Project Number 101017258

D2.6 Experience-Based Planning and Collaborative Intelligence of MRS

Version 1.0
30 June 2023
Final

Public Distribution

Bonn-Rhein-Sieg University

Project Partners: Aero41, ATB, AVL, Bonn-Rhein-Sieg University, Cyprus Civil Defence, Domaine Kox, FORTH, Fraunhofer IESE, KIOS, KUKA Assembly & Test, Locomotec, Luxsense, The Open Group, Technology Transfer Systems, University of Hull, University of Luxembourg, University of York

Project Partner Contact Information

Aero41 Frédéric Hemmeler Chemin de Mornex 3 1003 Lausanne Switzerland E-mail: frederic.hemmeler@aero41.ch	**ATB** Sebastian Scholze Wiener Strasse 1 28359 Bremen Germany E-mail: scholze@atb-bremen.de
AVL Martin Weinzerl Hans-List-Platz 1 8020 Graz Austria E-mail: martin.weinzerl@avl.com	**Bonn-Rhein-Sieg University** Nico Hochgeschwender Grantham-Allee 20 53757 Sankt Augustin Germany E-mail: nico.hochgeschwender@h-brs.de
Cyprus Civil Defence Eftychia Stokkou Cyprus Ministry of Interior 1453 Lefkosia Cyprus E-mail: estokkou@cd.moi.gov.cy	**Domaine Kox** Corinne Kox 6 Rue des Prés 5561 Remich Luxembourg E-mail: corinne@domainekox.lu
FORTH Sotiris Ioannidis N Plastira Str 100 70013 Heraklion Greece E-mail: sotiris@ics.forth.gr	**Fraunhofer IESE** Daniel Schneider Fraunhofer-Platz 1 67663 Kaiserslautern Germany E-mail: daniel.schneider@iese.fraunhofer.de
KIOS Panayiotis Kolios 1 Panepistimiou Avenue 2109 Aglatzia, Nicosia Cyprus E-mail: kolios.panayiotis@ucy.ac.cy	**KUKA Assembly & Test** Michael Laackmann Uhthoffstrasse 1 28757 Bremen Germany E-mail: michael.laackmann@kuka.com
Locomotec Sebastian Blumenthal Bergiusstrasse 15 86199 Augsburg Germany E-mail: blumenthal@locomotec.com	**Luxsense** Gilles Rock 85-87 Parc d'Activités 8303 Luxembourg Luxembourg E-mail: gilles.rock@luxsense.lu
The Open Group Scott Hansen Rond Point Schuman 6, 5th Floor 1040 Brussels Belgium E-mail: s.hansen@opengroup.org	**Technology Transfer Systems** Paolo Pedrazzoli Via Francesco d'Ovidio, 3 20131 Milano Italy E-mail: pedrazzoli@ttsnetwork.com
University of Hull Yiannis Papadopoulos Cottingham Road Hull HU6 7TQ United Kingdom E-mail: y.i.papadopoulos@hull.ac.uk	**University of Luxembourg** Miguel Olivares Mendez 2 Avenue de l'Universite 4365 Esch-sur-Alzette Luxembourg E-mail: miguel.olivaresmendez@uni.lu
University of York Simos Gerasimou & Nicholas Matragkas Deramore Lane York YO10 5GH United Kingdom E-mail: simos.gerasimou@york.ac.uk nicholas.matragkas@york.ac.uk	

Document Control

Version	Status	Date
0.1	Initial version ready for internal review	26 June 2023
0.2	Address feedback from YORK	29 June 2023
1.0	Final QA version for EC submission	30 June 2023

Table of Contents

Executive Summary

This deliverable reports about the research performed in Task 2.5 of the SESAME project. This work provides the foundation to endow MRS with collaborative intelligence. We consider the ability to share and compose *experiences* of individual robots with other robots as a crucial requirement for collaborative intelligence. We show in this deliverable that by sharing and composing past experiences in the form of recorded heterogeneous environmental maps (e.g., obtained by different mapping approaches) one can perform semantic navigation planning even in the presence of incomplete information or deviations. To this end, we introduce composition operators to compose heterogeneous experiences and to use them for performing context-aware task planning. This deliverable relates primarily to two other deliverable, namely D3.4 and D3.2. For the former the FloorPlan DSL allows to construct maps which can be composed with the approach presented in this work and for the latter the ExSce management informs the annotation of experiences either by developers or robots themselves.

Demonstration video

An accompanying video that illustrates the results of this deliverable interactively can be found at https://youtu.be/5UYgm_Uo5pw

Figure 1: Challenging architectural elements for occupancy grid maps: (a) small floor elevations, (b) low hanging structures, and (c) walls predominantly made of glass panels

1 Introduction

In applications of mobile robots in large environments, such as warehouses and hospitals, robots use a map to navigate around the environment. Typically, a single map is created and used by a robot [17, 14], such that occupancy grid maps [9] are used as a very common representation due to the fact that mobile robot bases are commonly equipped with a laser scanner. While a single map is often sufficient for navigating successfully, there are various cases in which a collection of maps is more suitable or feasible to create. A single-map representation can be challenging in environments with unfavorable architectural elements (such as those illustrated in Fig. 1), infrequent loop-closure opportunities, or access restrictions; in such cases, it may be more appropriate to create and use multiple maps, where each map is created with a sensor that is most appropriate for navigation in a given region. In addition, in the context of multi-robot teams, different robots may also need to create separate maps — potentially using different sensory modalities — such that it may be possible to transfer a map collection to other robots so that they can benefit from the available maps without the need to create their own maps; such knowledge transfer can then be facilitated by collaborative robot platforms [19]. In both of these cases, the mapping process needs to be split over multiple sessions and possibly using varying sensing modalities during each mapping session, which could result in a fragmented[1] and heterogeneous[2] collection of maps.

For the case of homogeneous maps, approaches exist that make it possible to compose fragmented maps so that a robot can use them for navigation [4]. When considering heterogeneous maps, however, *map composition* requires a structured process that makes it possible to specify when such composition is possible, but also to define concrete rules about how available maps should be composed. The first contribution of this deliverable is thus a formal representation of the map composition problem and a concrete composition approach based on a YAML-based domain-specific composition language, which allows performing map composition through a set of predefined composition operators. The proposed composition technique can act as a bridge component between (i) a collection of fragmented, heterogeneous, and potentially incomplete[3] maps (henceforth referred to as a *diverse* collection of maps), and (ii) navigation components that expect a single map of the environment, in a specific map representation, as shown in Fig. 2.

In certain applications, the collection of heterogeneous maps may include information encoded in a semantic map, which enables the use of a semantic navigation system [5]. A semantic map usually contains high-level

[1] *Fragmentation* refers to the process of creating smaller sub-maps that cover an environment partially.

[2] Two maps are considered to be *heterogeneous* if they differ in either the map representation (e.g. an occupancy grid map and a semantic map) or the set of properties associated with the representation (e.g. the occupancy grid cell resolution).

[3] Here, *incompleteness* refers to partial coverage of the environment by any collection of prior *homogeneous* maps.

