Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)
Papers from the AAAI 2018 Fall Symposium

SOMA: A Framework for Understanding Change in Everyday Environments Using *Semantic Object Maps*

Lars Kunze[1], Hakan Karaoguz[2], Jay Young[3], Ferdian Jovan[4], John Folkesson[2], Patric Jensfelt[2], Nick Hawes[1]
[1]Oxford Robotics Institute, Dept. of Engineering Science, University of Oxford, United Kingdom
[2]Centre for Autonomous Systems, KTH Royal Institute of Technology, Sweden
[3]Sheffield Robotics, United Kingdom
[4]Howey Research Group, Dept. of Engineering Science, University of Oxford, United Kingdom

Abstract

Understanding change related to the dynamics of people and objects in everyday environments is a challenging problem. At the same time, it is a key requirement in many applications of autonomous mobile service robots.

In this paper we present a novel semantic mapping framework which maps locations of objects, regions of interest, and movements of people over time. Our aim with this framework is twofold: (1) we want to allow robots to reason semantically, spatially, and temporally about their environment, and (2) we want to enable researchers to investigate research questions in the context of long-term scenarios in dynamic environments.

Experimental results demonstrate the effectiveness of the framework which was deployed on mobile robot systems in real-world environments over several months.

1 Introduction

Everyday environments such as our homes, hospitals, and offices are dynamic and change over time. They change because people perform a broad range of everyday activities in them. During those activities, the locations of people and objects change as they move (or are moved) from one place to another. However, understanding change related to the dynamics of people and objects in everyday environments is a challenging problem. Yet it is a key requirement for autonomous robots to accomplish their tasks successfully.

For example, whilst *setting a table for dinner* a person gets plates, cups, knives, and forks from cupboards and drawers in the kitchen and lays them out on a dining table. However, robots can only make partial observations of those events due to their limited sensing capabilities and their egocentric viewpoints. Whilst observing such a scene, a mobile robot might only perceive a person moving through space and objects appearing and disappearing at various places at different points in time. As the robot only perceives snapshots of the underlying events it needs to reason about *what* it has seen, *where* and *when* to infer what to do next.

What have I seen? Where? and *When?* are essential questions in many robotic tasks such as searching for objects and the surveillance of human activities (Galindo and others 2008; Kostavelis and Gasteratos 2015). *Semantic environment maps* can provide answers to these questions as they link semantic information about the world (e.g. objects and people) to spatio-temporal representations. To this end, they are important resources in many robotic tasks. They allow autonomous robots to interpret (or ground) high-level task instructions; to plan and reason about how to achieve a task in a given environment; and to communicate observations and results to humans. However, constructing, maintaining, and using such maps in everyday, dynamic environments poses several challenges: (i) observations from sensor data need to be interpreted at a semantic level; (ii) interpretations need to be integrated into a consistent map; (iii) the map needs to be continuously updated to reflect the dynamic changes in an environment (over extended periods of time); and (iv) queries to the semantic map need to provide task-related information by taking semantic and/or spatio-temporal constraints into account.

In the past, many semantic mapping approaches in computer vision and robotics have addressed challenges (i) and (ii) by interpreting and integrating data from various sensors including laser rangefinders (Nuchter and Hertzberg 2008; Rusu and others 2009; Blodow and others 2011), stereo and monocular cameras (Sengupta and others 2013; Sunderhauf and others 2016), and RGB-D cameras (Pangercic and others 2012; Hermans, Floros, and Leibe 2014; Gunther and others 2015). These approaches assume that the environment is static and hence focus on the mapping of large-scale structures such as rooms, walls and furniture.

In this work, we address challenges (ii), (iii), and (iv). With respect to challenge (i), we use and adapt state-of-the-art robot perception methods that provide intermediate semantic interpretations from sensor data. Our work focuses on dynamic aspects of everyday environments including the varying locations of objects, regions of interest that potentially change over time, and movements of people. To this end, we investigate how objects, regions, and movements of people can be indexed by space and time so that map updates and queries can be handled both effectively and efficiently.

To address these challenges, we have designed, developed, and evaluated SOMA; a framework for constructing, maintaining, and querying *Semantic Object Maps*. In our work, *Semantic Object Maps* model semantic and spatial information about objects, regions, and trajectories of agents over time. Therefore they can provide answers to the questions: *What, Where* and *When?*

The presented framework allows autonomous robots to construct and update (or revise) maps automatically from

Table 1: SOMA concepts, their definition and examples.

