SymboSLAM: Semantic Map Generation in a Multi-Agent System

Brandon C. Colelough,
School of Engineering and Information Technology, University of New South Wales, Australia

***Abstract*—Sub-symbolic artificial intelligence methods dominate the fields of environment-type classification and Simultaneous Localisation and Mapping. However, a significant area overlooked within these fields is solution transparency for the human-machine interaction space, as the sub-symbolic methods employed for map generation do not account for the explainability of the solutions generated. This paper proposes a novel approach to environment-type classification through Symbolic Simultaneous Localisation and Mapping, *SymboSLAM*, to bridge the explainability gap. Our method for environment-type classification observes ontological reasoning used to synthesise the context of an environment through the features found within. We achieve explainability within the model by presenting operators with environment-type classifications overlayed by a semantically labelled occupancy map of landmarks and features. We evaluate SymboSLAM with ground-truth maps of the Canberra region, demonstrating method effectiveness. We assessed the system through both simulations and real-world trials.**

***Index Terms*—Environment-Type Classification, Semantic Map, Multi-Agent System, SLAM, Ontology, Symbolic Reasoning**

I. Introduction

An environment is a "broadly defined term" that refers to the "natural or anthropogenic systems which can surround a living or non-living entity" [1]. The symbolic constructs attributed to environments are "created by human acts of conferring meaning to nature and the environment" [2]. Thus, environments may be represented by symbolic concepts and described through their physical attributes [1], [2]. This study focuses on assigning symbolic concepts to areas of an environment by examining the physical objects found within. Making this link between an object and an environment type is a problem that requires some level of human understanding present within the methodologies for classification, as the environment labels are complex societal constructs. Encoding this knowledge within an architecture requires a system that can understand the nuances and intricacies of the human world and the relationships people draw between environmental concepts and the objects associated with them. These manifest as challenges in modelling an environment for the purpose of classification, raising questions such as:

1) How can a symbolic representation of an environment be used to classify the environment type?
2) How can an environment be transformed into a symbolic representation?

Symbolic Artificial Intelligence (AI) is a field of study that offers a solution to question one. It incorporates domain-level expertise within an architecture to enable higher-order reasoning about concepts. An ontology is an instrument within the field of symbolic reasoning which includes the properties and relations of objects and ideas within a specific subject area [3]. It posits a viable solution to link the concepts of objects and environment types. It is evident from the literature presently available within symbolic reasoning and remote sensing fields that the information required for enabling environment-type classification is not readily available or easily obtainable. As such, a method to extract information is necessary to allow higher-order classification of the environment. Simultaneous Localisation and Mapping (SLAM) offers a solution having excellent applicability over a range of domains for which many systems have already demonstrated promising results [4]–[6]. Joo et al. demonstrated that it is possible to generate semantic understanding from defined spatial relations [7]. The incorporation of semantic knowledge to features in the environment offered a new method to describe SLAM situations transparently. The overarching motivation to research environment-type classification through symbolic reasoning was to transparently describe complex environments with a simple graphical representation. Transparency within a system can be achieved through the tenets of "interpretability, explainability and predictability" and is the "overarching concept" required for a human-machined teaming environment trust architecture [8]. Using an ontology for symbolic reasoning offers the opportunity to increase the explainability and interpretability of the classification result, enabling transparency to be maintained within the human-machine teamed environment. Thus, transparency within a system may be achieved by conducting high-level environment classification through the contextualisation of objects found within and displaying these classifications as a 2D map. Furthermore, an architecture that utilises an ontology as the symbolic reasoning component will be capable of encoding human constructs into the architecture's knowledge base to ensure the architecture converges upon a solution. The identified gap in the literature leads our research to answer the following overarching question; *can a symbolic reasoning approach for multi-agent SLAM increase the meaning for 2D maps?* The objectives of this research are:

RO1 Determine whether intelligent edge agents on a Multi-Agent System (MAS) can extract environmental features and share information with a centralised control agent.
RO2 Determine if a centralised control structure with intelligent MAS agents can conduct SLAM.
RO3 Determine if a symbolic representation of the environ-

ment is suitable to build classified environment-type 2D maps.

The main contribution of this study is a transparent method for environment-type classification, achieved through explainability within the map generation process. This study presents system explainability for environment-type reasoning through semantically labelled landmarks as an occupancy map. Furthermore, this study provides ontological reasoning of the semantically labelled landmarks as a method to conduct environment-type classification.

The remainder of this paper is organised as follows. First, in Section II, we review methods for feature extraction and spatial recognition on a MAS and symbolic reasoning through ontological design. The problem space is then constructed to describe both the project sub-tasks and how they relate to the overall aim of the proposed architecture. Next, Section III introduces the proposed solution to the main research question and research sub-objectives, with development steps in the appendix. Following this within Section IV, the metrics for success across all project contributions are introduced, and the method to evaluate the proposed SymboSLAM architecture is discussed. Finally, in Section V the simulated and measured experiments conducted with the SymboSLAM architecture are presented alongside the results gathered with supplementary information provided in the appendix and externally. We then conclude this paper with a discussion on the success of the proposed architecture in Section VI, followed by the proposal of further research questions for further investigation in Section VII.

II. Background

A. Environment Type Classification

Environment type classification is a field of study presently dominated by the implementation of remote sensing applications utilised as the primary source of information. These remote sensing techniques deal with "high-resolution geospatial data" to "find the land cover classes" and are broadly sorted into supervised, unsupervised and object-orientated classification techniques [9]. Semi-supervised classification techniques of hyperspectral images are used by Wang et al. [10] to classify non-urban regions of China, with a reported accuracy above 90%. Zhang et al. [11] compare land cover classification methods in an arid/semi-arid environment that demonstrates the ability of object-orientated classification techniques to differentiate between populated and non-populated areas. However, the recent focus of remote sensing applications has been to classify urban and built-up areas. Qiao et al. [12] demonstrate the use of classical techniques for object classification, such as a Support Vector Machine (SVM), to classify components of an urban area utilising high-resolution remote sensing imagery. Kanade et al. [13] demonstrate rapid mapping of urban regions for a high-density metropolitan city with varying built-up patterns, using remotely sensed data gathered from a synthetic aperture radar system. Li et al. [14] utilise aerial Light Detection and Ranging (LIDAR) with charge-coupled device (CCD) imagery to demonstrate an analytic hierarchy process for land area coverage classification, with the ability to differentiate between Urban and Non-Urban areas. Djamel et al. [15] evaluate classification schemes for urban area extraction using Landsat imagery and demonstrated that deep learning techniques outperformed traditional techniques for classification. Wen-mei Li et al. [16] compare a range of deep learning techniques to solve the environment-type classification problem in urban and built-up areas utilising remote sensing imagery. Man Li et al. [17] introduce hierarchical structuring to the land coverage classification problem to differentiate between land and water coverage from remote sensing imagery.

B. Place Recognition

Place recognition is the ability "to recognise the exact place despite significant changes in appearance and viewpoint" [18] within a map. Place recognition is commonly used throughout the literature to enable feature localisation within an occupancy grid [19]. Barros et al. [19] compare deep learning approaches for place recognition, utilising methods on the spectrum of supervised to unsupervised learning categories. This survey also explores end-to-end frameworks used to address a domain translation problem for place recognition, for which NetVlad [20] offer the most considerable impact. The survey presents three approaches to supervised learning, consisting of holistic, landmark and region based. From this, landmark-based supervised training effectively solves appearance change, perceptual aliasing and viewpoint changing. The techniques demonstrated by Sunderhauf et al. [21] employ convolutional neural network (CNN) feature extractors to develop semantic-based feature labels used for identifying potential landmarks in a visual feed. Rosinol et al. [4], and Lajoie et al. [22] utilise these extracted landmarks to generate a pose graph of key features within an environment.

C. Map Matching

Map matching techniques are required in SLAM algorithms, enabling updates to prior environment models. As Williams et al. [23] described, there are traditionally three main approaches: map-to-map, image-to-image and image-to-map. Integrating across a Gaussian distribution representation of a 6D voxels of an RGB-D sensor or point cloud system is one method to achieve spatial matching. Shuien et al. [24] detail a networked solution for feature merging through map alignment and data association for applications with SLAM solutions on a MAS. Yufeng et al. [25] present a semantic labelling system integrated with spatial map matching to generate a more effective SLAM result for loop closure. MurArtal et al. [26] demonstrate SLAM techniques that utilised critical frame solutions that allowed map matching to occur as a data insertion into a graph rather than the spatial matching techniques previously used. The architecture developed by Andersone [27] extended this, allowing for cross-platform map merging between platforms that did not share the same sub-mapping techniques through semantic representation for common understanding. Kong et al. [28] demonstrate a ConvNet landmark-based visual place recognition system that utilises sequence search and hashing-based landmark indexing, which significantly increases the efficiency of the map-matching process. The place recognition architecture designed by Garg et al. [29]conducts map matching through feature comparison by employing a semantic structure to generate an understanding of features. This system used CNN-based key-point matching, utilising semantic filtering and dense descriptor weighting to allow a place search procedure leading to a candidate match selection function.