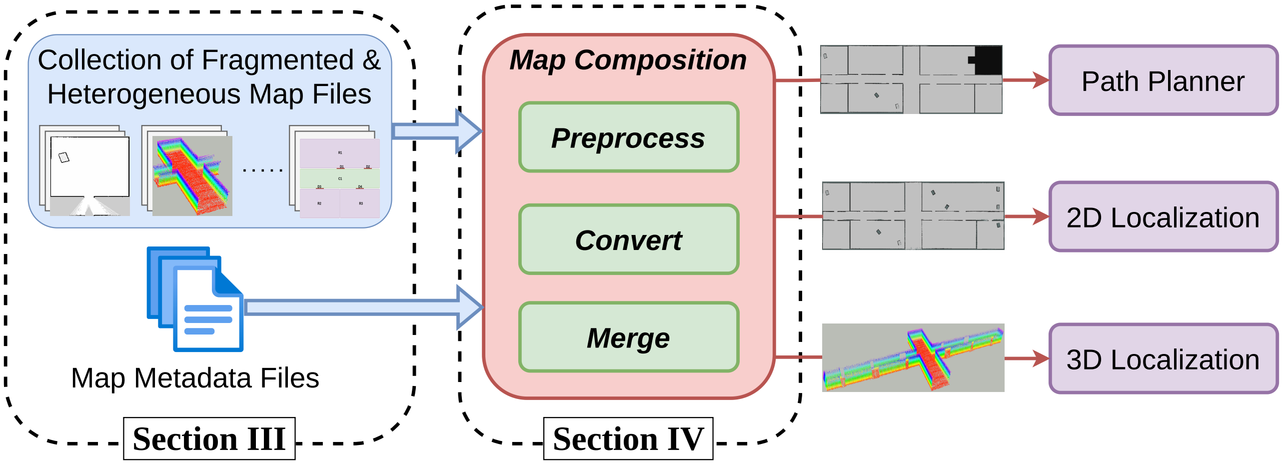


Figure 2: Map composition as a bridge between a collection of maps and standard navigation components, transforming and fusing map data to be usable by the navigation components.

conceptual knowledge, such as the type of indoor spaces (e.g. corridor or doorway) or forbidden regions where robot motion is prohibited. The benefit of semantic maps is that they enable a robot to generate and execute context-aware navigation plans, for instance enforcing speed limits near doorways or triggering robot alerts, in order to improve its efficiency, robustness, safety, and social acceptability; however, most semantic navigation systems expect a complete semantic map of the environment, which is a strong requirement that particularly increases the complexity of robot transfer to new environments. The second contribution of this deliverable is thus a navigation task planner that maximizes the use of any available semantic information to perform context-aware navigation, but that falls back to conventional grid-based path planning in the presence of incomplete semantic maps.

To demonstrate the feasibility of the proposed map composition framework and context-aware navigation task planner, we present a use case analysis for a KELO ROBILE[4] mobile robot platform in a collection of simulated environments based on the university building of Hochschule Bonn-Rhein-Sieg.[5] The analysis of the composition approach shows how maps can be combined at runtime, while the analysis of the navigation task planner illustrates how composed maps can be used for flexible, context-aware robot navigation.

[4] https://www.kelo-robotics.com/products/

[5] We focus on a conceptual analysis of the proposed components, so we opted for a simulation-based evaluation that provides greater flexibility with the investigated environments; however, it should be noted that the interfaces of the simulated robot are close to those of the real KELO ROBILE platform, which simplifies the transfer to the real robot platform for future studies.

2 Related Work

The problem of using existing maps to create new maps has been actively studied in the domain of multi-robot cooperative mapping [1]. Here, individual maps generated by robots belonging to a team are merged to form a globally consistent map. This process involves two tasks: (i) finding correspondences between a pair of maps to compute the relative coordinate transform between them, and (ii) using the transform to fuse the data from the two maps into a single map. As our work allows fragmentation and incompleteness of maps, it cannot be guaranteed that the maps always overlap and correspondences can be computed; for this reason, we assume that the transform between the maps is known a-priori and focus on the fusion of map data instead.

Considerable work in the literature focuses on 2D occupancy grid maps. In [3], a Bayesian method is proposed to fuse two occupancy grid maps using the probabilities associated with the overlapping occupancy map cells; the maps are assumed to be of the same map resolution and accurately transformed with respect to a global reference frame before being merged. A solution that relaxes the same map resolution assumption is provided in [18], while a more general approach that can be used to merge grid maps that differ in their scale and quality is presented in [8]. Existing work in map fusion is, however, predominantly focused on the fusion of maps belonging to the same map representation [4], or that the raw sensory data is available to generate the fused maps. In our work, we do not make an assumption that such data is available, so pre-existing maps are used for fusion instead. To the best of our knowledge, no work has addressed the challenge of fusing heterogeneous maps while being flexible and open to extensions for arbitrary map representations.

An early approach in semantic navigation [7] uses a dual-hierarchical model to represent the spatial and conceptual knowledge about an environment to improve human-robot interaction. This model is merged into a single multi-layered representation in [20] and the conceptual knowledge of indoor spaces is represented using a hand-crafted OWL-DL ontology. [16] uses a probabilistic ontology and an inference engine that can handle uncertainty in the grounding of semantic concepts using noisy sensory information. [15] extends the OpenStreetMap representation to encode robot-specific semantic knowledge about the environment as a graph representation. In this literature, the focus is on the construction of a semantic map rather than its usage. [12] addresses both the problem of building semantic maps and using those for navigation. [2] presents a context-based navigator that uses a simple polygonal semantic map representing indoor spaces. Unlike approaches that expect complete and fully-specified metric and semantic maps, our work proposes a mechanism to utilize a collection of incomplete semantic maps to maximize the coverage of the environment where a robot can perform context-aware navigation.

3 Map Composition

The objective of this work is to define an approach that enables the composition of fragmented and potentially heterogeneous maps so that a robot can benefit from the map collection in navigation tasks. In this section, we present a general formalization of the composition problem and define *composition operators* that make map composition possible. In the following section, we then discuss how the proposed operators are embedded in a YAML-based composition language that facilitates runtime map composition, and address the use of a composed map in conjunction with a context-based task planner in the subsequent section.

3.1 Formal Description

Let $\mathcal{R}$ denote a set of *map representations* (such as an occupancy grid or a semantic map), $\mathbb{M}^r$ be the space of maps of a given map representation $r \in \mathcal{R}$, and $\mathcal{P}^r \in \mathcal{P}$ denote a set of *property values* (e.g. grid resolution or map origin) associated with a representation r. Then, the i^{th} map having a representation r and a property value set $\mathcal{P}_i^r$ is denoted as[6] ${}^r m_i^{\mathcal{P}_i}$, while $\mathcal{M}^r$ denotes a set of n such *prior maps* from the same representation r, but with possibly varying properties $\mathcal{P}_i^r$:

$$\mathcal{M}^r = \left\{ {}^r m_0^{\mathcal{P}_0},\ {}^r m_1^{\mathcal{P}_1},\quad \cdots \quad,\ {}^r m_n^{\mathcal{P}_n} \right\} \tag{1}$$

Here, the term *prior* denotes that these are maps that have been created either manually or autonomously by a robot. In contrast, let $\hat{\mathcal{M}}^r$ denote a set of *composed maps* that are obtained as a result of map composition, namely as a result of combining prior maps and potentially other composed maps. Then, $\mathcal{X} = \bigcup_i \mathcal{M}^{r_i}$ and $\hat{\mathcal{X}} = \bigcup_i \hat{\mathcal{M}}^{r_i}$ denote a set of all prior and composed maps, respectively, while the power set $\mathbb{P} = \mathcal{X} \bigcup \hat{\mathcal{X}}$ denotes the complete collection of maps.

