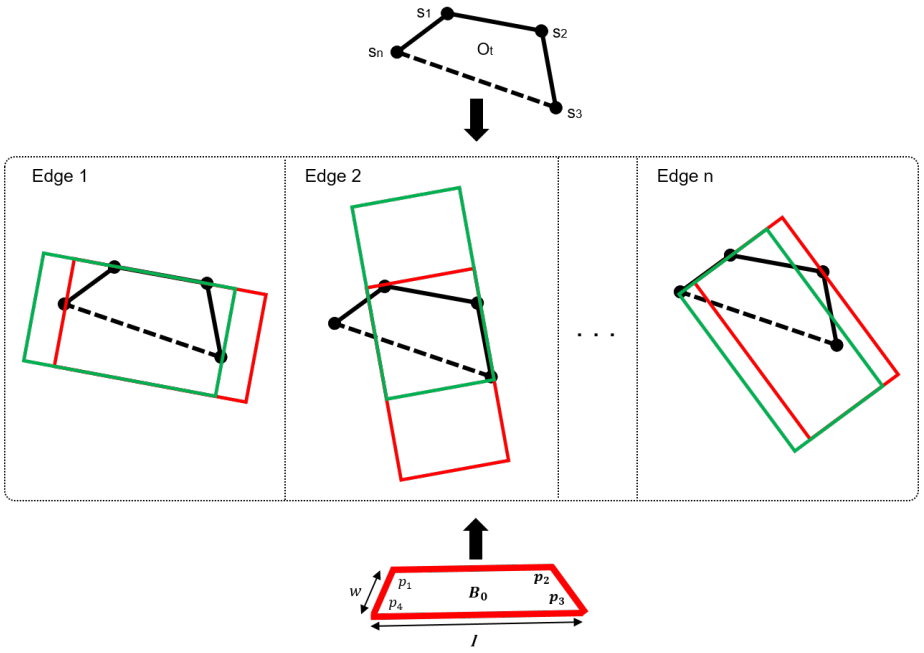


Fig. 4. Generation of candidate bounding boxes: O_t represents the polygon denoting the partial occupancy zone of the semantic object. The green and red boxes are two candidate options for augmenting this polygon. For each selected edge, the box orientation is determined by the edge's orientation, and two potential positions are proposed: the green box is positioned on the left side of the edge, aligned with its right extremum, while the red box is placed on the right side of the edge, aligned with its left extremum. Following the box generation process, the most suitable box will be chosen to represent the object.

2.3 2D geometric association method

The objective of our method is to establish a connection between the predefined bounding box B_o and the partial polygon of the object denoted as O_t at time t. This association enables us to estimate the actual occupied area of the object from partial occupancy.

Our approach is designed to determine both the orientation and position of the object by identifying potential connections between the object's bounding box and the edges of its polygon. To accomplish this, we introduce a method involving the generation of multiple candidate bounding boxes, followed by their association and evaluation, ultimately leading to the selection of the most suitable candidate to represent the object. The method is visually depicted in Figure 4. In particular, we propose creating two bounding boxes for each edge of the polygon : one shifted to the left (indicated in green) and the other to the right (indicated in red). This approach results in a total of $2n$ potential candidates, each exhibiting distinct positions and orientations, where n represents the number of edges of the polygon. Given that each edge can be associated with either the length or the width of the object, we apply this reasoning initially with bounding box dimensions {l, w}, and subsequently with dimensions {w, l}, resulting in a total of $4n$ candidate bounding boxes.

To evaluate the quality of each candidate association, we calculate an association score for each, selecting the bounding box with the highest score to accurately represent the object.

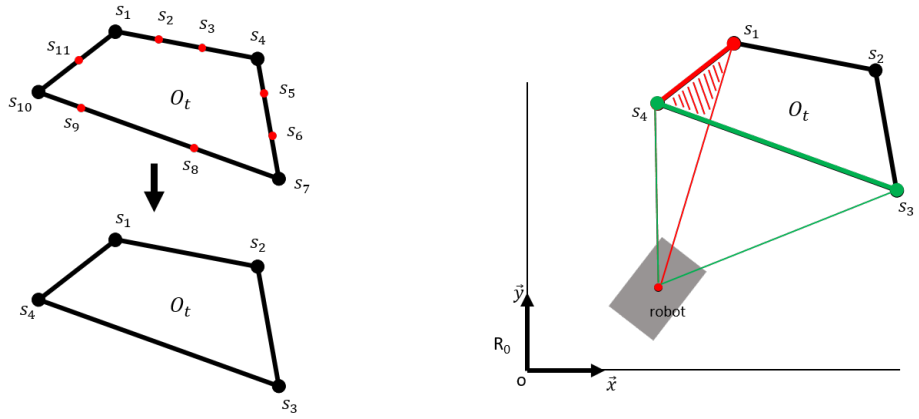
2.4 Algorithm description

Polygon simplification and foreground edges selection : The RGBD camera generates a high-density point cloud, leading to densely populated object point clouds. This density often causes over-segmented edges in polygons generated using the Quickhull algorithm. Consequently, this issue significantly increases the number of potential bounding boxes and extends processing time. Furthermore, this density factor directly affects the scoring function, which will be discussed later, as it relies on polygon edges length.