Concept (Representation)	Definition	Examples
Object (3D pose & bounding box)	*Object* denotes a tangible thing that occupies a volume in space	Table, chair, monitor, cup, book
Region (2D polygon)	*Region* denotes an area of interest in space	Workplace, entrance, waiting area
Trajectory (2D pose array)	*Trajectory* denotes a path an agent has followed through space as a function of time	Human/ robot trajectory

observations using state-of-the-art perception methods and to query them from within the robots' control programs. At the same time, maps can be edited and queried by knowledge engineers and researchers for providing domain knowledge and for investigating research questions in long-term scenarios. It is important to note that we have designed SOMA with these two different user groups in mind: autonomous robots and researchers. While robots can add their observations and query maps for decision making, researchers can model aspects of the environment and/or analyse and extract spatio-temporal data collected by autonomous systems. The latter enables researchers to build and learn novel models about the (long-term) dynamics in everyday environments.

In this work, we have focused on the mapping for objects, regions, and humans in long-term and dynamic settings. Table 1 provides an overview of the high-level concepts used within SOMA. At an abstract level, our approach is similar to other works as it uses similar concepts for representing entities in the environment. Concepts such as objects, regions, and trajectories are natural and common sense. However, our approach differs significantly in the way we store, index, link, and query observations, interpretations, and semantic concepts over time. The main contributions of this work are as follows:

- an open-source semantic mapping framework (SOMA), designed for long-term, dynamic scenarios;
- a multi-layered knowledge representation architecture linking observations, interpretations, and semantic concepts using spatio-temporal indices;
- an adaptable mechanism for grounding objects in sensor data and a set of (extendable) interfaces for updating *Semantic Object Maps* over time;
- a query interface for retrieving and manipulating objects in *Semantic Object Maps* using semantic and spatio-temporal constraints; and
- a long-term case study of SOMA and *Semantic Object Maps* in a real world environment.

2 Related Work

Most research in the area of *robotic mapping* is based on metric representations (Thrun 2003). However, in the last decade, many approaches to *semantic mapping* have been proposed; a detailed account on the topic is given by (Pronobis and others 2010; Pronobis 2011) and an overview is provided by (Kostavelis and Gasteratos 2015).

(Capobianco and others 2016) propose a standardised way of representing and evaluating semantic maps. They define semantic mapping as an incremental process that maps relevant information of the world (i.e., spatial information, temporal events, agents and actions) to a formal description supported by a reasoning engine. Our work adopts a similar approach, we incrementally map spatio-temporal information about objects, people and regions and query those information using both standardised database queries and specialised inference mechanisms.

Several semantic mapping approaches have mainly focused on both the interpretation and the integration of data from various sensors including laser rangefinders, e.g. (Nuchter and Hertzberg 2008; Rusu and others 2009; Blodow and others 2011), stereo and monocular cameras, e.g. (Sengupta and others 2013; Sunderhauf and others 2016) and RGB-D cameras, e.g. (Pangercic and others 2012; Hermans, Floros, and Leibe 2014; Gunther and others 2015). Most of these approaches assume that the environment is static and hence focused on the mapping of static, large-scale structures such as rooms, walls and furniture. Our work is different from these approaches in two aspects. First, we do not develop methods for interpreting sensor data, but rather build on and adapt state-of-the-art robot perception methods (Aldoma and others 2012; Wohlkinger and others 2012), and secondly, we focus on the mapping, updating, and querying of semantic maps in *dynamic environments*.

A few semantic mapping methods approached the topic from a different angle. They focused on the design of ontologies and linking those to low-level environment representations (Zender and others 2008; Tenorth and others 2010). For example, work by (Pronobis and Jensfelt 2012) shows how different sensor modalities can be integrated with ontological reasoning capabilities to infer semantic room categories. Representing environment maps using Semantic Web technologies also enables robots to exchange information with other platforms via the cloud (Riazuelo and others 2015). We consider these types of approaches complementary to our work, as the semantic categories in our framework can be integrated and linked with exiting ontologies. For example, in (Young and others 2017a) hypotheses about objects are linked to structured, semantic knowledge bases such as DBpedia[1] and WordNet (Fellbaum 1998).

(Elfring and others 2013) presents a framework for probabilistically grounding objects in sensor data. In general, our framework does not make any strong assumptions about how objects are grounded in robot observations. Instead, the grounding of objects needs to be specified or learned by a user in the Interpretation layer of the framework (cf. Section 3.3).

[1] http://wiki.dbpedia.org

(Bastianelli and others 2013) present an on-line, interactive approach and an evaluation (Gemignani and others 2016) for the construction of semantic maps. Similarly, our work supports labelling of discovered objects by the crowd (cf. Section 3.5). However, our approach is off-line and is designed to work in an asynchronous way.