We define *map composition* as the process that generates new maps using the maps in $\mathbb{P}$.

Definition 1. Let $m_1, ...m_k \in \mathbb{P}$ be maps. Then, *map composition* is the process of creating a new joint map $m^* = \cup_{i=1}^k \hat{m}_i$, where $\hat{m}_i$ is potentially a modified version of map m_i.

3.2 Composition Operators

According to Def. 1, the composition of maps is based on various requirements. First of all, composition is only meaningful if maps are of the same type, which implies that it should be possible to convert a map from one type to another. Furthermore, composition may also require modifying the properties of a single map, for instance to change its origin so that it is aligned with that of another map. Finally, given two homogeneous maps, namely maps that are of the same type and which have compatible properties, the generation of a new map from the two maps should be possible. We express these requirements through three map operators that facilitate the composition of maps in $\mathbb{P}$: (i) an operator that preprocesses a map by modifying its properties, (ii) an operator that converts between map representations, and (iii) an operator that combines two maps into a single map. The operators are defined below, such that it should be noted that none of them modify the maps in $\mathbb{P}$ directly, but rather use them to generate new maps that can then be added to $\mathbb{P}$ and used as a source for further composition.

3.2.1 Preprocessing

We first define an operator that changes the properties of a map without changing its type:

[6] For notational simplicity, the redundant superscript r from $\mathcal{P}_i^r$ is dropped when used in the expression of a map instance.

Definition 2. $\texttt{preprocess} : (\mathbb{M}^r, \mathcal{P}^r) \to \mathbb{M}^r$ modifies the property value set of an input map.

Concretely, the *preprocess* operator takes a map ${}^r m_{in}^{\mathcal{P}_{in}}$ and the desired property values $\mathcal{P}_{out}^r$ of the resulting map and produces a new map of type in with properties $\mathcal{P}_{out}^r$:

$$ {}^r m_{out}^{\mathcal{P}_{out}} = \texttt{preprocess}\left({}^r m_{in}^{\mathcal{P}_{in}},\ \mathcal{P}_{out}^r\right) \tag{2} $$

This operator can be used to perform operations such as down-sampling a point cloud, changing the resolution of a grid map, and so forth.

3.2.2 Map conversion

The next operator we define converts a map from one type to another type with a set of defined properties, such that it may be the case that only a subset of the original map is considered during the conversion:

Definition 3. $\texttt{convert} : (\mathbb{M}^{r_1}, \mathcal{R}, \mathcal{P}^{r_1}, \mathcal{P}^{r_2}) \to \mathbb{M}^{r_2}$ converts a map from type r_1 (or a subset thereof as specified by a set of properties) to a map of type r_2 with given properties.

It should be noted that the conversion between two map representations may, in some cases, be infeasible [1], so an implementation of `convert` needs to ensure that the conversion from type r_1 to type r_2 can indeed be performed.

As defined above, the operator produces an output map ${}^{r_{out}} m_{out}^{\mathcal{P}_{out}}$ given an input map ${}^{r_{in}} m_{in}^{\mathcal{P}_{in}}$, a desired map representation r_{out}, a set of parameters $\mathcal{S}^{r_{in}}$ that select a subset of the input map data during the conversion, and the desired property values $\mathcal{P}_{out}^{r_{out}}$ in the resulting map:

$$ {}^{r_{out}} m_{out}^{\mathcal{P}_{out}} = \texttt{convert}\left({}^{r_{in}} m_{in}^{\mathcal{P}_{in}},\ r_{out},\ \mathcal{S}^{r_{in}},\ \mathcal{P}_{out}^{r_{out}}\right) \tag{3} $$

For instance, when converting a point cloud map to an occupancy grid map, $\mathcal{S}^{r_{in}}$ can be used to define a range of height values of the points to be used during the conversion.

3.2.3 Merge operator

The final operator we define fuses the map data from two different *homogeneous* maps into a new map that has the same map representation and property value set as the input maps:

Definition 4. $\texttt{merge} : (\mathbb{M}^r, \mathbb{M}^r) \to \mathbb{M}^r$ creates a combined map of type r given two maps of the type r.

Concretely, the `merge` operator takes as input two maps ${}^r m_{in_1}^{\mathcal{P}}$ and ${}^r m_{in_2}^{\mathcal{P}}$ and produces a map ${}^r m_{out}^{\mathcal{P}}$:

$$ {}^r m_{out}^{\mathcal{P}} = \texttt{merge}\left({}^r m_{in_1}^{\mathcal{P}},\quad {}^r m_{in_2}^{\mathcal{P}}\right) \tag{4} $$

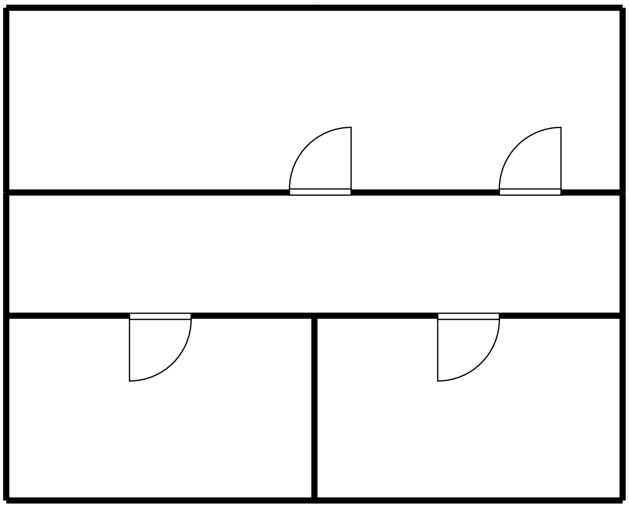
It should be noted that, if two maps m_1 and m_2 are heterogeneous, they must be homogenized by applying the `preprocess` and `convert` operators before merging.

Using these definitions of map operators, we can amend the definition of map composition as follows:

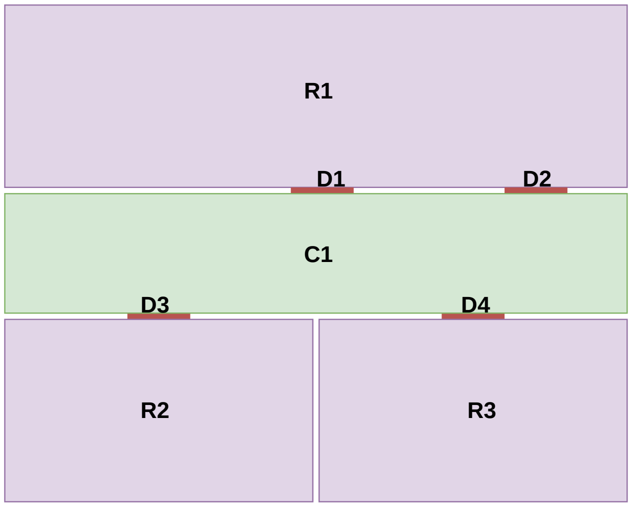
Definition 5. *Map composition* is an ordered and sequential application of one or more map operators

$$(\mathtt{o}_i : \mathtt{o} \in \{\mathtt{preprocess}, \mathtt{convert}, \mathtt{merge}\},\ i \in \mathbb{N})$$

on a collection of maps, to generate a new map with some desired map representation and property value set.