D. Ontology and Symbolic Reasoning

Symbolic AI is the term used for several related AI methods that reason about problems using high-level human-understandable representations (symbols) [3]. As Jean-Baptiste [3] describes, an ontology is a "set of entities, which can be classes, properties, or individuals", representing "complex knowledge sets about things and their relations". One use of an ontology is to standardise the knowledge base of a specific domain and allow readability by humans and machines, as has been demonstrated by Baxter et al. [30]. Gomez-Perez [31] describes that the standard components of an ontology ($\mathcal{O}$) are its instances (I), concepts (C), attributes (a), and the relationships (R) between them. Axioms (A) are developed within the ontology to generate assertions to describe the overall theory of the ontology for its application, thereby incorporating domain-level knowledge within the ontology. An ontology can be defined as the five-tuple [32]:

$$\mathcal{O} = < C, R, a, I, A > . \tag{1}$$

An application of an ontology explored by Tenorth et al. [33] is to standardise the semantic representation of language services available to cognitive agents within a system and create an ecosystem where knowledge between agents and operators is shared. This semantic representation of objects within the symbolic domain is essential for a shared language between individuals, required for communication to exist within a system [30]. Furthermore, enabling this communications layer within a system furthers the system's capabilities to explain agent actions and hence provide explainability to an operator. Hepworth et al. [8] explore the concept of system transparency and operator understanding throughout the proposed *Human-Swarm-Teaming Transparency and Trust Architecture*, asserting that system interpretability and explainability are key facets underpinning the ability of a system to provide transparency for agent actions. Utilising these ontologies provides a method to share semantic knowledge, promoting bidirectional transparency to cognitive agents, be they artificial or human [32]. This transparency provides situational awareness of an autonomous agent's actions, decisions, behaviours, and intentions to other agents within the system [34]. Furthermore, as an ontology provides transparency and explainability within a system, it may enable agents' joint function within MAS to complete a central role [35]. Hence, an ontology may provide a semantic understanding of the world to the system, enabling a communication layer with cognitive agents through shared language services, thereby providing system transparency.

Many fields have applied ontologies to incorporate expert-level domain knowledge into symbolic solutions. Utilising these hierarchical ontologies for guided learning in sub-symbolic systems is a concept explored by Campbell [36]. This application enables a sub-symbolic system to reason on abstract concepts and reduce the dimensionality of a problem space (through partitioning) by applying prior knowledge to a learning system; see, for example, Hepworth [37], where a hybrid approach to activity recognition is detailed that fuses both data- and knowledge-driven (ontological) approaches to the activity recognition problem space in the machine learning domain. The RoboEarth framework [33], [38] proposed a system that inherently integrates a knowledge base with visual SLAM, allowing a more accurate representation of the environment and recognition of the objects found within. The Triplet Ontological Semantic Model (TOSM) [7] utilised short and long-term memory with a static ontology to create an on-demand ontological knowledge graph representing an environment in the symbolic domain. The abstract map data structure developed by Talbot et al. [39] incorporated symbolic spatial information (such as signs) into the SLAM problem to create "malleable spatial models for unseen places", producing navigation results comparable to the ability of humans. Control agents were utilised within the HST-3 architecture [8] to mitigate against heterogeneous knowledge sets produced by a swarm, which hinders convergence to a central solution. The current, most advanced ontology designed specifically for SLAM tasks developed by Cornejo-Lupa et al. [40] combines data from a range of ontology sets to achieve superiority at the domain knowledge, lexical and structural levels. The onto4MAT is the first attempt to design an ontology for multi-agent teaming [32] systematically. It enables an operator to provide an intent as tasks to a multi-agent system and for the agents to provide feedback to the operator.

E. Critical Assessment of Symbolic Approaches for Environment Type Classification

The present literature on environment-type classification relies heavily on remote sensing techniques to provide a high-fidelity overview of an area. Current technology in this field utilises techniques similar to those found within the object detection realm to classify environments through pattern recognition from these area overviews. However, little to no literature on symbolic approaches for environment-type classification. Furthermore, there is little to no literature on utilising SLAM techniques for environment-type classification. The SymboSLAM architecture introduces a method that combines the feature extraction strengths available to sub-symbolic AI architectures with symbolic approaches for reasoning to develop a system capable of conducting environment-type classification, thereby bridging this gap within the literature.

III. Methodology

A. Project Overview and Scope

The SymboSLAM architecture is a hybrid edge-driven context reasoning approach for use on a MAS of intelligent edge agents. This architecture produces environment-type classifications for an area, presented as 2D ground maps with features and landmarks represented symbolically. Environment-type classification is made possible by transforming the environment into a set of state-space variables that are semantically queryable. Subtasks 1-3 below show the general scheme for this transformation:

- T1 **Feature extraction.** This sub-task enables the system to extract useful information from the environment and generate an information pipeline. The edge agents on the MAS architecture achieve this through feature extraction and localisation.
- T2 **State Space Maps.** This task takes the information pipeline from T1 and generates useful semantically labelled maps. Both the edge agents at an individual level and the control agent at a collective level conduct this subtask. This task is deployed on a MAS and is achieved through map-matching / map-merging techniques.
- T3 **Ontology.** This sub-task takes the semantically labelled state space representations of the environment developed from T2 and formulates the labels for segments of the presently mapped environment. The control agent achieves this task through an ontology to achieve symbolic reasoning and hence enable transparency in the meaningfully labelled maps of the environment.

We propose a hierarchical structure with edge agents reasoning about their environment, employing sensor-based and visual data sources. The proposed architecture utilises a blend of sub-symbolic and symbolic reasoning techniques. First, sub-symbolic AI modules are used for object detection to achieve feature extraction. Next, the extracted features are semantically labelled and placed within a pose graph, transforming the environment into a set of queryable state space variables. Finally, a symbolic AI module for context reasoning then works to infer the environment type of segmented portions of a map on a collectively referenced coordinate system. The symbolic component queried to determine the environment type is an ontology.

B. *Edge Agent Search Method*

The simulated implementation utilises a random walk search strategy that extends the random-tree search described by Washburn [41] to introduce reactionary and long-range coordinated targeting strategies through movement incentives to achieve a more significant distribution level for higher area coverage. The proposed hierarchical control agent architecture shown in Figure 5 will coordinate the swarming movement of the edge agents through movement incentives similar to the shepherding behaviour presented by Hepworth et al. [42] to effectively disperse the search agents throughout an environment for target search and SLAM. A human operator was utilised as the cognitive agent for search methodology to instantiate the edge agent architecture on a physical device.

C. *Feature Extraction*

The CNN feature extraction submodule detects 56-class objects selected to model the Canberra landscape across the seven possible environment types available. Feature abundance in the environment and access to objects in existing datasets were the determining factors for selection into the SymboSLAM custom dataset. Image-classification pairs were taken from 10 of the most popular datasets available. These were combined with the SymboSLAM dataset to generate 200k, 12k, and 6k labelled images for training, validation and testing.

D. *Place Recognition*

A landmark-based technique is employed at both the edge agent and control agent architecture level to achieve the place recognition component of the SLAM problem whilst building a queryable state space representation of the environment. Ground vehicles controlled by the control agent architecture explore within an environment employing the random walk search strategy described above, conducting locally-scaled SLAM tasks. Each edge agent constructs an individual map of its environment, utilising onboard Spatio-temporal sensor feeds. The simulated ground vehicles utilise a camera, Inertial Measurement Unit (IMU), Time of Flight (ToF) and Lidar sensors, sequentially timestamped; the implementation on the android application uses a camera and GPS. Each edge agent implements a feature extraction submodule described above. The edge agent architecture then utilises semantic knowledge of these extracted features to sub-categorise objects into either dynamic or static objects. Landmarks within sub-maps used for matching represent the collected static features, and both the static and dynamic objects are utilised in later recognitions for environment-type classification. The depth sensor is then utilised for the simulated ground vehicles to determine the closest of the static targets, and the edge agent will then navigate towards it. The edge agent continues on its path towards this target, attempting to minimise the proximity to the target whilst maintaining the target within the entire frame of the camera feed, as demonstrated in algorithm **??**. Once the distance between the agent and the target is minimal, the edge agent utilises its onboard depth sensor to determine the feature's location relative to the agent's location. Finally, the feature is inserted into the edge agent's individual map data structure as an element within a pose graph.