Our work is similar to (Mason and Marthi 2012) and (Herrero, Castaño, and Mozos 2015) which both focus on semantic querying of maps in dynamic environments. (Herrero, Castaño, and Mozos 2015) propose an approach based on relational databases which hold semantic information about objects, rooms and their relations required for mobile robot navigation. Our approach is similar as it also considers objects and regions in space (but not only rooms). However, in our approach relations between objects and regions do not have to be modelled explicitly, but can be inferred using spatial reasoning. (Mason and Marthi 2012) focus on semantic querying and change detection for objects. In their work, *objects* mean geometrically distinct occupied regions on a plane whose locations are described in a global reference frame. In contrast, our work can distinguish between unknown objects, classified objects, and known object instances. Our spatial indexing allows us to relate objects in a local, a global, and the robot reference frame. Furthermore, we can relate objects to regions and human trajectories.

Most similar to our approach is the semantic mapping framework by (Deeken, Wiemann, and Hertzberg 2018). Their framework is designed to maintain and analyse the spatial data of a multi-modal environment model. It uses a spatial database to store metric data and link it to semantic descriptions via semantic annotation. Spatial and semantic data can be queried from the framework to augment metric maps with topological and semantic information. This design and functionality is very similar to our approach. However, our approach goes beyond spatial and semantic information as it also includes temporal information about objects, regions, and people. Thereby it allows robots and users to reason not only about static configurations but also about temporally extended events such as everyday activities.

3 The SOMA Framework

3.1 Overview

Figure 1 provides a conceptual overview of the designed framework. The framework consists of two parts: (1) the SOMA core and (2) a set of SOMA extensions (or tools). Overall, the core has four layers. The three horizontal layers are interconnected and manage the information at different levels of abstractions: from observations (i.e. raw sensor data) and their interpretations to semantic concepts. These three layers are responsible for the representation within SOMA. The vertical interface layer provides access to all three levels. A set of extensions (or tools) use this layer for visualising, editing, querying and extending *Semantic Object Maps*. This allows knowledge engineers to extend and analyse them. Similarly, robots and user applications can access and manage maps through the interface layer.

Let us now consider the process of storing new information in SOMA. First, the robot's observations, in form of raw

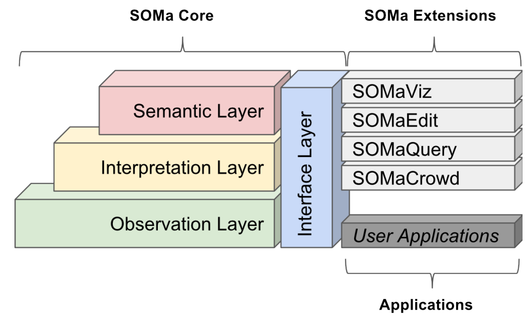

Figure 1: The SOMA framework consists of the SOMA Core and a set of extensions/tools for visualising, editing, and querying *Semantic Object Maps*.

sensor data, are stored and spatially and temporally indexed in an observation layer. Second, an interpretation layer analyses these observations using perception methods, such as segmentation, object recognition, object classification, and people tracking, consolidates the results, and generates consistent descriptions at a semantic level. Finally, observations, interpretations, and semantic descriptions are linked together which allows the robot to query them at various levels using spatial, temporal, and/or semantic constraints.

3.2 Observation Layer

The role of the observation layer is to store both raw, unprocessed sensor data from the robot, along with any meta-data that might be useful in interpreting and processing that data by its systems. In order to accomplish this the observation layer stores the input from the robot's sensors during learning tasks. All other layers of SOMA also access this stored data. We store *views* which contain data from a single robot perception action, and we collect series of views into *episodes*. For our object learning tasks, a single *view* stores the point cloud, RGB image, depth image, current pose of the robot as well as any odometric transforms. An *episode* collects the series of views chosen by a planning algorithm for a particular learning task. Episodes and views may also have attached meta-data tags, which allows multiple different perception pipelines—perhaps all using different criteria to trigger, control and interpret data from learning tasks—to make use of the same database.

One of our design goals was to produce a way of storing raw robot perception data that would allow us to fully re-generate a SOMA database, performing all requisite processing steps along the way. Given just a copy of a robot's observation layer, this is possible, as the enclosed raw observations can be *re-played* as though they are being made in real-time. This is a key capability for evaluating different perception algorithms and pipelines on or off a robot. This functionality also helps in terms of fault tolerance—for instance, if a robot has been running for period of time with an undetected fault in a segmentation or object recognition layer, we are able to correct the error and fully re-generate the database from the observation layer, processing the data with the new corrected system, resulting in no loss of data.