(a)



(b)

Figure 3: Example of the semantic map representation used in this work. (a) The floor plan of a sample environment. (b) The semantic map represented as a set of polygons with a unique name and type. Colors represent the type of space — purple: standard (e.g. room); green: corridor; red: door.

4 Composition Language and Examples

The above section formalizes the composition problem, but does not clarify how we achieve composition in practice. In this section, we will briefly describe the map representations that we consider in this work, discuss a YAML-based language for specifying map composition tasks, and present various examples that illustrate the use of the map composition operators for creating complex maps.

4.1 Map Types

In this work, we use the following map representations to demonstrate the proposed map composition approach:

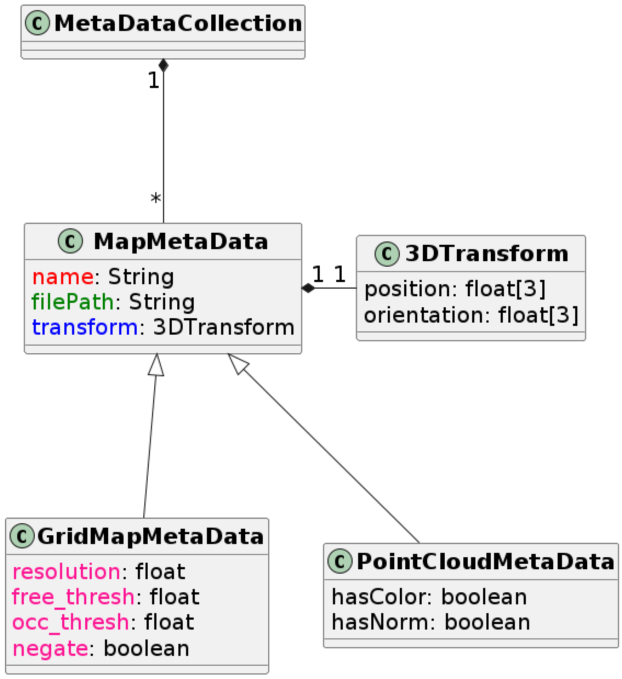
2D occupancy grid [6]: Discretizes the space into a 2D cellular grid and stores the probability of each cell being occupied by some obstacle.

3D point cloud: Stores a set of 3D points representing sub-samples of surfaces of objects in the environment.

Polygonal 2D semantic map: We use a custom representation to include high-level semantic knowledge about the environment, based on the indoor spaces ontology [13]. The representation partitions the environment into a set of adjoining, but non-overlapping spaces, as shown in Fig. 3, such that each space is represented as a 2D polygon that is assigned: (i) a unique name, (ii) an indoor-space type (e.g. corridor), and (iii) an optional set of attributes (e.g. speed limit).

4.2 Map Metadata

For all prior maps of a particular representation, we use a *metadata* file corresponding to that representation. Fig. 4 illustrates the model of a metadata file for an occupancy grid map and shows how this is represented in a YAML-based file. Irrespective of the map representation, the minimal expected metadata for any map includes: (i) a unique map name, (ii) the location of a file in which the map data is stored, and (iii) a coordinate transform that registers the map in a fixed world frame, shared by all prior maps. The transform ensures that



(a)

```yaml
occupancy_grid_maps:
  room1:
    filepath: room1.pgm
    transform:
      position:
        - -26.325
        - 33.25
        - 0.0
      orientation:
        - 0.0
        - 0.0
        - 0.7845
      resolution: 0.05
      free_thresh: 0.196
      occ_thresh: 0.65
      negate: false
```

(b)

Figure 4: The map metadata model. (a) A UML class diagram formalizing the contents of a metadata file. Since our semantic map does not require additional attributes, it is modelled using the `MapMetaData` base class. (b) An example of a metadata file for 2D occupancy grid maps containing a single map and expressed using a YAML-based specification.

all map fragments are correctly aligned with each other upon loading the maps and eliminates the need for performing map alignment, which we do not deal with in this work.

4.3 Recursive Map Composition

To illustrate how the composition operators enable map composition to be performed, we consider an environment for which we are given (i) a semantic map of an office, (ii) a point cloud map of a corridor, and (iii) 2D occupancy grid maps of two rooms, such that the objective is to create a 2D occupancy grid map of the whole environment that a robot can subsequently use for navigation. This composition scenario is illustrated in Fig. 5. Formally, this complex operation can be broken down into an ordered sequence of operator applications as expressed below:

$$ {}^{r_{occ}}m^{\mathcal{P}}_{office} = \mathtt{convert}\left({}^{r_{sem}}m^{\mathcal{P}}_{office},\ occ, \mathcal{S}^{sem}, \mathcal{P}\right) \tag{5} $$

$$ {}^{r_{occ}}m^{\mathcal{P}}_{cor} = \mathtt{convert}\left({}^{r_{cloud}}m^{\mathcal{P}}_{cor},\ occ, \mathcal{S}^{cloud}, \mathcal{P}\right) \tag{6} $$

$$ {}^{r_{occ}}m^{\mathcal{P}}_{loc} = \mathtt{merge}\left({}^{r_{occ}}m^{\mathcal{P}}_{office}, {}^{r_{occ}}m^{\mathcal{P}}_{cor}\right) \tag{7} $$

$$ {}^{r_{occ}}m^{\mathcal{P}}_{loc} = \mathtt{merge}\left({}^{r_{occ}}m^{\mathcal{P}}_{loc}, {}^{r_{occ}}m^{\mathcal{P}}_{room1}\right) \tag{8} $$

$$ {}^{r_{occ}}m^{\mathcal{P}}_{loc} = \mathtt{merge}\left({}^{r_{occ}}m^{\mathcal{P}}_{loc}, {}^{r_{occ}}m^{\mathcal{P}}_{room2}\right) \tag{9} $$

The result of Equation 9 is then the desired map that contains fused data from the four input maps. To specify this operator application declaratively, we propose the use of a YAML-based representation, as illustrated in Fig. 6.

We discuss some details about the concrete implementation of the operators that we used to evaluate the composition procedure in Section 6.