To tackle this challenge, we propose the introduction of a simplification step before the association process. This step involves merging consecutive edges within the polygon, especially those with angles greater than 178 degrees. Although these edges may visually appear as a single edge, they are actually composed of multiple smaller segments. Figure 5.a provides an example of a polygon before and after undergoing this process.

Moreover, our initial association process involved considering all polygon edges to generate candidate bounding boxes, leading to significant time consumption. To optimize this process, we now exclusively focus on foreground edges in relation to the robot, as they are more likely to be observed.

To determine whether a particular edge is in the foreground of the robot or not, we propose a method involving the computation of a triangle using the edge's vertices and the robot's position, as shown in Figure 5.b. Subsequently, we assess



(a) Polygon simplification

(b) Foreground Edges identification

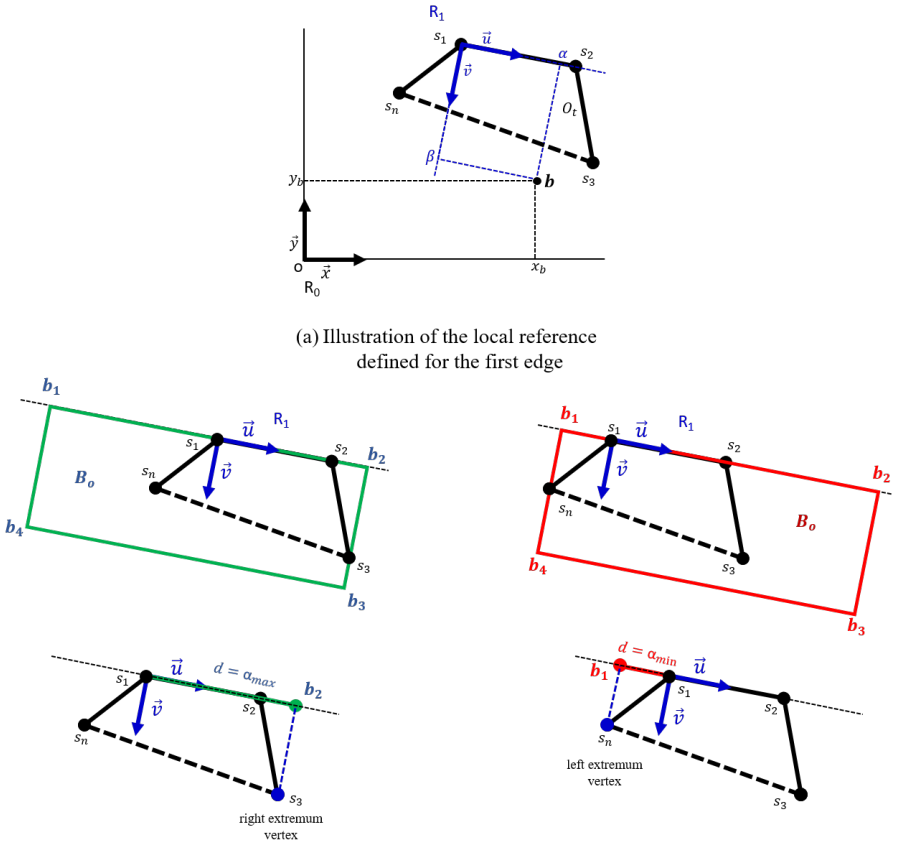
Fig. 5. Illustration of polygon simplification and selection of foreground edges: (a) The simplification step merges consecutive edges in the polygon with angles greater than 178 degrees, resulting in the removal of all red vertices. The total number of edges in the polygon is reduced from 11 to 4. (b) The green edge is considered a foreground edge because there is no intersection between the green triangle and the polygon. In contrast, the red edge is not considered a foreground edge because there is an intersection between the red triangle and the polygon.

whether there is an intersection between this triangle and the object polygon. If no intersection is detected (the green triangle), the edge is considered to be in the foreground. Conversely, if an intersection is detected (the red triangle), the edge is not considered to be in the foreground. To compute this intersection, we employ the Weiler-Atherton clipping algorithm [18], which effectively clips the polygon using the triangle, generating a new polygon that represents the intersection region.