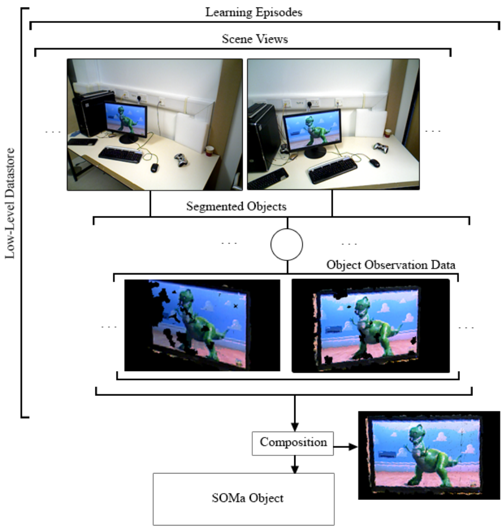

Figure 2: Data structures used in the Interpretation layer to compose high-level objects in the Semantic layer. First, individual views of a scene are stored. Second, object segments are found and linked with multiple observations of the same object across views. Finally, a SOMA object is created.

3.3 Interpretation Layer

The interpretation layer takes input from the observation layer and mainly contains application-specific methods for processing data. While the observation layer can be thought of as a wrapper around a robot's sensors, the interpretation layer is regarded as the part of the system that engages in application-specific processing of that data. In object learning, the first step in interpretation is to apply a segmentation algorithm, such as a depth-based segmentation or a region proposal network, in order to extract object proposals for further processing. SOMA provides a way of structuring the output of such segmentation algorithms by providing a scene graph-like object structure. This provides tools for storing data about individual segmented objects and their relationship to a view, and an episode, and allows a developer to collate observations of objects as further views are taken. The exact choice of algorithms used for scene segmentation, object tracking between views, and otherwise, are all left up to the developer as part of the design of their own application-specific interpretation layer.

Once the interpretation pipeline has processed and filtered the raw sensor output provided by the observation layer, high-level SOMA objects can be composed from the processed data. An example of this is shown in Figure 2. High-level objects represent the results of processing, and may record the output of object recognition algorithms over a series of views of an object, merged 3D models constructed from multiple views, and meta-data. These high-level objects link back to the low-level observations from which they are composed—the developer can go back and forth as desired from complete, merged objects to their constituent parts. The objects can then be used in further applications built on-top of SOMA—they can be shown to end-users in a labelling application, emailed, tweeted, visualised on a website, used in an application to locate lost mugs, processed further, or anything else the developer may desire.

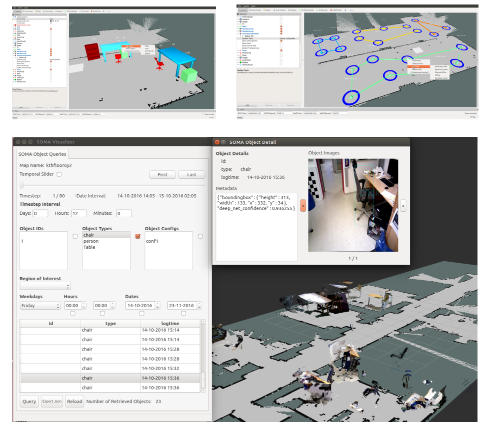

Figure 3: Examples of SOMA extensions. Top: SOMAEdit can be used for adding, editing or manipulating objects (left) and regions of interest (right). Bottom: SOMAViz is used for building spatio-temporal queries and visualising results.

3.4 Semantic Layer

The semantic layer stores high-level knowledge extracted from the robot's observations (cf. Table 1). The high-level knowledge could be the recognised object instances received from various recognition/detection pipelines or tracked objects from segmentation/tracking pipelines. Each high-level data instance is stored with spatio-temporal information such that the evolution of the knowledge about each object instance is maintained and can be retrieved. Moreover, each high-level SOMA object is linked with other SOMA layers by means of SOMA IDs in order to access all the knowledge about the object within the framework.

Furthermore, the semantic layer can store additional information about the object such as 3D models, camera images and any type of meta-data to build up a complete knowledge base. The stored high-level information could help the user to understand the semantics of individual environments, allow the robot to do high-level reasoning for accomplishing tasks such as object finding and/or grasping.

3.5 Interface Layer & SOMA Extensions

The interface layer acts as a backbone between the different SOMA layers and the user for exchanging data. As such, the robot/user can insert, delete, update and query data using SOMA extensions and other applications (Figure 3).

SOMAEdit allows users to create virtual scenes without any perceptual data. With this editor, users can add, remove or move objects and regions on top of a metric map.