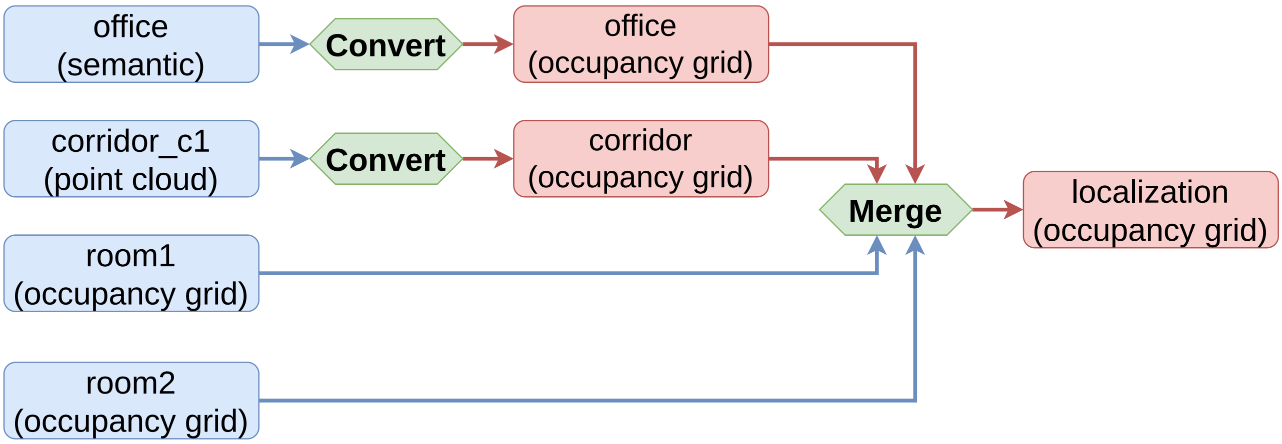


Figure 5: A sample map composition scenario to generate three composed maps (red) using four diverse prior maps (blue) and performing three parameterized map operations (green).

```
map_metadata:
  occupancy_grid: occupancy_grids.yaml
  point_cloud: point_clouds.yaml
  semantic: semantic_maps.yaml

composed_maps:
  - result_name: office
    result_type: occupancy_grid
    operation:
      - convert:
        input_name: office
        input_type: semantic
        config: {gridResolution: 5cm}
  - result_name: corridor
    result_type: occupancy_grid
    operation:
      - convert:
        input_name: corridor_c1
        input_type: point_cloud
        config: {minZ: 0.5m, maxZ: 3.0m}
  - result_name: localization
    result_type: occupancy_grid
    operation:
      - merge: [office, corridor, room1, room2]
```

Figure 6: Map manifest file corresponding to the example shown in Fig. 5, with keywords highlighted in green. The `map_metadata` lists the map metadata file paths, and `composed_maps` contains definitions for the three composed grid maps. Each definition contains: (i) result map description (`result_name`, `result_type`), (ii) operator type, (iii) input map description (`input_name`, `input_type`), and (iv) optional operation configuration parameters `config` ($\mathcal{S}^{r_{in}}$, $\mathcal{P}^{r_{out}}_{out}$). Since the `merge` operator expects *homogeneous* maps, the map representation of the input maps is inferred from the `result_type`.

5 Context-aware Task Planning

The map composition framework presented above can, in principle, be used as a standalone component with most navigation systems.[7] As described in the previous section, however, one of the map types that we consider is a semantic map, which a robot can use to generate contextually appropriate navigation plans, such as slowing down near intersections or navigating carefully around doors. While there are various approaches in the literature that can be used for semantic navigation, existing systems are incapable of exploiting partial semantic information, namely they expect a complete semantic map, which particularly complicates the portability of semantic navigation systems to new environments. In this section, we describe a navigation task planner that generates context-aware navigation plans, but that is robust to incomplete semantic maps.

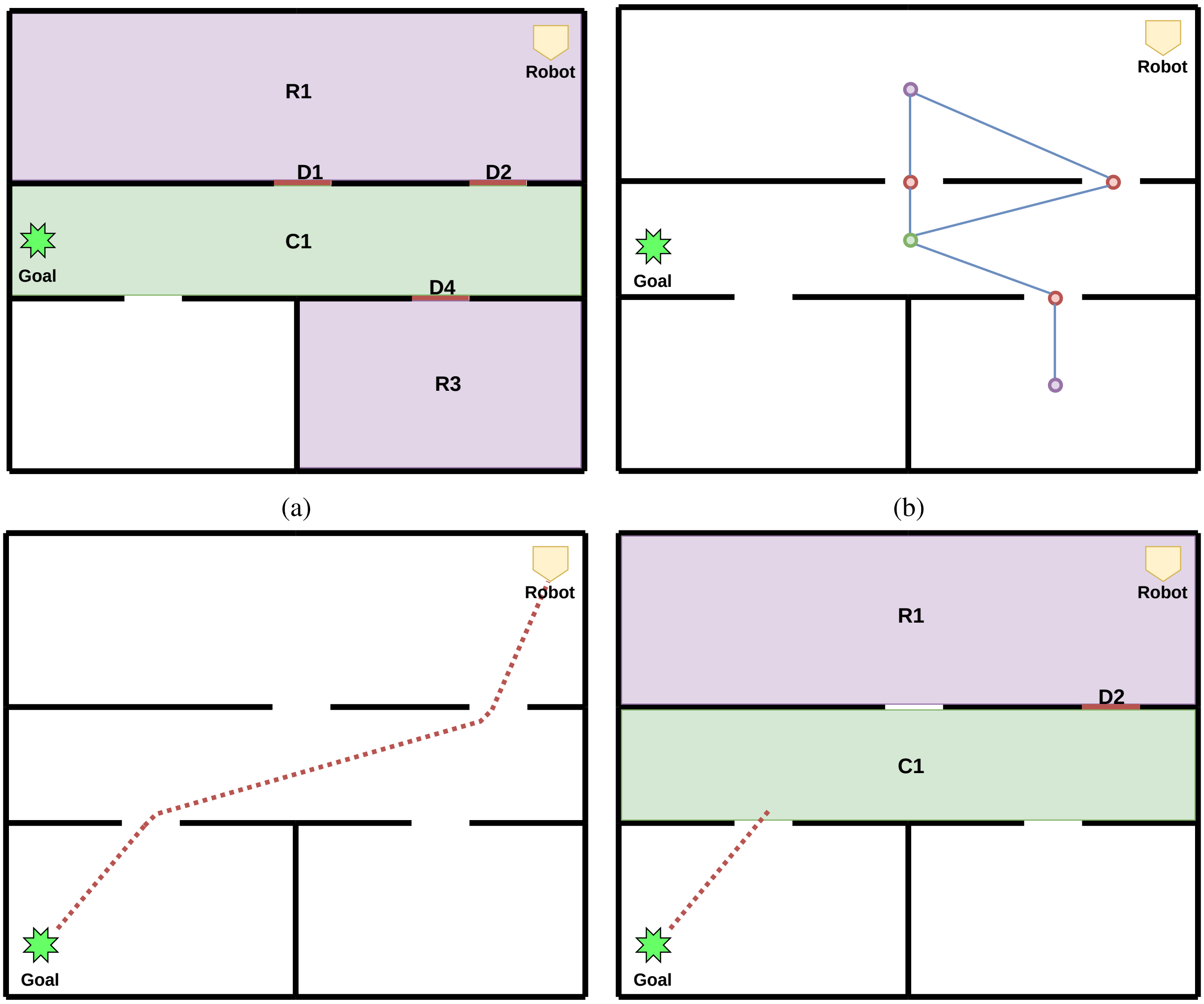
Fig. 7a illustrates a semantic map that partially covers the environment, such that it should be noted that the *incompleteness* of a semantic map is, in principle, relative to a given navigation task. For instance, the semantic map as shown here has all necessary information for solving the navigation task presented in Fig. 7a, but this is not the case for the task presented in Fig. 7c, where the goal lies outside of the semantic map. Based on this observation, our proposed task planner prioritizes the use of a semantic map whenever possible, but falls back to an occupancy grid map whenever semantic information is not available or semantic navigation is not possible. Similar to other semantic navigation systems [12, 15], the planner relies on a topological graph that is generated by (a) representing each indoor space from the semantic map as a node and (b) connecting two adjoint spaces by an edge, as shown in Fig. 7b. A navigation task is then solved in three stages: (i) determining the start and goal topology nodes by identifying the indoor spaces containing the robot's and goal poses, (ii) performing A^* graph search between the start and goal nodes, and (iii) translating the solution into a context-aware plan. Such a plan contains a series of spaces that lead to the goal space, and ultimately to the navigation goal.