Establishing local reference frames for edges : To elucidate the process of generating bounding boxes, our methodology incorporates specific conditions. Firstly, we assume a consistent clockwise arrangement of vertices for the polygon, denoted as $O_t = \{s_i, i = 1, ..., n\}$ at time t. Simultaneously, we ensure that the vertices b_i of the resulting bounding boxes, referred to as $B_o = \{b_1, b_2, b_3, b_4\}$, also maintain a clockwise orientation.

After applying the Quickhull algorithm, the polygon vertices are represented in the global reference frame $R_0(o, \overrightarrow{x}, \overrightarrow{y})$ with coordinates (x_{s_i}, y_{s_i}). This global frame serves as the reference coordinate system for the semantic map and is crucial for the robot's localization and map updates.

Our association method entails the establishment of a local reference frame for each edge of the polygon. This frame serves the purpose of identifying can-



(a) Illustration of the local reference defined for the first edge

(b) Creation of the shifted left bounding box

(c) Creation of the shifted right bounding box

Fig. 6. Example of creating the predefined bounding boxes for the first edge

didate bounding boxes for the edge within its local context and defining the association features required for score computation. Subsequently, the box coordinates are transformed into the global frame to enable map updates. In what follows, we provide a detailed explanation of how the local frame is defined for each edge and the necessary transformations for transitioning between the global frame and the local frame, as well as the reverse transition.

For each edge, denoted as $e_i = \{s_i, s_{i+1}\}$ and represented by the two points s_i and s_{i+1}, we establish a local reference frame $R_i(s_i, \overrightarrow{u}, \overrightarrow{v})$ as shown in Figure 6.a. This local reference frame at point s_i is determined by two essential vectors: $\overrightarrow{u}$, which is a unit vector aligned with the direction of e_i, and $\overrightarrow{v}$, a unit normal vector to e_i pointing inward toward the polygon. Together, the pair of vectors $(\overrightarrow{u}, \overrightarrow{v})$ forms an orthonormal basis for the local reference frame at point s_i.

For $\overrightarrow{u}$, we express it as :

$$\overrightarrow{u} = \begin{bmatrix} \Delta_x & \Delta_y \end{bmatrix}^T \tag{1}$$

Here, Δ_x and Δ_y are determined as :

$$\Delta_x = \frac{x_{s_{i+1}} - x_{s_i}}{\|e_i\|} \qquad \Delta_y = \frac{y_{s_{i+1}} - y_{s_i}}{\|e_i\|}$$

where $\|e_i\|$ represents the norm of the edge. To derive $\overrightarrow{v}$, we apply a rotation of $-\pi/2$ to $\overrightarrow{u}$ using the rotation matrix R_θ :

$$\overrightarrow{v} = R_{\theta=-\pi/2}\overrightarrow{u} = \begin{bmatrix} \Delta_y & -\Delta_x \end{bmatrix}^T \tag{2}$$

Now, Eq. 3 is employed to establish the global coordinates (x_b, y_b) in the reference frame R_0 for a given point b based on local coordinates (α, β), as depicted in Figure 6.a :

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} x_{s_i} \\ y_{s_i} \end{bmatrix} + \begin{bmatrix} \Delta_x & \Delta_y \\ \Delta_y & -\Delta_x \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3}$$

Furthermore, the inverse transformation for Eq. 3 can be obtained as follows :

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = A^{-1}B \;\; with \;\; A = \begin{bmatrix} \Delta_x & \Delta_y \\ \Delta_y & -\Delta_x \end{bmatrix} \;\; and \;\; B = \begin{bmatrix} x_b - x_{s_i} \\ y_b - y_{s_i} \end{bmatrix} \tag{4}$$

Here, the inverse of matrix A, denoted as A^{-1}, is given by :

$$A^{-1} = \frac{1}{det(A)} \begin{bmatrix} -\Delta_x & -\Delta_y \\ -\Delta_y & \Delta_x \end{bmatrix} \tag{5}$$

With $\det(A) = -1$, we can simplify the inverse transformation :

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \Delta_x & \Delta_y \\ \Delta_y & -\Delta_x \end{bmatrix} \begin{bmatrix} x_b - x_{s_i} \\ y_b - y_{s_i} \end{bmatrix} \tag{6}$$

Given Eq. 3 and Eq. 6, we have the ability to either transform the global polygon coordinates into local coordinates for the purpose of generating bounding boxes or utilize the inverse transformation to update the map.