E. *Map Matching*

A map-to-map, map-matching technique utilising landmarks as key features within a pose-graph data structure is the technique featured in both edge and control agent architectures for individual and collective mapping. Figure 2 demonstrates this functionality at the control agent level to conduct map meshing. The control agent receives and maintains the individual maps for each edge agent. Each of these individual map data structures is referenced to begin at $(x, y) = (0, 0)$. The control agent then amends the received individual maps to reference them on the collective map referencing plane, as the control agent knows the true starting location of each edge agent. The edge agents' map features are then merged onto the control agents' map utilising the landmark description and believed place. The control agent compares the landmark descriptor of each new landmark received with all features within some radius a within the control agent's map to determine whether they are semantically similar. If they are semantically

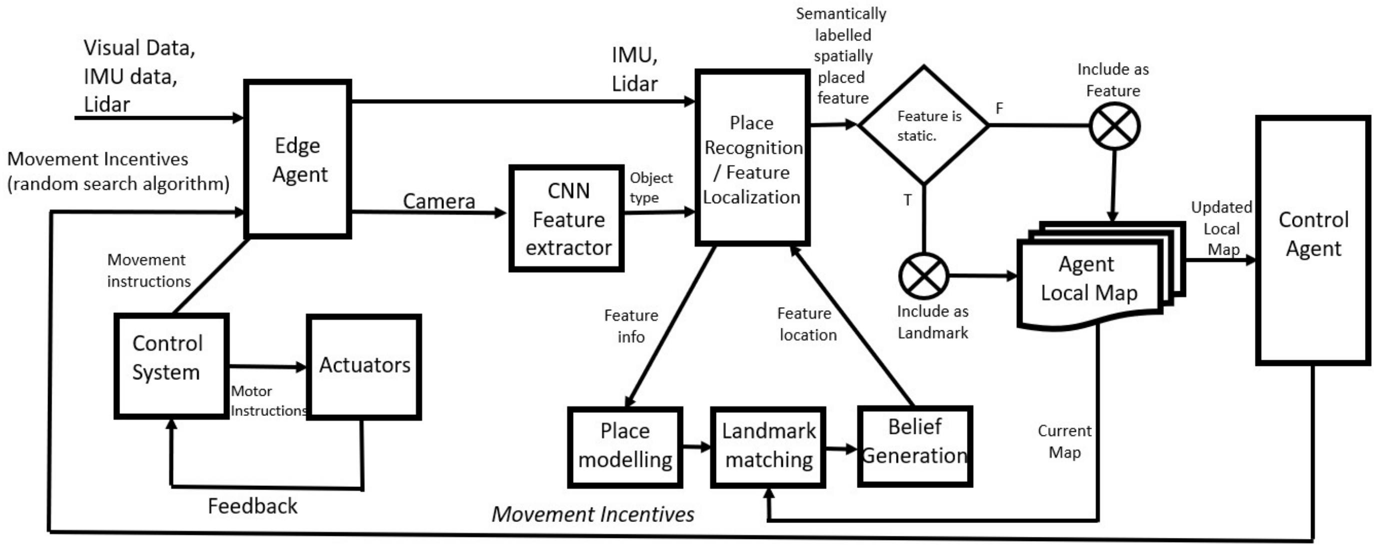

Fig. 1: SymboSLAM Edge Agent architecture for a simulated robotic platform. Note that the physical instantiation of an edge agent holds the same basic topology but utilises GPS instead of an IMU.

similar, the feature with the highest confidence is the new landmark, and this landmark position is the average of the two previous landmark positions. Suppose the two landmarks are not semantically similar, depending on the tolerance of feature proximity and the feature confidence. Then, the new landmark will either be added to the pose graph as a new entry or discarded. The discarded landmarks are stored, and if there are greater than two discarded landmarks within close spatial proximity (tolerance again specified by operator) with similar classes, then the landmark at that position will be updated with the discarded feature.

F. SymboSLAM Ontology

The SymboSLAM ontology, depicted in Figure 3, is designed for the SymboSLAM architecture. This ontology has domain-level knowledge for environment contextualisation and subsets of the Onto4MAT [32] and OntoSLAM [40] ontologies for swarm control and SLAM problems. This ontology enables bidirectional communication between all cognitive agents (edge, control, operator) by allowing the system to conduct SLAM tasks with symbolic representations. A single control agent is used to ensure that the explainability of a solution is converged upon, enabling transparency in the developed system to foster trust by the operator.

G. Semantics Engine

Figure 4 shows the submodule responsible for environment classification through the contextualisation of the features found within a segment. For each segment of the partitioned map, this module takes the feature class instance, feature class confidence and spatial distances between each feature over a segment. It outputs a probability distribution of environment types for which the maximum probability within this distribution is this segment's environment type. The SymboSLAM ontology is queried with a feature class instance to determine the environment type superclass and the semantic proximity to each environment superclass; note that multiple environment types may be returned. This module intends to reward:

1) semantic closeness of feature class and environment superclass;
2) spatial closeness of feature classes with alike environment superclasses; and
3) a larger number of inferences made between feature classes within a segment, for instance, more feature classes identified with alike environment superclasses.

The semantics engine module penalises the spatial closeness of feature classes with dissimilar environment superclasses. For each feature in a map segment, the semantic proximity for each possible feature environment super class is $SP(e_n, f_x)$. The distances between each feature class within the segment is then calculated to be $d(f_x, f_y)$ and each distance is normalised to the maximum possible distance of each segment such that $d(f_x, f_y) \sim N(0, 1)$. The confidence that each segment is of environment n is calculated by summing the normalised distance between each feature node of that environment type weighted by the confidence of each feature class:

$$C(e_n) = \sum_x \sum_y (SP(e_n, f_x)\cdot f_x + SP(e_n, f_y)\cdot f_y)\cdot(1 - d(f_x, f_y)) \tag{2}$$

The environment confidence is calculated for each environment n contained within the SymboSLAM ontology to produce a probability distribution across all possible environment types for each segment. The confidence of environment type is then normalised to the number of inferences made between the features f_x, f_y for each environment confidence calculation so that $C(e_n) \sim N(0, 1)$. Equation 2 above solves reward intents one and two listed above. The semantics engine then calculates a weighted sum for the environment confidence $C(e_n)$ and the number of inferences made for each environment confidence calculation normalised to the maximum number of inferences

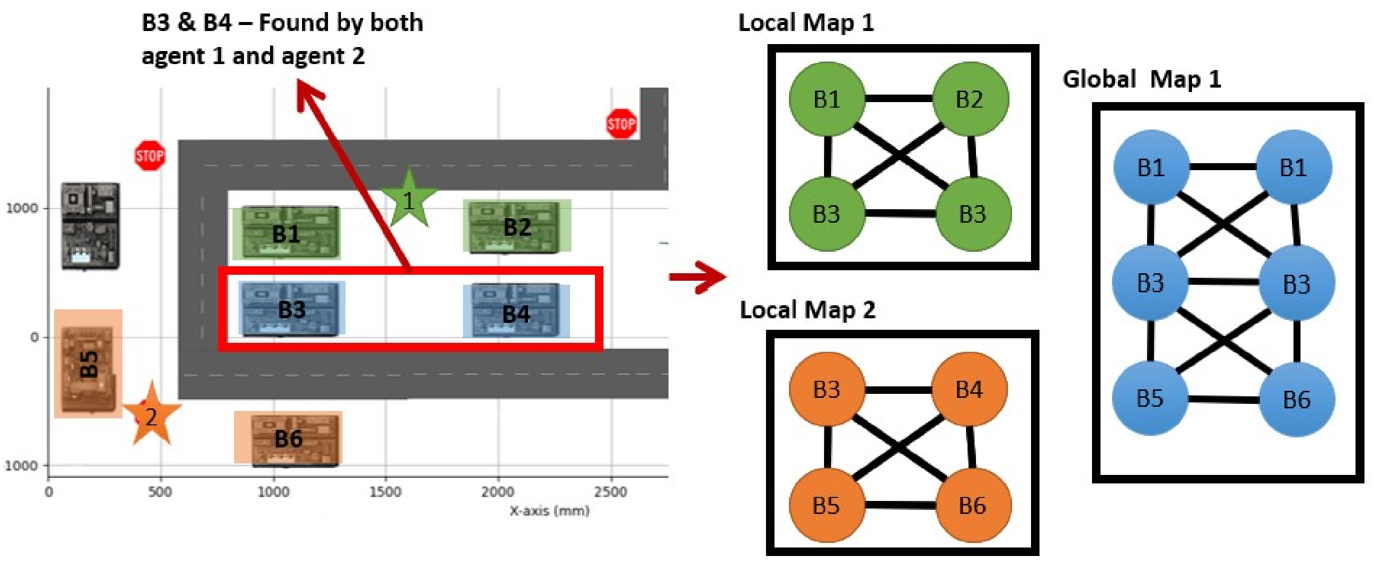

Fig. 2: Map matching technique employed by the SymboSLAM architecture at both an individual and collective level. Note that the nodes shown within each map are an entry within a pose graph and then displayed to the operator as a point on a 2D occupancy map.