SOMAQuery allows users to query maps using semantic, spatial, and/or temporal constraints. A query might ask for all objects of a certain type ("Select all cups"). Such a query can be further constraint by spatio-temporal constraints ("Select all cups in meeting rooms on Mondays be-

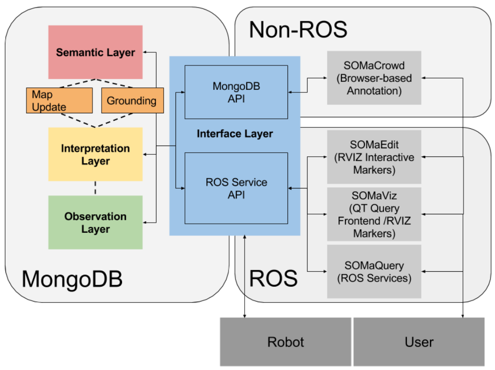

Figure 4: Implementation details of SOMA.

tween 10:00–12:00"). Spatial constraints can be used to determine if spatial entities are *near* to another entity, *within* an entity (region), or if they *intersect* with another entity. Temporal constraints can be formulated by using time points or temporal intervals. To discover temporal patterns and periodic processes *hour of the day*, *day of the week*, and *day of the month* are of particular interest.

Other extensions allow users to visualise query results (SOMAViz) and crowd-source missing object labels (SOMACrowd).

4 Implementation

We have implemented SOMA[2] based on ROS and MongoDB. The overall implementation structure of the framework is shown in Figure 4. ROS is used as the backbone of the entire SOMA framework as it is the most common platform used in the robotics research community. The individual SOMA layers and components are all developed as ROS nodes so that each of them can communicate with any other ROS component. Data structures used to store SOMA objects are themselves ROS messages composed of primitive ROS types. This provides a common interface between systems so long as they are built on the ROS stack.

For data storage and handling at all different layers, MongoDB is used. MongoDB provides great flexibility both in terms of the data structure and query capabilities. As such, all the SOMA data that is populated at different layers can be stored as ROS messages in MongoDB. Moreover, users can perform complex spatial queries using MongoDB's Geospatial API and/or projection criteria.

For visualisation tools, we have employed the Qt framework. Qt offers an end to end solution for developing graphical user interface as front-end and combining the back-end with ROS framework using multiple threads.

5 Experimental Validation

Our work was motivated by the European project, STRANDS (Hawes and others 2016). In STRANDS, we investigated spatio-temporal representations and activities in

[2]`https://github.com/strands-project/soma`

Figure 5: Objects discovered during deployment. SOMA stores the RGB-D images, point clouds, and other meta data.

long-term scenarios. Within the project, we were interested in providing services to humans in everyday environments. Tasks that our robots performed included object search, object discovery as well as activity recognition and movement analysis. In this context, we conducted a series of robot deployments in real-world office environments over several months in which we have evaluated this work.

Traditional, database-centric measures such as performance metrics over read/write access times and load-bearing tests would necessarily be evaluating the technology on which SOMA is *built*, rather than SOMA itself as a technology. Instead, we prefer to evaluate SOMA by looking at the range of applications it has been used for.

SOMA was used in multiple long-term deployments at two different sites—the Transport Systems Catapult (TSC) and a facility belonging to Group 4 Security (G4S), both in the United Kingdom. We report details of these deployments with respect to the three main entities that are represented within SOMA: *Objects*, *Regions*, and *Trajectories*.

5.1 Objects

While SOMA has been used as the underlying technology behind the development of more advanced and robust robot perception systems as described above, it has also served a dual role as the key component in many user-facing robot applications. Figure 5 shows a small sample of objects learned by the robot at the Transport Systems Catapult (TSC) deployment site, using the perception pipeline of (Young and others 2017b). In particular, 2D images of objects were used to pass to a Convolutional Neural Network (CNN) for identification, as in (Young and others 2017b; 2017a), as well as being passed to end-users at the site for live labelling. SOMA is also a data collection platform—we have used it to store and distribute scenes for future, offline labelling by human annotators. SOMA's performance in this area is dependent on the perception pipelines that feed information to it. Overall, the system stored 141 scenes and 341 scenes respectively, during the first and the second deployment at the TSC. The design of SOMA means that these

Table 2: Object learning performance (TSC, Y3/Y4).