Candidate bounding boxes generation: Now that we have all the essential components in place, we outline the procedure for generating bounding boxes, a fundamental aspect of our solution. The steps for creating shifted bounding boxes within the global reference frame R_0 are as follows:

1. Generation of leftward shifted bounding box:

- For a leftward shift (Figure 6.b), the Shift Distance (d) along the u-Axis is set to α_{max}, representing the α value of the extremum right vertex. The identification of the right extremum vertex involves calculating local coordinates (α, β) for all polygon vertices using Eq. 6. Subsequently, the vertex with the maximum α value is selected.
- Further, determining the coordinates of b_2 involves aligning the bounding box edge b_{12} with e_i while ensuring that the edge b_{23} passes from the extremum right vertex with α value equal to α_{max}. Applying Eq. 7 gives the coordinates of b_2.

$$\begin{bmatrix} x_{b_2} \\ y_{b_2} \end{bmatrix} = \begin{bmatrix} x_{s_i} \\ y_{s_i} \end{bmatrix} + \alpha_{max} \ \overrightarrow{u} \tag{7}$$

- After determining the coordinates of b_2, Eq. 8 is applied to compute coordinates for the remaining bounding box vertices.

$$\begin{bmatrix} x_{b_m} \\ y_{b_m} \end{bmatrix} = \begin{bmatrix} x_{b_2} \\ y_{b_2} \end{bmatrix} + a\overrightarrow{u} + b\overrightarrow{v} \tag{8}$$

  For $m = 1$, $a = -l$ and $b = 0$; for $m = 3$, $a = 0$ and $b = -w$; for $m = 4$, $a = -l$ and $b = -w$.

2. Generation of rightward shifted bounding box:
   - Conversely, for a rightward shift (Figure 6.c), the Shift Distance (d) is set to α_{min}, denoting the α of the extremum left vertex.
   - Further, determining the coordinates of b_1 involves aligning the bounding box edge b_{12} with e_i while ensuring that the edge b_{14} passes from the extremum left vertex with α value equal to α_{min}. Applying Eq. 9 gives the coordinates of b_1.

$$\begin{bmatrix} x_{b_1} \\ y_{b_1} \end{bmatrix} = \begin{bmatrix} x_{s_i} \\ y_{s_i} \end{bmatrix} + \alpha_{min} \ \overrightarrow{u} \tag{9}$$

   - After determining the coordinates of b_1, Eq. 10 is applied to compute coordinates for the remaining bounding box vertices.

$$\begin{bmatrix} x_{b_m} \\ y_{b_m} \end{bmatrix} = \begin{bmatrix} x_{b_1} \\ y_{b_1} \end{bmatrix} + a\overrightarrow{u} + b\overrightarrow{v} \tag{10}$$

     For $m = 2$, $a = l$ and $b = 0$; for $m = 3$, $a = l$ and $b = w$; for $m = 4$, $a = 0$ and $b = -w$.

As previously mentioned, this process is executed twice for each edge. Initially, it is performed using a bounding box with dimensions $\{l, w\}$, where $\|b_{12}\| = l$ and $\|b_{23}\| = w$, resulting in the generation of two candidate bounding boxes for the edge. Subsequently, the process is repeated with the bounding box having its dimensions flipped to $\{w, l\}$, where $\|b_{12}\| = w$ and $\|b_{23}\| = l$. At this stage, two distinct candidate bounding boxes, differing in terms of both position and orientation, are generated. This approach allows for the consideration of a broader range of potential candidates, ultimately enhancing the final selection.

Association score computation : In order to select the best bounding box for an object representation, a scoring function is calculated for each generated bounding box to assess the quality of the association. The scoring function, named $\mathcal{S} \in [0, 1]$, provides an indication of the deviation between the polygon and the box. The boxes with scores higher than a specified threshold are kept, and the one with the highest score is selected as the final representation of the object. The selection threshold ϵ is determined through testing.

The computation of the $\mathcal{S}$ is performed using the following equation:

$$\mathcal{S} = 1 - (w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3) \tag{11}$$

where f_1, f_2 and f_3 are the features that are considered in evaluating the association and w_1, w_2 and w_3 are the corresponding weights, with the constraint $w_1 + w_2 + w_3 = 1$.

Definition of the f_1 equation: The f_1 equation quantifies the quality of the alignment between the created bounding box and its associated polygon. Since bounding boxes are inherently rectangular, they inherently possess angles of 90°. The primary objective of this equation is to minimize its value when the bounding box aligns perfectly with the polygon at a 90° angle.

To achieve this alignment, two conditions must be met. First, as we shift the bounding box to the right, α_{min} should approach zero, signifying that the bounding box aligns with the leftmost edge of the polygon. Second, as we shift the bounding box to the left, $\alpha_{max} - \|e_i\|$ should tend to zero, indicating alignment with the rightmost edge of the polygon. These conditions ensure that the bounding box aligns precisely with the orientation of the polygon, achieving the desired 90° orientation.