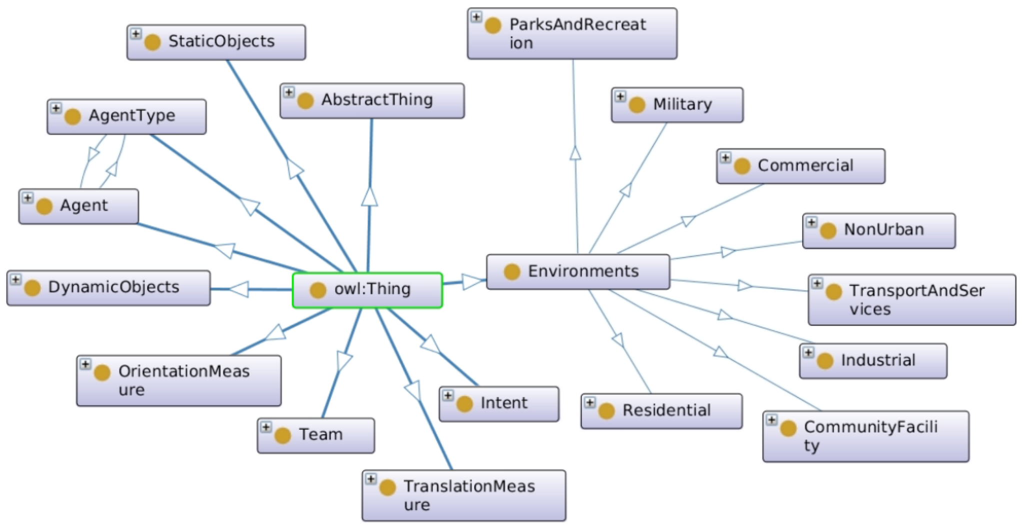

Fig. 3: SymboSLAM ontology. Environment types entries reflect the Canberra region. The SymboSLAM ontology also features concepts from the OntoSLAM and ONTO4MAT ontologies to enable more effective SLAM and swarm control functionality, respectively.

allowable for a set of segment features. This calculation solves reward intent three and produces a probability distribution of the likeliness of environment type. For example, for a segment with z features, this is given by:

$$P(e_n) = \frac{1}{a} \cdot C(e_n) + \frac{1}{1-a} \cdot \frac{e_n(\text{inferences})}{z(z-1)} \tag{3}$$

Where $a, C(e_n), P(e_n) \sim N(0,1)$ and a are specified by the operator depending on the sparsity of feature classes within a map, as some environment types may inherently have a low number of features, e.g. a desert environment.

H. Grid Map Segmentation

The control agent attempts to increase the environment-type classification probability through map segmentation methods. Figure 6 depicts the functionality of this module (left) and shows the output of the segmentation process (right), Which the semantics engine module then utilises. The control agent first identifies nine critical regions of interest (ROI) within the map. The semantics engine then conducts environment-type classification using this updated segmented map data structure. If the probability distribution for environment type classification does not return above a specified threshold, then

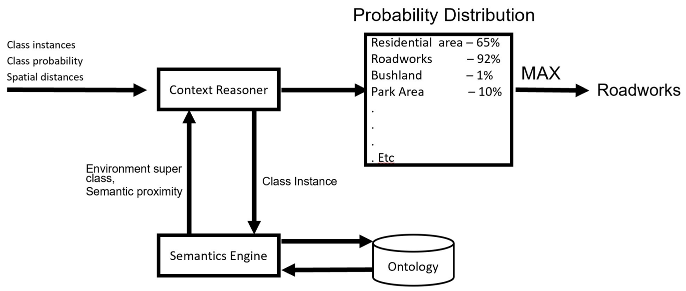

Fig. 4: SymboSLAM symbolic reasoning modules. The ontology featured is the SymboSLAM ontology.

the segment is further partitioned into four quadrants, as shown in Segment-$(2, 2)$ of Figure 6. This process is repeated until one of three conditions are met. The first is if the quadrant of a segment returns an environment classification confidence above a specified threshold, and the next is if the quadrant is empty. The third is if the sparsity of information within the quadrant reduces classification performance.

I. N-Nearest Neighbour Map Segmentation

The second segmentation methodology implemented utilises the environment probability calculation from the semantics engine. This segmentation methodology is based on the K-nearest-neighbour algorithm and is a novel progressive clustering algorithm. The user supplies this algorithm with a seeding position from the control agent's map (where to begin clustering) and a measure of momentum (how many landmarks the algorithm skips between clustering), from which the algorithm will conduct parses of the control agent map. The branch segmentation method then uses the three landmarks closest to the cluster centre to determine its environment probability distribution. The number of landmarks within a cluster is incremented by one until the prior environment probability stored is larger than the posterior environment probability.

The N-Nearest neighbour algorithm aims to cluster the entire landmark map into N previously unknown fragments, each with a unique number of landmarks. The branch segmentation algorithm then compares the neighbours of these N fragments to merge neighbours with the same environment classifications. The above algorithm essentially takes some number of Landmarks L within a Map M:

$$M = \{L_0, L_1, ...L_{\text{end}}\} \text{ , where } L \in \mathbb{R}^D \tag{4}$$

And places them into a fragmented map set:

$$M = \{F_0, F_1, ...F_z\} \tag{5}$$

Where $F = \{L_0, L_1, ...L_N\}$ and $F_{\text{env}} = \text{argmax}(P(e_n))$

To ensure fragmented sections of the map did not bisect, the control agent used a modified SVM algorithm to create decision boundaries surrounding clusters. As map fragments may have complex shapes, landmarks were compared as pairs to check for bisects with landmarks from neighbouring fragments. The modified SVM algorithm used the landmarks immediately surrounding the bisection portion of neighbouring fragments to draw the decision boundary. The control agent then coordinated trade between neighbouring clusters based on these decision boundaries.

$$\text{For } F_a \rightarrow L_{a_1}, L_{a_2} \text{ and } F_b \rightarrow L_{b_1}, L_{b_2} \tag{6}$$

$$L_{a_1}, L_{a_2} \rightarrow \text{line}_1, \text{ and } L_{b_1}, L_{b_2} \rightarrow \text{line}_2 \tag{7}$$

Then given line_1 and line_2 bisect, a decision boundary can be drawn between two sets of landmarks from fragments A and B:

$$L \in \mathbb{R}^D \text{ , } \phi : \mathbb{R}^D \rightarrow \mathbb{R}^M \tag{8}$$

$$\phi(L) \in \mathbb{R}^M \tag{9}$$

$$H : w^T \phi(L) + b = 0 \tag{10}$$

And the distance between the decision boundary and the landmarks within a segment's immediate neighbour cluster (excluding the bisecting points) is found with:

$$d_H(\phi(L_0)) = \frac{|w^T \phi(L_0) + b|}{||W||_2} \tag{11}$$

The decision boundary is then updated to maximise the minimum distance from the decision boundary hyperplane to each landmark point in a map segment according to the following:

$$W^* = \underset{\text{w}}{\text{argmax}} \frac{1}{||W||_2} [\underset{\text{q}}{\text{min}} \cdot y_n \cdot [W^T \phi(L_q) + b]] \tag{12}$$

J. Full SymboSLAM architecture

The SymboSLAM architecture synthesises practices from the SLAM, swarm and symbolic domains to conduct environment-type classification in a novel manner. The SymboSLAM architecture transforms the environment into a query-able state space representation through a Multi-Agent System featuring a hybrid reasoning orientation. The SymboSLAM architecture utilises intelligent edge agents capable of processing information to collect data and infer knowledge about their environment. These edge agents apply ontologically backed semantic labels to extracted features, thereby generating symbolic representations of their environment. Sub-symbolic methods for feature extraction are used in conjunction with place recognition techniques to generate area overview maps with semantic markers embedded within the spatial information provided. Map matching techniques are then incorporated within the architecture to merge these semantically labelled individual maps produced by intelligent edge agents into a central structure. A control agent collates these individual maps and generates a collective map to infer knowledge about the environment. An ontology is then featured within a symbolic reasoning approach to take the semantically labelled representations of the features within this central map structure and produce a 2D map of environment-type classifications.