Performance Measure	Y3	Y4
# Learning Episodes	56	**80**
# Views Taken	141	**341**
# Waypoints Visited	**25**	10
# Avg. Views / Learning Episode	∼2.5	∼**4**
# Avg. Episodes / Waypoint	∼2.24	**8**
# Autonomously Segmented Objects	445	**668**

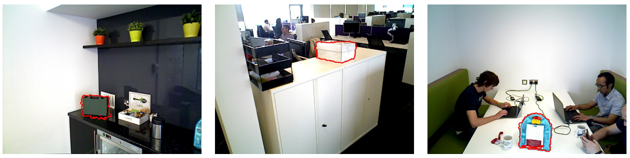

Figure 6: SOMA was used to generate reports of predefined surface areas in which objects were highlighted (TSC, Y3).

scenes can be re-processed later offline, using different perception pipelines, algorithms or filters if so desired, to extract different objects or different kinds of information from them. A comparison between the object learning pipelines used in year three (Y3) and year four (Y4) at the TSC site are shown in Table 2.

In the first deployment at TSC (Y3), SOMA was used to provide reports of objects discovered on predefined surfaces at the site (Figure 6). As the robot engaged in its normal object learning tasks, reports were generated and presented in a web-based blog interface for end-users to access.

In other experimental work, we designated two surfaces at the TSC site to be "learning tables", where office workers could bring objects for the robot to learn about. The robot would then visit the tables twice a day, and attempt to learn and identify any objects it had found. Using a CNN trained on a large image database, with 1000 possible categories, it would tweet about them while attempting to identify them. Internally, the system is made possible by SOMA's function of announcing when new objects are entered into the system, which then triggers the object identification and tweeting process by passing to those functions the 2D images of objects entered into SOMA.

During the last long-term deployment in TSC site (Y4), we have employed a CNN based object detection pipeline. This pipeline was able to detect 20 object categories including person, chair, monitor, etc. and it was possible to extract a partial 3D view of the object using the registered depth information. As such the object location with respect to the robot and the global metric map can be identified. Figure 7 shows an example of a detected chair and the extracted partial 3D view. The detected objects were then stored as high-level SOMA objects with spatio-temporal information. Table 3 shows some detailed statistics about the objects detected with this pipeline during the deployment. The results show that the most detected objects were chairs, people and monitors which can be expected given that the robot was working on an office environment (Table 3).

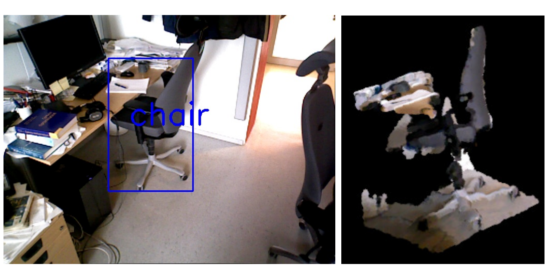

Figure 7: Processing steps of CNN-based object detection. Left: Detect object candidate. Right: Extract partial view.

Table 3: Detection results (TSC, Y4, ≈120 days).

Object type	Number of detections
People	178
Chairs	171
Monitors	104
Other objects	574

Table 4: Perceived objects (#) per time of day (TSC, Y4).

Hours	07:00-12:00	12:00-17:00	17:00-00:00
Monday	56	98	0
Tuesday	21	85	0
Wednesday	18	175	3
Thursday	184	224	0
Friday	0	67	0

We have also analysed temporal aspects of the Y4 deployment in terms of high-level SOMA object perception using the SOMAQuery interface. Table 4 shows the daily object perception statistics w.r.t the time of the day for the entire deployment. From the table it is observed that the robot was mostly active during Wednesdays and Thursdays while it has never been used in the weekends for object perception. It is also observed that the robot was most active during the afternoon hours but it was only rarely used during out of office hours (after 17:00). In total, the robot has perceived 930 high level SOMA objects during the entire Y4 deployment.

5.2 Regions

SOMA was used as the main memory component in (Karaoguz and others 2017) for human-centric partitioning of the environment. In the work, it was assumed that the co-occurrence of objects and humans can be used to identify densely populated areas. For this task, a CNN-based object detector and RANSAC-based tabletop detector were employed to detect objects. During the robot's operation, the detected objects were all stored as high level objects within in SOMA. After a set of observations were made, a reasoning module was employed to query SOMA objects and locate object clusters. These object clusters were then used to identify the dense regions. Figure 8 (left) shows the resulting regions. The proposed system discovered 16 regions of

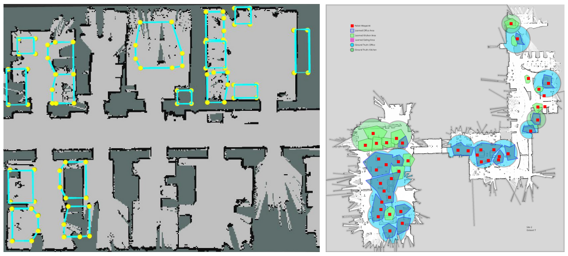

Figure 8: Examples of learned regions. Left: Auto-generated human-centric partitions of offices. Right: Learning of semantic labels associated with regions at TSC (Y4).

which 14 actually corresponded to manually annotated ones. The people density analysis showed that nearly 90% of the dense areas lay within or around the generated regions.