The f_1 feature is calculated using the following equation:

$$f_1 = \frac{|\alpha_i| - i \cdot \|e_i\|}{l}$$

In this equation, when performing a right shift, i is set to 0, and α_i corresponds to α_{min}. For a left shift, i is set to 1, and α_i corresponds to α_{max}. The division of the error distance by l serves to normalize the f_1 value within the range of 0 to 1, allowing for meaningful comparisons and assessments of alignment.

Definition of the f_2 equation: The f_2 equation quantifies the offset distance between the length of the bounding box and that of its associated polygon. In our analysis, we define the length of the polygon as the distance between its extreme vertices in the local frame R_i, measured along the direction of the edge used for association, and this length is represented as $|\alpha_{min}| + \alpha_{max}$. The primary objective of f_2 is to minimize the error between the real length (l) and the polygon length ($|\alpha_{min}| + \alpha_{max}$).

The f_2 feature is calculated using the following equation:

$$f_2 = \frac{l - (|\alpha_{min}| + \alpha_{max})}{l}$$

Similar to f_1, the division of the error distance by l serves to normalize the f_2 value within the range of 0 to 1.

Definition of the f_3 equation: The f_3 equation quantifies the offset distance between the width of the bounding box and that of its associated polygon. More precisely, we define the width of the bounding box as the distance from the edge used for association to the lowest vertex in the local frame R_i. This distance is equal to β_{max}, representing the coordinate of the lowest vertex in the local frame R_i. The primary objective of f_3 is to minimize the error between the real width (w) of the object and the width of the polygon (β_{max}).

The f_3 feature is calculated using the following equation:

$$f_3 = \frac{w - \beta_{max}}{w}$$

Similar to f_1 and f_2, normalizing the f_3 value by dividing the error distance by w ensures that it falls within the range of 0 to 1.

The score $\mathcal{S}$ is only calculated for boxes that meet two conditions:

$$\begin{cases} |\alpha_i| + (1 - i) \cdot \|e_i\| < l + \psi \\ \beta_{max} < w + \psi \end{cases}$$

where ψ represents a predefined constant to take into account the scaling errors stemming from observation inaccuracies. These two conditions serve to filter out bounding boxes with less relevant characteristics, particularly when the length or width of the bounding box exceeds the actual dimensions of the object it represents. The inclusion of ψ ensures that bounding boxes are not discarded due to scaling errors caused by observation inaccuracies, thus preserving their relevance for analysis.

Regarding the weights, there are many methods to tune them. For the first version of our solution, we propose a method of manually determining them by tuning parameters according to some constraints, testing, and retuning the parameters that give the best results (Section 3.2).

3 Experimentation

We evaluated the performance of our semantic mapping approach against Dengler *et al.* open-source RGBD-based approach [13]. We selected this comparative framework due to its close alignment with our own approach. Dengler *et al.*'s method, like ours, leverages RGBD data to construct point clouds of objects and employs the Quickhull algorithm to define object occupation zones. However, our approach introduces a novel association solution that enriches object representation through predefined bounding boxes. Therefore, this state-of-the-art method serves as an ideal reference point to accentuate the distinctive contributions of our work.

To ensure a fair comparison, we used the same pre-trained detection model as [13], namely Faster R-CNN [19], trained on the OpenImages dataset [20],

encompassing over 600 common object categories found in home and office environments. Our solution's modularity allows for easy substitution of the detection model, as it outputs object categories and detection bounding boxes.

In what follows, we present our experimental setup, highlight the importance of polygon simplification and foreground edge selection solutions to reduce association processing time. Then, we compare the semantic maps obtained by the two approaches.

3.1 Experimental setup

We conducted experiments to evaluate our approach using a computer equipped with an i7-7700K CPU. Our approach assumes that the robot pose is provided, so we generated a metric map using the ROS gmapping node and used the robot ground truth pose for localization. The experiments were performed in two different environments using the MiR100[4] mobile robot with an Asus Xtion Pro camera that had a resolution of 640x480 pixels. The camera was placed at a height of 1.0 m above the robot and 5 degree pitch angle facing the ground.

We evaluated our solution's performance through tests in simulated environments. Since the association method operates downstream of the mapping process and does not directly depend on low-level sensor data, this type of testing allowed us to assess our solution across various contexts and collect diverse performance metrics. We created three simulated office environments. In this initial solution version, we established a knowledge base comprising four object models: a chair, a table, a shelf, and a sofa bed. Subsequently, we populated all three environments with multiple instances of these four object models. One of these environments served as the test environment, where we exhaustively defined the hyper-parameters as described in Section 3.2. We then used these hyper-parameters to obtain validation results in the other two distinct environments (Tables 2 and 3). The first validation environment covers an area of approximately 100 m^2 and contains spaced objects along with some partially hidden objects. The second environment shares the same surface area but features a higher object density, with many objects concentrated in the middle and some hidden, making it a more challenging.