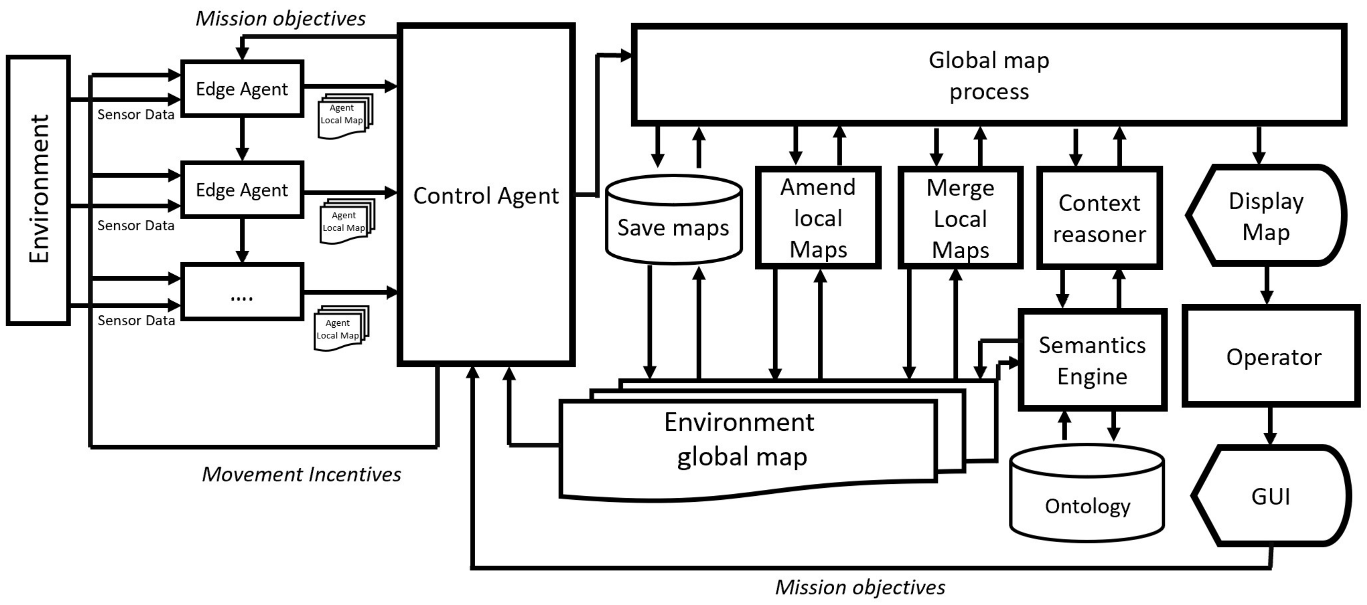

Fig. 5: SymboSLAM control agent architecture.

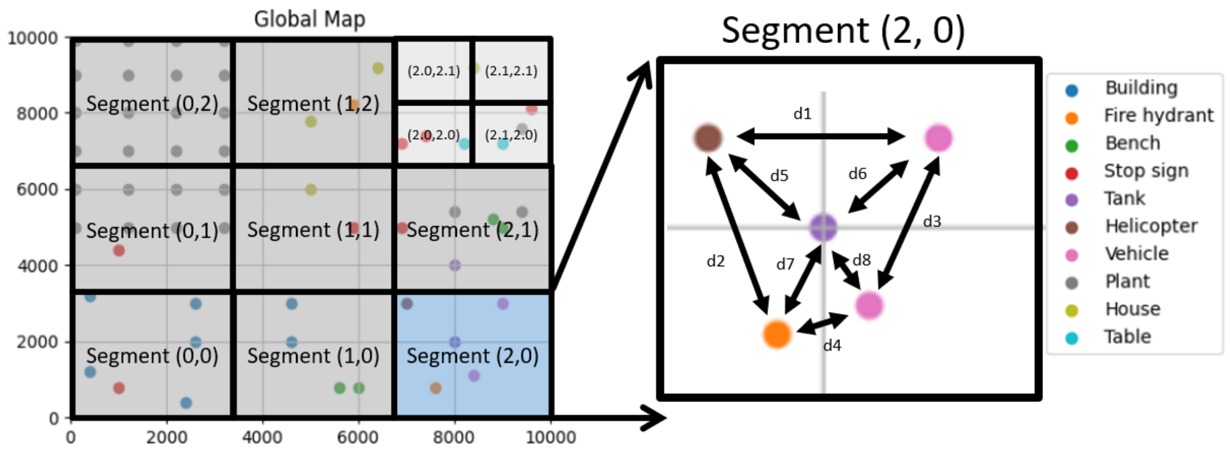

Fig. 6: Segmentation process for map partitioning - Segment (2,2) shows further partitioning where the confidence of environment classification is less than a determined threshold

.

IV. Architecture Evaluation

Nine metrics were selected to evaluate the SymboSLAM architecture, as shown in Table I. These evaluation metrics reflect the specific domain from which each submodule was derived. The evaluation metrics are described below. The studies shown in Table I reference instances from work published within the same domain as the submodule being evaluated for which these metrics have previously been used.

A. Multi-Agent System

The area coverage, area dispersion and time taken by the architecture to complete are the metrics used for evaluating the multi-agent system component of the SymboSLAM architecture. These evaluation metrics were tested in simulation only, as no robotic edge agent platforms were used for physical instantiations. A'/A scoring as described by Washburn [41] is the evaluation metric for area coverage, where $A' = VWt$ is the searched area of the maximum possible searchable area A. The Agent dispersion from the global centre of mass, as described by Abbass et al. [45], was taken as the success metric area dispersion. The dispersion was measured at intervals of 1 second throughout the simulated trials, and an average of this data was taken to determine agent area dispersion. The simulated trials concluded when all objects were discovered by an edge agent within the environments presented. The measure of area coverage, area dispersion and time taken to complete was calculated after the simulated trial.

Architecture Evaluation metrics - Studies and their respective metrics	
Type of metrics	Studies
Multi Agent System	
Area Coverage - A'/A scoring	[41], [43], [44]
Area Dispersion - average dispersion measure	[42], [45]
Time - average time to complete	[4], [5], [7]
Simultaneous Localisation and Mapping	
Feature Extraction - mAP	[46]–[48]
Place Recognition - ground truth comparison	[4], [21], [22], [49]
Map Matching - average error of centre offset	[28], [29], [50]
Symbolics Engine	
SymboSLAM Ontology - OOPS!	[32], [51], [52]
Map Partitioning - IoU / Kappa Coefficient	[11], [15], [16]
Area Type Classification - AP	[10]–[13], [16], [17]

TABLE I: Nine metrics utilised to evaluate the SymboSLAM architecture with published studies that have previously used these evaluation metrics

B. Simultaneous Localisation and Mapping

As the feature extraction submodule consists of a CNN object detection system, the intersection over union (IoU) and the mean average position (mAP) for the feature extraction submodule were tested as this unit's first evaluation metric. The IoU for each bounding box for a detected object class in a frame is:

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{X \cap Y}{X \cup Y} \tag{13}$$

Where X is the detected feature bounding box, and Y is the ground truth. The precision and recall of the model to detect objects are given by

$$\text{Precision} = \frac{TP}{TP + FP}\ , \text{Recall} = \frac{TP}{TP + FN}, \tag{14}$$

where TP & FP = true & false positive, FN = false negative
The average precision is determined by finding the area under a precision-recall curve when conducted over several prediction outputs:

$$AP = \int_0^1 p(r)\, dr \tag{15}$$

And lastly, the mAP is found by taking the mean of the AP over all the feature classes for which the model is trained.

$$mAP = \frac{AP_1, AP_2, ...AP_n}{n} \tag{16}$$

The SymboSLAM feature extraction module is evaluated against the testing dataset of 6k images. The place recognition component of the SLAM problem is assessed through ground truth map comparison, where maps are evaluated using matched topology graphs. As the ground truth and generated maps are already 2d topology graphs, their evaluation in the method described above can be directly applied. Next, the coverage (percentage of matching vertices between ground truth and generated map) and global accuracy (Correctness of positions of the matched vertices in the collective reference frame) were determined and used as the metric for place recognition. Lastly, the average error of centre offset is the evaluation metric used for spatial consistency of matched landmark pairs. The average error of centre offset is a sum of squared error measure for the distance between the believed location and the actual location of all landmarks within a generated occupancy map:

$$Er = \frac{1}{L}\sum_{i}^{L-1}\sqrt{(L_{ix} - L_{(i+1)x})^2 + (L_{iy} - L_{(i+1)y})^2} \tag{17}$$

Where L is a landmark within a map.

C. Symbolic Engine

The SymboSLAM ontology was evaluated using OOPS! [51][1] as demonstrated by Hepworth et al. [32]. This evaluation saw the checking of requirements and competency questions and testing of the ontology in the target application environment. The IoU and the AP between the generated 2D environment-type classification maps and the ground truth environment-type maps are calculated for all 16 simulated and measured trials. The IoU was used as the correlation coefficient (similar to a Kappa coefficient) to determine the inter-rater reliability between the generated and truth environment type map. The AP was used to determine the accuracy of the symbolic engine of the SymboSLAM architecture.