SOMA was also used in (Young and others 2017a) as part of a system for learning semantic labels associated with regions of space. Here, SOMA was used to represent objects discovered in the environment (Ambruş and others 2014). These objects were then passed to a CNN to be labelled. The system then used text mining of large text corpora (in this case Wikipedia) to find those room categories most strongly related to the labels of the discovered objects. Results from the TSC deployment are show in Figure 8 (right).

Discovery of these semantic room labels allowed us to draw bounding polygons around the areas of space where the related objects were observed. The result, as shown in Figure 8 where blue regions indicate office areas and green regions indicate kitchen areas, largely covered the same areas as annotated by human annotators. These learned, labelled regions can then be fed back in to SOMA and its own internal region representation, and potentially used by a robot for various tasks such as object search or activity recognition.

In general, SOMA has been a powerful API for improving both the internal object perception pipelines used on our robots—for instance, the region representation is key to our approach to view planning—but also a tool for building user-facing applications that provide a robot's-eye view of the world. SOMA's ability to support input from multiple, arbitrary perception pipelines has also been a great tool in development and debugging of these systems along with the suite of visualisation tools available.

5.3 Trajectories

In STRANDS, we have used a multi-sensor-based approach for people detection and tracking (Dondrup and others 2015). During the deployments, we gathered thousands of human trajectories at both sites. (Jovan and others 2016) analysed and predicted the level of activities at G4S. The level of activity was measured by the number of trajectories, within a particular time and location. This kind of analysis on human trajectories in temporal and spatial scale are possible via SOMAQuery. Temporal queries which involve periodic intervals during the week such as "Select all trajectories between 08:00–17:00 on every Monday" or specific time intervals such as "Select all trajectories between 08:00–12:00 on Sunday 12th March 2016" can be interpreted by SOMAQuery and visualised by SOMAViz. With help of SOMAQuery and background knowledge about the normative behaviour of people at G4S (e.g. "no employee is allowed to work during the weekend"), (Jovan and others 2016) employed a trajectory filtering to filter false detections captured during the deployment. A trajectory statistics for a set of temporal queries is shown in Table 5.

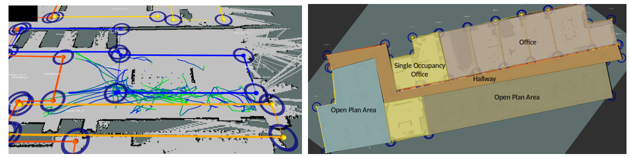

Figure 9: Left: Trajectories in kitchen area (TSC, Y4). Right: Region classification based on trajectories (G4S, Y2).

We used SOMAEdit to segment the sites' map into regions based on their functions. Given these regions, querying trajectories within these particular areas becomes possible. Thereby, SOMAQuery is able to interpret spatial queries such as "Select all trajectories within kitchen area" at different times of day (Figure 9). A brief comparison between G4S and TSC on spatial queries is shown in Table 5. (Jovan and others 2016) discovered that the temporal predictive model of human activities for each region provides a spatio-temporal signature which can further be used to classify regions based on their functionality. The region classification based on trajectories can be seen in Figure 9 (right).

6 Conclusions

In this paper, we have presented a semantic mapping framework for mobile robots, called SOMA. SOMA uses a three layered architecture to model objects, regions, and trajectories of agents. We explained how these layers interact with each other, how they can be accessed via an interface layer from both robots and user applications. We presented an experimental validation of SOMA by showcasing several use cases for the framework in real-world, long-term scenarios.

SOMA stores discrete observations of objects and regions. However, many new semantic mapping applications are based on continuous observations from sensors (e.g. cameras), and recent work on visual SLAM makes collection of this kind of data easy (Stückler and others 2015). We will investigate, in future work, how this kind of data, as well as segmented contents, can be integrated into our maps. Conceptually, however, our representation tools are general enough to support these, and other, new approaches. Hence, we believe that the open-source framework SOMA can have a wide within the robotics community.

Acknowledgments The work was supported by RAIN, EPSRC [EP/R026084/1]; Mobile Robotics Programme Grant, EPSRC [EP/M019918/1]; ALOOF, EPSRC [EP/M015777/1]; STRANDS, EU FP7 [600623].

References

Aldoma, A., et al. 2012. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation.