There are two cases in which we consider a semantic map to be *incomplete* for a given navigation task: (i) if the start or goal pose is not contained inside any semantically mapped space, or (ii) if the A^* graph search fails to find a solution. If the semantic map is found to be incomplete, our planner falls back to grid-based global path planning using an occupancy grid map and generates a global path plan (given as a sequence of poses) as shown in Fig. 7c.

From Fig. 7a and 7c, it can be seen that a large portion of this path plan overlaps with semantically mapped spaces and hence it would be beneficial to perform context-aware navigation when traversing through those spaces. The path plan is thus post-processed to create a hybrid context-aware plan. The post-processing involves three steps: (i) the semantic map is used to identify the points at which the path plan enters or exits the semantically mapped regions, (ii) the path plan is split at these points to generate a set of sub-paths that alternatively overlap with the semantic and non-semantic regions of the environment, and (iii), for each sub-path that overlaps with the semantic map, the series of spaces traversed by the path are identified and converted into a context-aware plan. The final hybrid navigation plan contains two types of alternating actions, namely context-aware navigation (`goto`), and path following (`follow_path`). We illustrate the hybrid plan in a YAML-based format in Fig. 8; this format can be consumed by a context-aware navigation component, such as [2][8], which we prototypically use in our evaluation.

[7] Such as the ROS navigation stack: http://wiki.ros.org/navigation or the semantic navigation component in [2]

[8] An implementation of this navigation component can be found at https://github.com/DharminB/cabin_nav



(a) (b) (c) (d)

Figure 7: Planner demonstration with an incomplete semantic map. (a) A planning scenario containing robot and goal positions drawn on top of an occupancy grid map (black pixels represents occupied cells, the rest are free cells) and overlaid with a partial semantic map (missing some indoor spaces). (b) Topological representation of the semantic map. (c) For a goal pose outside the semantic map, a global path plan is computed using the grid map. (d) A post-processed hybrid plan consisting of a context-aware navigation action through the colored regions until the boundary of the semantic map, followed by a sub-segment of the global plan to the goal.

```yaml
plan:
  - action_type: goto
    goto_plan:
      - R1: Room
      - D2: Doorway
      - C1: Corridor
    goal: {x: -1.74, y: 0.15, theta: 2.81}

  - action_type: follow_path
    path:
      - {x: -1.74, y:  0.15, theta: 2.81}
                        :
      - {x: -2.50, y: -1.12, theta: 0.61}
```

Figure 8: A hybrid navigation plan containing two actions: (i) context-aware navigation up to the boundary of a semantic map, (ii) path following until the goal, as shown in Fig. 7d.

6 Evaluation

To investigate the feasibility of the proposed map composition and navigation task planning components, we present the results of a qualitative evaluation that we performed in a Gazebo environment[9], illustrated in Fig. 9a, using a simulated KELO ROBILE platform; this environment is based on our university building and includes elements such as those in Fig. 1. For this, we created a diverse collection of 11 prior maps, which are shown in Fig. 9b; we used GMapping [9] and hdl_graph_slam [11] to generate the fragmented occupancy grid and point cloud maps, respectively, while a partial semantic map was created manually. In the figure, it can be observed that none of the three representations used by the prior maps completely cover the environment; they thus serve as an illustration of the map composition framework and the subsequent partial semantic navigation plan. In the evaluation, we first analyze the map composition in terms of the prior maps and demonstrate how they can be composed and used to generate a hybrid navigation plan; we then analyze whether context-aware navigation has any concrete benefits compared to traditional path following.[10]

6.1 Map Composition Analysis

To evaluate the quality and suitability of the composed maps for use in standard navigation components, we used the ROS navigation stack, which requires an occupancy grid map that covers the complete environment. In the considered use case, while the robots can navigate within individual rooms using the fragments represented by the prior occupancy grid maps, it is not possible to achieve inter-room navigation that requires traversing the corridors. For this reason, there is a need for map composition that makes it possible to leverage information from the other existing representations for populating the missing occupancy information.

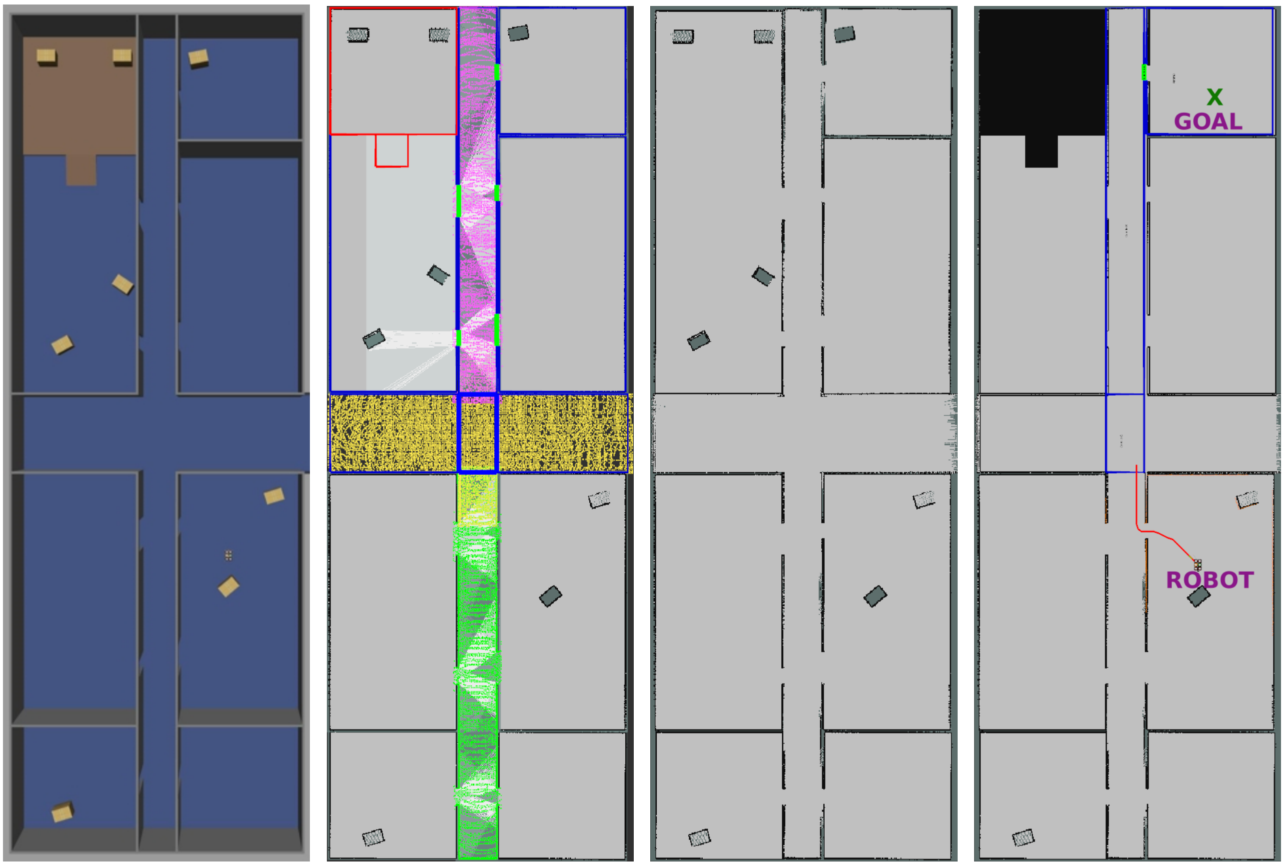
We implemented three versions of the map operators to achieve the desired map composition: (i) a point-cloud-to-occupancy-grid conversion operator, (ii) a semantic-map-to-occupancy-grid conversion operator, and (iii) a merging operator for occupancy grids. We use the OctoMap toolbox [10] to convert a point cloud to an occupancy grid map. The conversion from a semantic map to an occupancy grid is similar to [15], such that it considers the boundaries of areas to be occupied, while everything else is considered to be free space (including doors). For the merging of occupancy grids, each cell is considered as *free*, *occupied*, or *unknown*, such that a combined grid (with a size equal to the bounds of the axis-aligned rectangle encompassing both input maps) is first created and initialized to include only *unknown* cells; all the cell values from the first input map are then filled in the combined cellular grid and, finally, the cell values from the second input map are only filled in the *unknown* cells of the combined cellular grid.[11]