D. Simulated Evaluation

There is "little prior work on symbolic navigation of unseen places and no relevant benchmarks for evaluating performance" [39] and less still on applying symbolic or sub-symbolic AI to the environment contextualisation problem. Therefore, simulated environments incorporated expert domain-level knowledge for classifying an environment type through contextualisation using the feature classes found within. The SymboSLAM architecture and ontology were

[1]https://oops.linkeddata.es/response.jsp

deployed on the EyeSim robotics simulation software, employing three edge agents and one control agent to conduct environment contextualisation in real time. In addition, the SymboSLAM SLAM modules incorporated an oracle system to offset the errors from the simulated sensor feeds introduced by the Unity physics engine, causing a cascading effect due to the quasi-random dispersion SLAM algorithm. Six areas were simulated from the Canberra region to evaluate the SymboSLAM architecture within the simulated trials. These areas include Gunghalin (mix of environment types), the airport area (primarily non-urban), Fyshwick (industrial principally), Kingston (primarily commercial and high-density residential), the Train Depot area (mainly transport and services) and the Civic City precinct (commercial principally). Each simulation contained up to 56 separate 3D object classes, some classes with multiple models and all features with various instances.

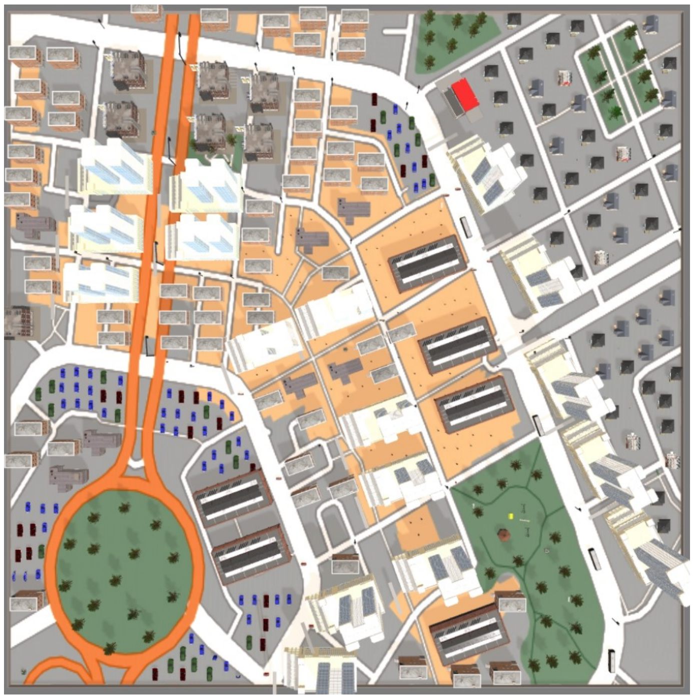

Fig. 7: Example simulated area using the Unity 3D environment, depicting a region of Civic in Canberra.

E. *Real World Experiments*

Two real-world experiments were conducted using the SymboSLAM mobile phone application, acting as edge agents to collect landmark information. These experiments were conducted throughout the Civic and Gunghalin areas to showcase the functionality of the architecture in a real-world setting.

V. Results

A. *Multi-Agent System*

Table II shows the results for simulation area and area coverage, average and normal dispersion and time taken for the simulation to conclude. Note that the normal dispersion was calculated by taking the average dispersion of all edge agents from the GCM (column 5) and normalising it to the maximum theoretical dispersion to the GCM.

B. *Simultaneous Localisation and Mapping*

The 6k image testing dataset was utilised to evaluate the full-scale and tiny-scale CNN feature extraction module for the simulated and real-world SymboSLAM instantiations, respectively. The results for mAP, IoU and inference time are shown in Table III. A demonstration of the full-scale simulated instantiation is shown in Figure 8 (left) and an example of the tiny-scale real-world instantiation is shown in Figure 8 (right), and a full set of simulated and real-world results are available online [2]. Figure 9 shows the average error of centre offset for discovered landmarks. The exponential style in which this error accumulated led to the need to utilise an oracle system to test the functionality of the remaining SymboSLAM submodules.

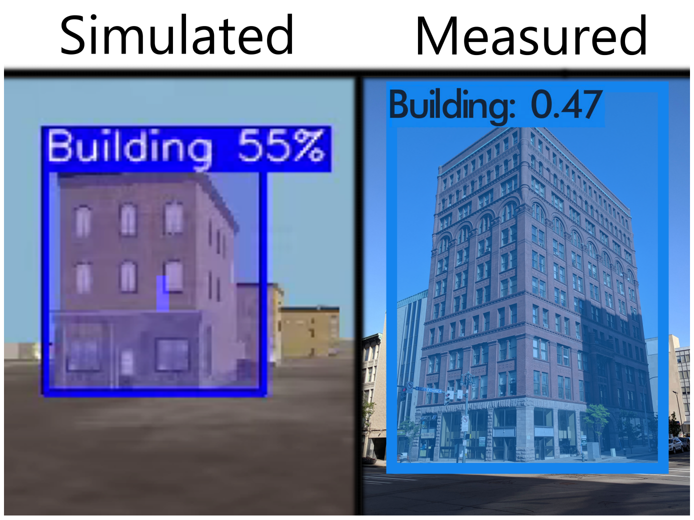

Fig. 8: CNN Feature extractor predictions output from simulated (left) environment and real-world (right) environments

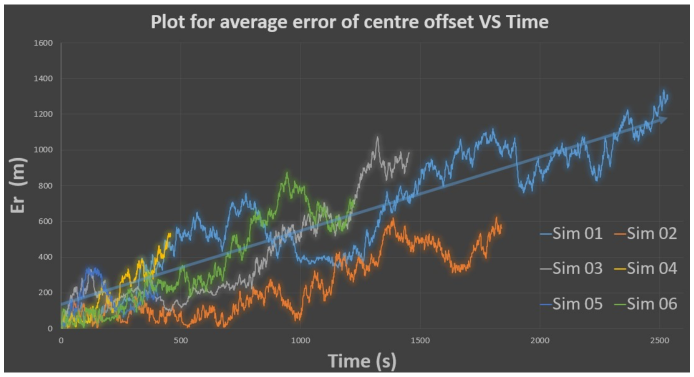

Fig. 9: The average error of centre offset for Landmarks discovered by Edge agents when conducting map matching with ground truth maps for Simulations 1-6

C. *Symbolic Engine*

The SymboSLAM ontology was run through the Ontology Pitfall Scanner, OOPS!. Fourteen minor pitfalls are present

[2]https://cloudstor.aarnet.edu.au/plus/s/0T0pZFYgeyMgMyP

MAS Metric Results						
Simulation trial	Total Area (A, km^2)	area Searched (A', km^2)	Area Coverage(A'/A)	Avg Dispersion (km)	Norm Dispersion	Time (mm:ss)
01 - Gunghalin	29.5609	26.4414	89.44%	2.9945	77.88%	84:14
02 - Airport	27.8891	20.7748	74.49%	2.6738	71.60%	43:39
03 - Fyshwick	6.6667	5.5463	83.19%	0.8567	46.92 %	30:14
04 - Kingston	0.5041	0.4423	87.74%	0.2432	48.44%	14:37
05 - Train Depot	1.0976	0.9777	89.07%	0.4452	60.10%	12:43
06 - City	1.8225	1.7943	98.45%	0.3098	32.45%	40:22

TABLE II: 3 EyeSim robots were placed at equidistant points within each simulation listed. These edge agents collected feature information and were controlled by a control agent. The simulation concluded when all features from within the environment were discovered

CNN Feature Extraction Results			
	mAP	IoU	Inference time (ms)
SymboSLAM-Full	43.6	0.6545	154
SymboSLAM-Tiny	17.1	0.5877	842

TABLE III: Feature extraction results for Full size and tiny size of SymboSLAM CNN object detection modules. Results for full size obtained using NVIDIA GeForce GTX 1080 GPU with Intel(R) Core(TM) i9-7900X CPU OS Linux. Results for tiny size obtained using CAT s64 Pro Android Smartphone Adreno 512 GPU Octa-core CPU OS Android 10

SymboSLAM Simulated and Measured Results					
	Grid		Branch		
	AP	IoU	AP	IoU	# Features
Sim 01 - Gunghalin	48.43	0.52	38.72	0.46	3694
Meas 01 - Gunghalin	8.51	0.27	1	0.01	2209
Sim 02 - Airport	75.52	0.84	64.20	0.60	199
Sim 03 - Fyshwick	28.99	0.73	72.40	0.65	152
Sim 04 - Kingston	33.85	0.90	41.20	0.75	212
Sim 05 - Train Depot	8.16	0.90	31.99	0.31	167
Sim 06 - City	72.05	0.78	80.89	0.89	429
Meas 02 - City	19.27	0.48	1	0.01	833
Overall	AP		IoU		
Simulated	49.7		0.69		
Measured	7.4		0.19		

TABLE IV: IoU and AP for Full SymboSLAM architecture to generate 2D topo maps of environment types. Simulated results utilised an oracle system, and measured results utilised GPS.

within the ontology, but no critical or important pitfalls were detected. The SymboSLAM ontology was then used with the complete SymboSLAM architecture to conduct both simulated and real-world trials, as shown in 10 and 11 (the remainder of simulated trials can be found in the appendix). The IoU and AP for the six simulated and two real-world trials are shown in Table IV. As the granularity of the 2D topographic environment maps generated by the SymboSLAM architecture could be segmented into a maximum of 24x24 partitions, the IoU and AP for the trials were calculated using all 24x24 segments for each generated environment map.