Table 5: Trajectory statistics for temporal and spatial queries

Constraint Type	Query	G4S		TSC (Y4)	
		# Trajectories/hour	$\overline{\text{Length}}$	# Trajectories/hour	$\overline{\text{Length}}$
Temporal	Workdays 08:00-12:00	16.08	2.47	9.07	3.61
	Workdays 12:00-17:00	11.42	2.51	11.34	3.38
	Workdays 17:00-00:00	0.19	2.31	0.36	3.34
	Weekend	0.08	0.77	0.00	0.00
Spatial	Open Plan Area	3.02	2.45	0.18	3.90
	Kitchen	0.55	2.82	1.43	3.60
	Meeting Room	0.27	1.66	0.07	3.88
	Hallway	3.51	2.72	1.60	3.86

IEEE RAM 19(3).

Ambruş, R., et al. 2014. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *IROS*.

Bastianelli, E., et al. 2013. On-line semantic mapping. In *ICAR*.

Blodow, N., et al. 2011. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In *IROS*.

Capobianco, R., et al. 2016. A proposal for semantic map representation and evaluation. *CoRR* abs/1606.03719.

Deeken, H.; Wiemann, T.; and Hertzberg, J. 2018. Grounding semantic maps in spatial databases. *RAS* 105:146 – 165.

Dondrup, C., et al. 2015. Real-time multisensor people tracking for human-robot spatial interaction. In *ICRA*.

Elfring, J., et al. 2013. Semantic world modeling using probabilistic multiple hypothesis anchoring. *RAS* 61(2).

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication.

Galindo, C., et al. 2008. Robot task planning using semantic maps. *RAS* 56(11).

Gemignani, G., et al. 2016. Interactive semantic mapping: Experimental evaluation. In *Experimental Robotics*.

Gunther, M., et al. 2015. Model-based furniture recognition for building semantic object maps. *Artificial Intelligence*.

Hawes, N., et al. 2016. The STRANDS project: Long-term autonomy in everyday environments. *IEEE RAM*.

Hermans, A.; Floros, G.; and Leibe, B. 2014. Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images. In *ICRA*.

Herrero, J. C.; Castaño, R. B.; and Mozos, O. M. 2015. An inferring semantic system based on relational models for mobile robotics. In *IEEE ICARSC*.

Jovan, F., et al. 2016. A Poisson-Spectral Model for Modelling the Spatio-Temporal Patterns in Human Data Observed by a Robot. In *IROS*.

Karaoguz, H., et al. 2017. Human-centric partitioning of the environment. In *ROMAN*.

Kostavelis, I., and Gasteratos, A. 2015. Semantic mapping for mobile robotics tasks: A survey. *RAS* 66:86 – 103.

Mason, J., and Marthi, B. 2012. An object-based semantic world model for long-term change detection and semantic querying. In *IROS*.

Nuchter, A., and Hertzberg, J. 2008. Towards semantic maps for mobile robots. *RAS* 56(11):915 – 926.

Pangercic, D., et al. 2012. Semantic object maps for robotic housework - representation, acquisition and use. In *IROS*.

Pronobis, A., and Jensfelt, P. 2012. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *ICRA*.

Pronobis, A., et al. 2010. Semantic modelling of space. In *Cognitive Systems*.

Pronobis, A. 2011. *Semantic Mapping with Mobile Robots*. Ph.D. Dissertation, KTH Royal Inst. of Tech., Sweden.

Riazuelo, L., et al. 2015. Roboearth semantic mapping: A cloud enabled knowledge-based approach. *IEEE T-ASE* 12(2).

Rusu, R. B., et al. 2009. Model-based and learned semantic object labeling in 3d point cloud maps of kitchen environments. In *IROS*.

Sengupta, S., et al. 2013. Urban 3d semantic modelling using stereo vision. In *ICRA*.

Stückler, J., et al. 2015. Dense real-time mapping of object-class semantics from rgb-d video. *JRTIP* 10(4).

Sunderhauf, N., et al. 2016. Place categorization and semantic mapping on a mobile robot. In *ICRA*.

Tenorth, M., et al. 2010. Knowrob-map - knowledge-linked semantic object maps. In *IEEE-RAS Humanoids*.

Thrun, S. 2003. Exploring artificial intelligence in the new millennium. In *Robotic Mapping: A Survey*.

Wohlkinger, W., et al. 2012. 3dnet: Large-scale object class recognition from cad models. In *ICRA*.

Young, J., et al. 2017a. Making sense of indoor spaces using semantic web mining and situated robot perception. In *The Semantic Web: ESWC 2017 Satellite Events*.

Young, J., et al. 2017b. Semantic web-mining and deep vision for lifelong object discovery. In *ICRA*.

Zender, H., et al. 2008. Conceptual spatial representations for indoor mobile robots. *RAS* 56(6).