3.2 Tuning the scoring function weights

To determine the appropriate weights for our scoring function $\mathcal{S}$, we initiated the process by defining a range of values [0, 1] with a step size of 0.1 for each weight. Subsequently, we conducted an exhaustive grid search to identify the optimal combination of weights. This involved testing numerous combinations, and the one yielding the best results was selected. Given the time-intensive nature of this process, we implemented optimizations inspired by the underlying principles of our method.

[4] MiR ROS packages: https://www.github.com/dfki-ric/mir _robot/tree/melodic

Table 1. An illustration of the evolution, on the basis of two sequences, of the total number of edges used for association after polygon simplification and foreground edges selection, as well as the average association time.

Environment	Total number of edges			Average association time per polygon (s)
	Initial polygons [13]	After polygon simplification	After foreground edges selection	
Spaced	12060	4137	**1369**	**0.0037 (0.001)**
Cluttered	25699	6813	**2435**	**0.0039 (0.002)**

One key insight guiding our weight selection was the importance of minimizing the offset angle of association (f_1), as this parameter directly influences the position of the bounding box. Consequently, a higher weight was assigned to this parameter. On the other hand, the errors associated with the width (f_3) and the length (f_2) have approximately the same effect on determining the orientation of the bounding box, leading to close weight values for these features.

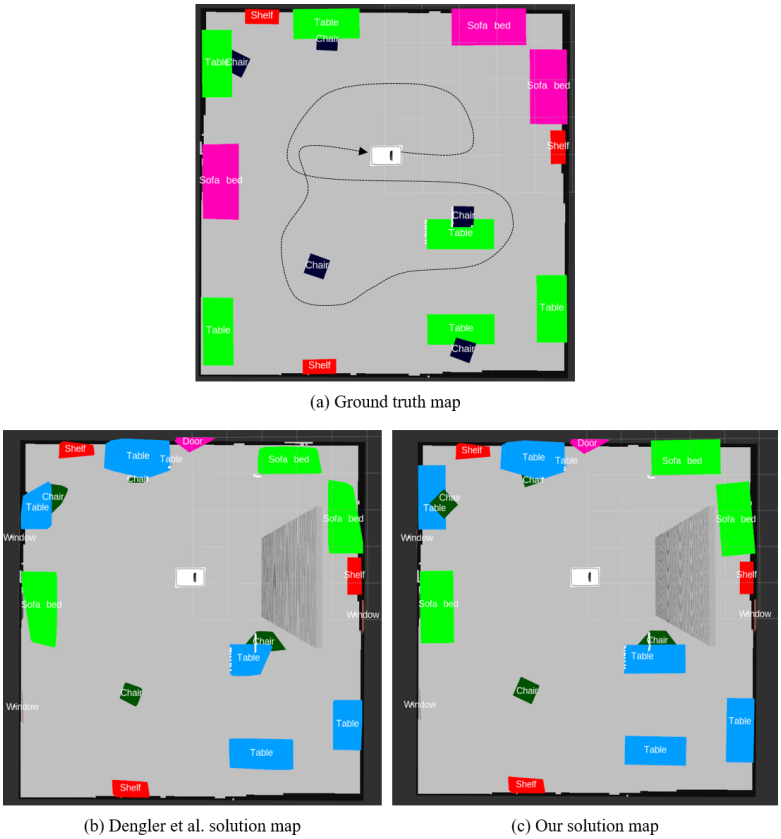
To streamline the weight selection process, we first fixed w_2 and w_3 to values within the range [0, 0.5], while varying w_1 within the range [0.5, 1]. Subsequently, we adjusted the values of w_1 and w_2 to fine-tune the scoring function. Regarding the selection threshold, we systematically tested values within the interval [0 to 0.9] with a step size of 0.1 and retained the value that produced the best results. The most efficient score function $\mathcal{S}$ was obtained for the set of weights $w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$. The scaling error constant was set to $\psi = 0.1$m, and the selection threshold was set to $\epsilon = 0.6$.

3.3 Evaluation of polygon simplification and foreground edges selection

The number of polygon edges generated by the Quickhull algorithm in Dengler *et al.* is significant. As described in Section 2.4, we introduced polygon simplification and foreground edges selection processes to speed up the association processing time. Tab. 1 illustrates the evolution of the total number of polygon edges after the introduction of these two pre-association steps. The results obtained for a sequence in a cluttered environment and a sequence in a spaced environment show that only 10% of the total number of edges are retained for association, thus the association processing time is reduced by about 10 times compared to using the initial polygons. The last row of Tab. 1 shows that the association process takes about 4 ms, including the pre-association steps. Since the number of polygon edges varies from object to object, this value is an average of the association time of all polygons processed per sequence.