VI. Discussion

A. Multi-Agent System

Table II shows that simulations with a greater level of complexity and a higher number of features within the environment required a higher percentage of area coverage from the entire space available. This phenomenon is most noticeable in the area coverage difference between the Airport and City simulation, wherein an area of $\sim$ 74% was required for seven items per square km for the Airport, and $\sim$ 98% was required for 238 items per square km for the city. Hence, the quasi-random search strategy is most effective for sparsely populated environments. Still, more complex environments would likely benefit from implementing more thorough search strategies such as a simple coordinated grid method. The time to finish, and normal dispersion, are a reflection of both simulation area size and complexity. The results from Table II describing this are hardly surprising (larger area leads to more time and greater dispersion), but this does demonstrate that the long-range coordinated targeting strategies implemented are functioning as intended.

B. Simultaneous Localisation and Mapping

The trained YOLOV4 module used for feature extraction achieved a mAP of 14.3% and 6.6% lower than the scores achieved by the same network when trained on the MS COCO datasets [47] [3]. The reduction in mAP and IoU is due to the reduction in images within the training dataset (328k in MS COCO to 200K in SymboSLAM). Other factors that may have affected the mAP and IoU could have included dissimilar feature types causing an underfit for low-level filters within the model, and the reduced number of feature classes reducing the complexity of the model. Updating the CNN feature extractor [4] would likely improve the mAP and IoU of the module; however, the results achieved with the custom dataset were sufficient for the SymboSLAM architecture. Figure 8 demonstrates the effectiveness of the trained feature extractor model in both simulation and the real-world environment.

The average error of centre offset for landmarks detected by edge agents, shown in Figure 9, quite clearly indicate that the SLAM component of the SymboSLAM architecture does not yet function as intended. The recorded data shows a linear relationship between increased error and simulation time due to the compounding nature of the error in location measurement. The maximum recorded error in distance between landmark true and believed the location was almost 1.4 km, which is well out of tolerance for the SymboSLAM system requirements. The observed failure of the SLAM module is due to the error present in the Unity (and, by extension, EyeSim) sensors such as ImU and ToF. Compounding location error is a

[3]https://pjreddie.com/darknet/yolo/

[4]YOLOv7 was released when this article was written - see https://paperswithcode.com/sota/real-time-object-detection-on-coco for the latest models available

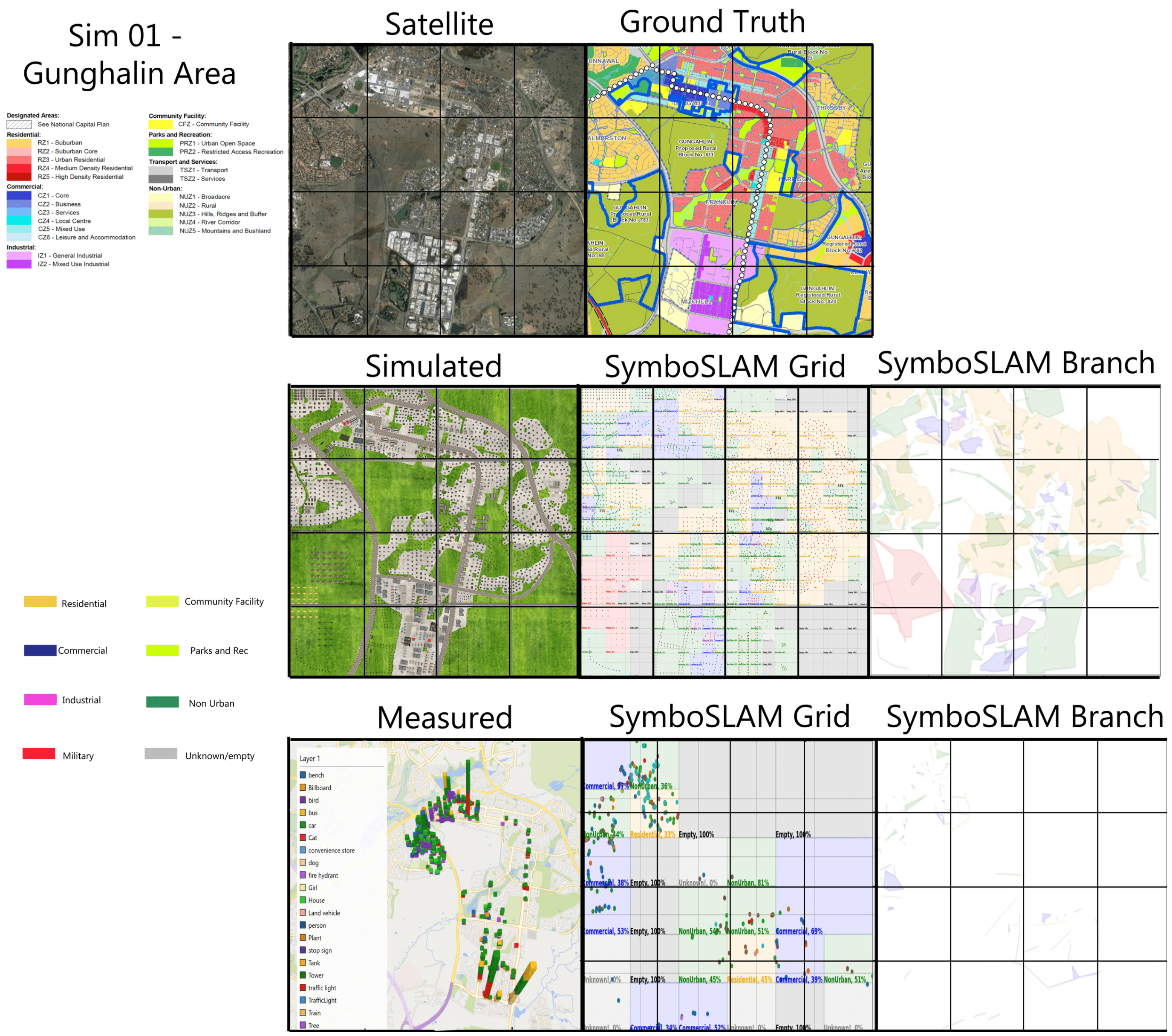

Fig. 10: Comparison of ground truth and SymboSLAM results for ACT Gunghalin simulated and real-world area

problem-set present in the current literature for robotic SLAM systems [19]. At the outset of the SymboSLAM project, it was hypothesised that this problem would be negligible, as the granularity for feature extraction was much larger than most SLAM systems currently available, which was, however, not the case.

As such, to ensure the system's functionality and allow testing of other components, an oracle system was utilised to reveal landmarks on the map for which edge agents had discovered that were within a tolerable distance threshold to the true location of the landmark. The location of this landmark was then utilised by the oracle system to essentially 'reset' the average error of centre offset for that edge agent's map by adjusting the edge agent's landmark map to align better with this known location. Utilising a simulator / real sensors with a much lower measurement error will alleviate this issue for small-range areas (less than 1 square km) that are not complex (less than ten items per square km) but are not a viable long-term solution for the system. Many architectures offer loop closure as a solution for SLAM error issues [26], [53]. Notwithstanding, studies into this area have not yet generated a system capable of amending maps on a scale posed for the SymboSLAM problem. GPS offers a viable solution to this issue for both complex and large areas, as demonstrated when the system was deployed in the real environment in experiments 1-2. Relying on a sensor such as GPS for the SLAM module of the SymboSLAM system does, however, heavily limit the range of applications for which the system can be applied. Utilising the SymboSLAM architecture with GPS on a scale conducted in the eight trials will produce viable results.