To achieve navigation throughout the environment, we composed two maps: (i) a grid map for localization (shown in Fig. 9c) using the prior occupancy grids for the rooms and doors, and the point cloud maps for the corridors, and (ii) a grid map for path planning (shown in Fig. 9d), which additionally includes information obtained from the semantic map. It can be observed that the elevated floor region and the ramp, which are marked as *forbidden regions* in the semantic map (red polygons in Fig. 9b) are marked as occupied cells in the composed grid map that is used for path planning; this prevents generating any undesired path plans that pass through these regions, thus considering the robot's safety. The ability to prioritize an occupancy grid map (in this case the converted semantic map) by simply changing the merge order can also be observed in the top half of Fig. 9c and 9d. Using the composed maps, we could observe that the robot was able to successfully navigate in all parts of the environment (where the start and goal locations shown in Fig. 9d represent just one example); this demonstrates that the proposed composition method can produce maps that are useful for robot navigation.

[9] https://github.com/kelo-robotics/robile_gazebo

[10] An accompanying video that illustrates the results of this section interactively can be found at https://youtu.be/5UYgm_Uo5pw

[11] It should be noted that this merging policy prioritizes the first map in the merging order, so changing the order of maps can generate different results and thus affect the behavior of a robot using a composed map.



(a) (b) (c) (d)

Figure 9: Map composition results. (a) The simulated environment containing 7 *rooms* with few obstacles, 11 *doorways*, and 4 *corridors* meeting at a *junction*. The brown-colored region on the top left is an elevated region of the floor (as shown in Fig. 1), where the robot is prohibited from traversing. (b) An overlay of all prior maps consisting of 1 occupancy grid per room, 3 point cloud maps (pink, yellow and green points), and 1 partial semantic map (blue/red polygons and green lines) covering only the top half of the environment (including the middle *corridors* and the *junction*). (c) The composed occupancy grid map generated for localization. (d) Visualization of a hybrid plan where a robot follows the red path until it enters the *junction*, and then follows the highlighted semantic spaces until the goal.

6.2 Planning with Incomplete Semantic Maps

To evaluate the proposed navigation task planner and its ability to generate context-aware navigation plans using partial semantic information, we used the available semantic map along with the composed maps that were generated in the above composition evaluation. To execute the context-based plans generated by our planner, we replaced the ROS navigation stack with the semantic navigator proposed in [2], which executes different behaviors depending on the semantic types of the areas in which a robot is navigating.

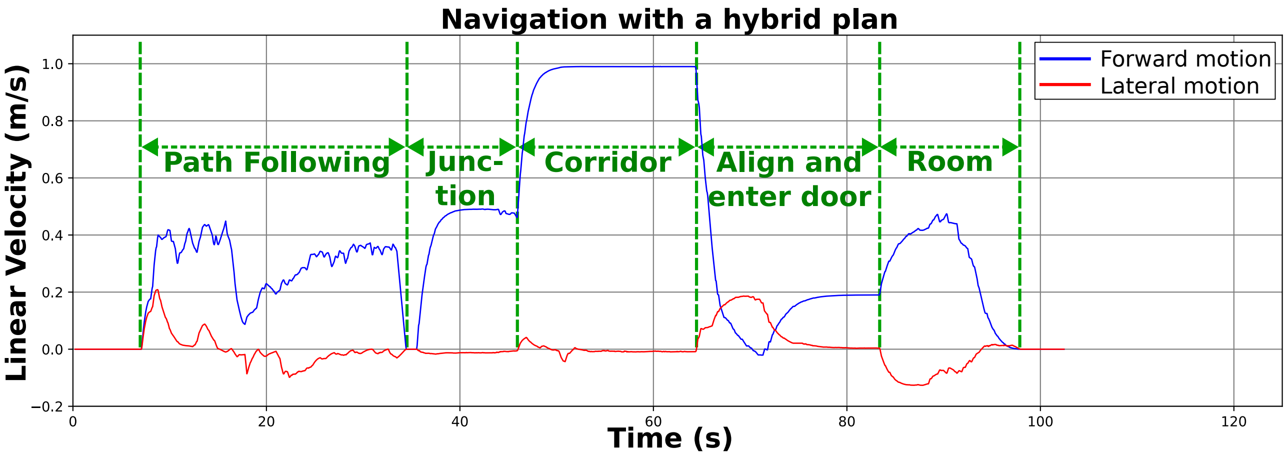
Fig. 9d illustrates a navigation task where the robot is located outside the bounds of the semantic map, such that the corresponding hybrid plan generated by our planner involves path following (illustrated by the red line) until it enters the semantic map, at which point the robot switches to executing pre-configured motion behaviors designed for the type of indoor space being traversed (the parts of the plan in which semantic navigation is performed are illustrated by the blue polygons surrounding the areas). This demonstrates the feasibility of the proposed task planner for creating context-aware navigation plans and executing them with a suitable semantic navigation component.

In addition to investigating the feasibility of executing context-aware plans, we investigated whether context-aware navigation can lead to any quantifiable benefits for a robot, other than making the robot's behavior more suitable for human-centered environments. For this, every motion behavior designed for a specific indoor space type was configured with a maximum allowed speed as shown in Table 1, such that we analyzed the velocity profile of the simulated robot to see whether context-aware navigation can produce more efficient execution. For the `follow_path` action, a speed limit of $0.5m/s$ was experimentally chosen to maximize the chances of passing through narrow doorways.