3.4 Evaluation of the association algorithm

We conducted 12 mapping sequences in each validation environment, and for each environment, we defined the waypoints for the robot to follow. The trajectory of the robot varies from one sequence to another, in order to provide



(a) Ground truth map

(b) Dengler et al. solution map

(c) Our solution map

Fig. 7. (a) Visualization of the ground truth map, (b) the resulting semantic map from the Dengler *et al.* approach [13], and (c) from our approach.

a performance that is agnostic to the viewpoints of the different objects in the scene. Figure 7.a illustrates the path followed by the robot during the mapping process in the first environment.

We calculated the average metrics for each object class relative to the ground truth map for each sequence in both environments. Subsequently, we averaged the metrics per object over all sequences in each environment. We utilized the following metrics for our evaluation:

- Intersection over Union (IoU), which measures the overall similarity between two shapes.
- The 2D Center of Mass (CoM) offset, which measures the distance between the center of mass of two shapes and provides information about the magnitude of the object displacement.
- True Positives (TP), which indicate the number of correctly mapped objects for which the IoU relative to ground truth is greater than 0.2.

Table 2. Mapping results for the spaced environment (TP : True Positives, IoU : Intersection over Union, CoM : Center of Mass)

Metrics	TP	IoU		CoM offset (m)	
Class		Our solution	Dengler solution [13]	Our solution	Dengler solution [13]
Chair	48	**0.8216 (0.04)**	0.6559 (0.07)	**0.0455 (0.02)**	0.0782 (0.02)
Table	58	**0.8825 (0.03)**	0.7521 (0.05)	**0.0672 (0.02)**	0.1799 (0.05)
Shelf	12	0.6477 (0.10)	**0.7044 (0.10)**	0.1030 (0.03)	**0.0914 (0.05)**
Sofa bed	24	**0.8241 (0.04)**	0.6709 (0.04)	**0.1078 (0.02)**	0.1770 (0.04)
Total	142	**0.7940 (0.08)**	0.6958 (0.03)	**0.0809 (0.02)**	0.1316 (0.04)

Table 3. Mapping results for the cluttered environment (TP : True Positives, IoU : Intersection over Union, CoM : Center of Mass)

Metrics	TP	IoU		CoM offset (m)	
Class		Our solution	Dengler solution [13]	Our solution	Dengler solution [13]
Chair	48	**0.6435 (0.07)**	0.5510 (0.09)	**0.0731 (0.01)**	0.1046 (0,02)
Table	59	**0.8134 (0.04)**	0.6410 (0.04)	**0.1000 (0.04)**	0.2603 (0.04)
Shelf	37	0.6819 (0.06)	**0.7211 (0.06)**	0.0964 (0.02)	**0.0574 (0.03)**
Sofa bed	34	**0.8669 (0.04)**	0.7708 (0.03)	**0.0778 (0.03)**	0.1001 (0.01)
Total	178	**0.7514 (0.09)**	0.6710 (0.08)	**0.0868 (0.01)**	0.1306 (0.07)

While the False Positive (FP) metric, which represents falsely mapped instances, is generally used as a complement to TP to evaluate the accuracy of the detection model, we are only interested in evaluating the association method on correctly mapped objects. Therefore, we compute only TP to show the number of objects considered when computing the average IoU and CoM offset metrics.

The average results over the 12 sequences for each environment are presented in Tab. 2 and Tab. 3. These results were obtained after approximately 54 minutes of mapping, or about 2 minutes / sequence for the spaced environment and 2 minutes and 30 seconds / sequence for the cluttered environment. A total of 142 objects were mapped in the spaced environment (compared to 52 in [13]) and 178 objects in the cluttered environment (compared to 68 in [13]).

The results depicted in Tab. 2 shows that our approach incorporating the augmentation step outperforms Dengler *et al.* approach for all objects, except for the shelf, where the results are almost equivalent. Our solution well performs in mapping large objects, such as tables or sofa beds, with large unseen parts, leading to significant improvements in both the average IoU and average CoM offset. We noticed also an enhancement in the shape of foreground objects, like chairs in this environment. Our approach also performs well for the shelf class, but it shows relatively inferior results compared to other objects, similar to Dengler *et al.* solution. This can be explained by the fact that although the orientation of the box was correctly estimated, the offset side was not selected accurately in some cases. Since the object is small, this offset has a substantial influence on the average IoU value.

Similarly, Tab. 3 shows that our approach performs better for all objects, except the shelf, for the same reason mentioned above. In this setting, there are

more partially invisible tables due to the chairs positioned in the foreground, and we can observe that Dengler *et al.* solution performance declines, while our approach still performs well, especially for the table class. Moreover, our approach almost systematically reduces the standard deviation for both environments, and is therefore more stable.

4 Conclusion and perspectives

In this paper, a 2D semantic mapping approach is presented, designed for mobile robots equipped with RGBD cameras. The method leverages RGBD camera data to initially construct a primitive representation of objects using convex polygons. Subsequently, prior object dimensions obtained from their 3D models are incorporated to enhance this representation. These known dimensions are used to predefine rectangular bounding boxes that accurately cover the real occupied surfaces of the objects. An association method is then introduced to define the best alignment, including correct orientation and position, between these bounding boxes and the polygonal representations of the objects. This approach differs from prior works that solely relied on sensor data for this purpose.

A comparative analysis of our method against the approach presented in [13], conducted in two distinct office settings, demonstrates several notable advantages. Firstly, our solution significantly reduces the complexity of the polygonal representation. Secondly, the use of predefined bounding boxes for object representation significantly enhances the approximation quality for nearly all objects, particularly those that are partially visible or occluded.

Several directions for further improvement of our method can be explored. Firstly, expanding the knowledge base with objects of more complex geometries will allow us to assess how our solution performs in response to such challenges. Additionally, addressing issues arising from limitations in the detection model, such as class confusion or false detections, is a priority for future work. These challenges will be tackled by enriching the knowledge base with additional information about object relationships, including the possibility of object superposition. This supplementary data will be employed to rectify contextual inconsistencies in the map, either in real-time or during post-processing. Lastly, we intend to enhance the method's capability to handle various object models linked to the same label or category.

5 Acknowledgment

This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The version of Record of this contribution is published in Robotics, Computer Vision and Intelligent Systems, and is available online at `https://doi.org/10.1007/978-3-031-59057-3_8`.

References

1. A. Achour, H. Al-Assaad, Y. Dupuis, and M. El Zaher, "Collaborative mobile robotics for semantic mapping: A survey," *Applied Sciences*, vol. 12, no. 20, p. 10316, 2022.
2. J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, "Semantic information for robot navigation: A survey," *Applied Sciences*, vol. 10, no. 2, p. 497, 2020.
3. W. Liu, A. Daruna, M. Patel, K. Ramachandruni, and S. Chernova, "A survey of semantic reasoning frameworks for robotic systems," *Robotics and Autonomous Systems*, vol. 159, p. 104294, 2023.
4. R. Stark, C. Fresemann, and K. Lindow, "Development and operation of digital twins for technical systems and services," *CIRP Annals*, vol. 68, no. 1, pp. 129–132, 2019.
5. W. Purcell and T. Neubauer, "Digital twins in agriculture: A state-of-the-art review," *Smart Agricultural Technology*, vol. 3, p. 100094, 2023.
6. C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," *Agricultural Systems*, vol. 189, p. 103046, 2021.
7. M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3d object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, 2019.
8. N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5079–5085, 2017.
9. T. Zaenker, F. Verdoja, and V. Kyrki, "Hypermap mapping framework and its application to autonomous semantic exploration," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 133–139, 2020.
10. C. Zhao, W. Mei, and W. Pan, "Building a grid-semantic map for the navigation of service robots through human–robot interaction," *Digital Communications and Networks*, vol. 1, no. 4, pp. 253–266, 2015.
11. X. Qi, W. Wang, M. Yuan, Y. Wang, M. Li, L. Xue, and Y. Sun, "Building semantic grid maps for domestic robot navigation," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, 2020.
12. C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.
13. N. Dengler, T. Zaenker, F. Verdoja, and M. Bennewitz, "Online object-oriented semantic mapping and map updating with modular representations," *CoRR*, vol. abs/2011.06895, 2020.
14. A. C. Hernandez, C. Gomez, R. Barber, and O. M. Mozos, "Exploiting the confusions of semantic places to improve service robotic tasks in indoor environments," *Robotics and Autonomous Systems*, vol. 159, p. 104290, 2023.
15. M. Hiller, C. Qiu, F. Particke, C. Hofmann, and J. Thielecke, "Learning topometric semantic maps from occupancy grids," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4190–4197, IEEE, 2019.
16. B. Kaleci, K. Turgut, and H. Dutagaci, "2dlasernet: A deep learning architecture on 2d laser scans for semantic classification of mobile robot locations," *Engineering Science and Technology, an International Journal*, vol. 28, p. 101027, 2022.
17. L. F. Posada, A. Velasquez-Lopez, F. Hoffmann, and T. Bertram, "Semantic mapping with omnidirectional vision," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1901–1907, IEEE, 2018.

18. K. Weiler and P. Atherton, "Hidden surface removal using polygon area sorting," *ACM SIGGRAPH computer graphics*, vol. 11, no. 2, pp. 214–222, 1977.
19. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
20. I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, *et al.*, "Openimages: A public dataset for large-scale multi-label and multi-class image classification," *Dataset available from https://github. com/openimages*, vol. 2, no. 3, p. 18, 2017.