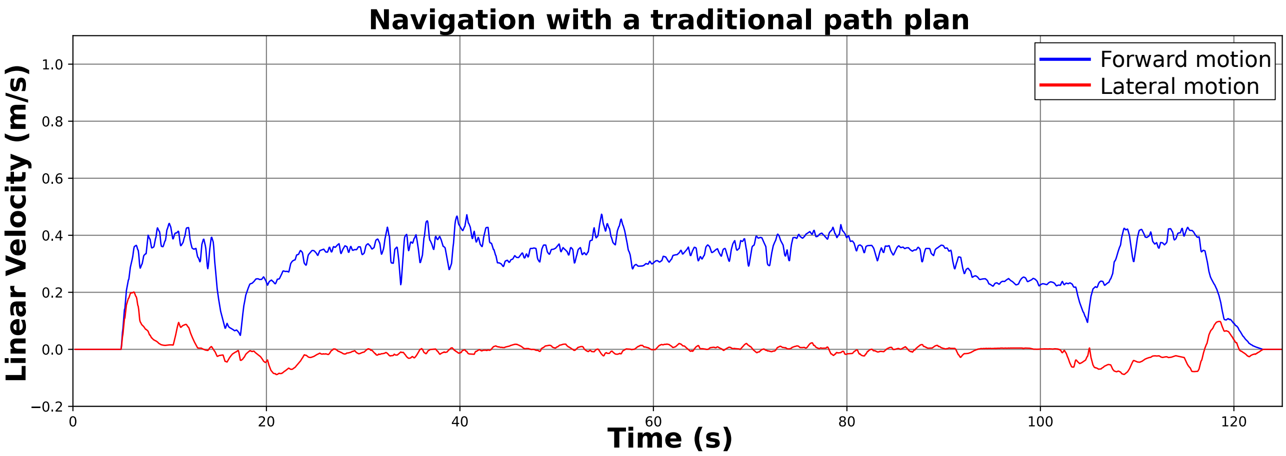
Table 1: Maximum allowed forward linear velocities when traversing different types of indoor spaces

Indoor space type	**Max. forward linear velocity (m/s)**
Junction	0.5
Corridor	1.0
Doorway	0.2
Standard space (e.g. room)	0.5

Fig. 10a demonstrates the benefit of using context-aware navigation as opposed to pure path following. Here, it can be observed that the robot produced non-smooth motion when following the path plan until it entered the semantic map, at which point the semantic motion behaviors are activated; this results in smooth, stable and predictable robot velocity based on the constraints in Tab. 1. Fig. 10b shows the velocity profile if a planner is unable to utilize the partial semantic information and simply produces a global path plan to be followed. In this plot, it can be seen that path following — executed by the ROS navigation stack — results in non-smooth motion and also takes longer (about $25s$) to reach the goal. In particular, it can be seen that the speed limit imposed to successfully navigate narrow doorways has a negative effect in other non-restrictive spaces, such as corridors, where higher speeds can be more efficient for navigation.



(a)



(b)

Figure 10: Linear velocity profile of robots executing two navigation plans generated for the same navigation task: (a) the hybrid navigation plan shown in Fig. 9d, and (b) a traditional path plan generated without using the partial semantic information. Execution of the hybrid plan requires about 20% less time than a traditional path plan.

7 Discussion and Conclusions

We presented an approach for generating composed maps using a prior collection of potentially heterogeneous robot maps. We particularly presented a formal representation of the composition problem in terms of operators for (i) preprocessing maps, (ii) converting maps from one type into another, and (iii) merging two maps of the same type; we also discussed how this formal model can be used to define a composition language that allows map composition tasks to be specified declaratively. Rather than relying on a single map of a specific map representation, map composition enables standard navigation components to leverage information from multiple fragmented, heterogeneous, and potentially incomplete maps. The proposed approach can simplify the mapping process in large environments, particularly in the context of multi-robot teams, as it enables the creation of multiple smaller, but accurate partial maps of the environment; these can then be merged online to generate an accurate map covering the complete environment. The use of such smaller maps also allows for an easier distribution and update of maps, without the need to completely remap a large environment due to changes in small regions thereof.

This work also considered the use of semantic maps for generating context-aware navigation behaviors, such that we also presented a context-based navigation task planner that is robust to incomplete semantic maps. The hybrid plans generated by the planner enable a robot to perform context-based navigation behaviors, while falling back to traditional path following in regions where semantic information is not available. This robustness makes it possible to build a partial semantic map, where only a few sub-regions of interest in a large environment can be semantically mapped to assist the robot during navigation, thereby reducing the deployment time in new environments.

There are various limitations of this work that should be addressed in follow-up studies. First of all we assumed that all prior maps have been registered in a common global reference frame, but future work should relax this assumption by auto-registering the available prior maps. Related to that, in this work, we discussed and implemented three map operators, but the repertoire of map operators can also be increased to improve the practical usefulness of the composition framework; for instance, an operator for map registration can be included so that auto-registration can be performed. With respect to the context-aware navigation, our work only exploits semantic knowledge about the environment for generating context-based navigation plans; however, conceptual knowledge about a robot itself can also be included in the planning process, particularly to identify and avoid indoor spaces that are challenging for a particular robot platform. Finally, the evaluation in this work was only performed to illustrate the feasibility of the proposed approach; in future work, we would like to evaluate the context-aware navigation with a real robot platform.

References

[1] Ilze Andersone. Heterogeneous map merging: State of the art. *Robotics*, 8(3), 2019.

[2] Dharmin Bipinbhai Bakaraniya. Context-based navigation using composed behaviours. Master's thesis, Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 St. Augustin, Germany, September 2021.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, volume 1, 2000.

[4] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robotics*, 37(6):1874–1890, 2021.

[5] Jonathan Crespo, Jose Carlos Castillo, Oscar Mozos, and Ramón Barber. Semantic information for robot navigation: A survey. *Applied Sciences*, 10(2), 2020.

[6] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 1989.

[7] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Aug 2005.

[8] Saeed Gholami Shahbandi and Martin Magnusson. 2d map alignment with region decomposition. *Autonomous Robots*, 43(5), Jun 2019.

[9] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robotics*, 23(1):34–46, 2007.

[10] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3), Apr 2013.

[11] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2), 2019.

[12] Ioannis Kostavelis, Konstantinos Charalampous, Antonios Gasteratos, and John K. Tsotsos. Robot navigation via spatial and temporal coherent semantic maps. *Engineering Applications of Artificial Intelligence*, 48, 2016.

[13] Nishith Maheshwari, Srishti Srivastava, and Krishnan Sundara Rajan. Development of an indoor space semantic model and its implementation as an indoorgml extension. *ISPRS International Journal of Geo-Information*, 8(8), 2019.

[14] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[15] Lakshadeep Naik, Sebastian Blumenthal, Nico Huebel, Herman Bruyninckx, and Erwin Prassler. Semantic mapping extension for openstreetmap applied to indoor robot navigation. In *Proc. Int. Conf. Robotics and Automation (ICRA)*, pages 3839–3845, 2019.

[16] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2012.

[17] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[18] Sebahattin Topal, Ismet Erkmen, and Aydan M Erkmen. A novel map merging methodology for multi-robot systems. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, 2010.

[19] M. Waibel et al. RoboEarth. *IEEE Robotics & Automation Magazine*, 18(2):69–82, 2011.

[20] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J.M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6), 2008.