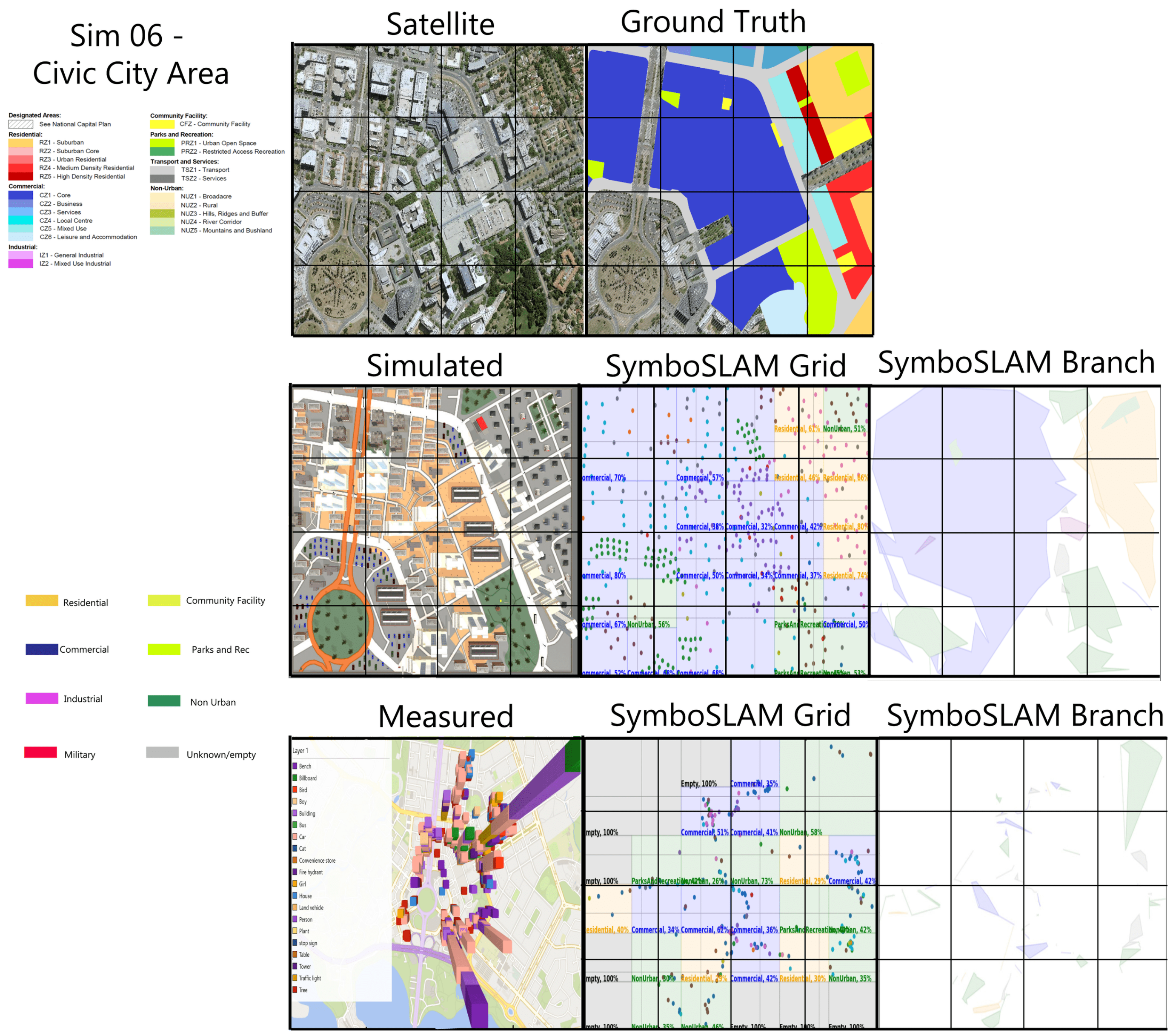

Fig. 11: Comparison of ground truth and SymboSLAM results for ACT Civic simulated and real-world area

.

C. Symbolic Engine

An AP and IoU of 49.7 and 0.57 for the 12 simulated trials and 7.4 and 0.19 for the measured trials were achieved for the environment type classification utilising the SymboSLAM system. From Table IV, a clear relationship between the AP and the number of features is observed across the simulated results in that more features generate a higher AP of environment classification. An outlier to this relationship is the Airport which achieved the highest AP due to the ground truth of this area consisting of primarily non-urban areas, which is likely an oversight by the ACT government planning commission. By comparison, the sparsely populated area contained fewer community facility features than non-urban features, i.e. the sim had more trees than planes which highlight a shortcoming with the symbolic component of the SymboSLAM system - it does not take into account feature importance. When determining the classification of an area, some features are more important than others; for example, a skyscraper is indicative of only a city environment, whereas a tree could lead to many environments, as it is common in many areas.

Table IV also shows that the branch segmentation method outperformed the grid segmentation method for all trials except for Gunghalin due to the increased white space (unknown area) introduced by the branch segmentation method. This white space is present in all trials but is most noticeable across the two measured trials in the Gunghalin simulated trial. Whilst this may be seen as a downfall of the system, it can also be observed as an advantage, as the branch segmentation method does not make assumptions across areas for which edge agents have not yet explored or for which there are no observable features. Theoretically, the IoU obtained from the branch segmentation method should be lower than the grid segmentation method, as the rigidity of the boundaries for this method is much more flexible. However, a higher IoU when using the branch segmentation method is not observed within the results of Table IV except for the city simulated trial. Feature abundance is again the reason for a decreased IoU within the branch segmentation method, as the boundaries between environment types are not clearly defined throughout the remainder of the simulated environments. The AP and IoU of all measured experiments were substantially lower than the simulated counterpart across both the grid and branch segmentation methods.

Figures 10 and 11 show a tendency for features to be clustered around streets and walkways due to the increased complexity involved with feature extraction from real-world environments. Space is present between all the features within the simulated environment as features extracted by the edge agents were placed on the control agents map at a level of homogeneity not achievable in the real world. Unknown environments are also much more prevalent within the measured results due to restricted and limited access areas hindering data collection and are most prevalent in the branch segmentation trials for the measured data collected. Hence, the IoU and AP obtained will increase with increasing feature abundance and perform best with the grid segmentation method. This increased feature abundance can be easily achieved by altering the simulated models but would likely require another platform for data collection for measured results. Mounting the current platform to an aerial vehicle will likely alleviate this issue as it would obtain higher freedom of movement throughout the environment and, as such, would be able to collect data more homogeneously than the current method.

VII. Future Work

A. Landmark Representation

A more accurate representation of landmarks should be incorporated into later iterations of the SymboSLAM architecture. Research looking to further this area may incorporate more traditional methods for SLAM to achieve this.

B. Human Level Representation of Areas

Incorporating the ability to understand human navigational aids such as signs [39] will increase the feature extraction ability of the SymboSLAM architecture and allow for a much richer depth of information to be accumulated about environmental features. For example, the SymboSLAM feature extraction modules can presently determine if an object is a building, but distinguishing between the types of buildings within a city is a task achievable through the understanding of signage.

C. Simultaneous Localisation and Mapping

The issues identified above within the SLAM modules of the SymboSLAM architecture must be addressed, and the solutions presently being investigated throughout the SLAM literature do not offer a viable solution for the scale of the problem posed in this paper. However, exploring the current oracle modules utilised for the SymboSLAM architecture may solve the SLAM problem. For example, a solution wherein landmarks are used in a resection formation to enable proper landmark belief generation, and subsequent confirmation through the map-matching process may be a workable solution.

D. Hierarchical Chaining

Building a belief generation of areas at a more granular level to be chained hierarchically may assist with improving the robustness and reliability. Increased explainability through an extended hierarchical chaining process may also enhance the trustworthiness of classified areas.

VIII. Conclusion

This paper proposes a novel approach to symbolic SLAM that uses symbolic reasoning through ontological design to create 2D environment-type maps through a multi-agent system with a hybrid orientation for contextual reasoning. The proposed system makes use of an intelligent edge agent architecture as well as a control agent architecture. Edge agents conduct local SLAM tasks within the environment through a random walk search strategy that extends the random-tree search method to introduce reactionary and long-range coordinated targeting. Intelligent edge agents semantically label extracted features and localise them spatially through place recognition, transforming the observed environment into a queryable set of state space representations. A control agent then receives many edge agents' maps to amend these maps (given the edge agent starting position) by conducting map matching techniques through a landmark map matching methodology to create a semantically labelled map of the environment. The semantics engine then takes in segmented partitions of these maps and utilises a purpose-built ontology to generate a probability distribution of likely environments. The SymboSLAM architecture was deployed in the Canberra region's simulation and the real world. It achieved an average precision and input over union of 49.7 and 0.57 for the 12 simulated trials and 7.4 and 0.19 for the measured trials, respectively, for the environment type classification.

References

[1] P. L. Buttigieg, N. Morrison, B. Smith, C. Mungall, and S. Lewis, "The environment ontology: contextualising biological and biomedical entities," *Journal of biomedical semantics*, vol. 4, p. 43, 12 2013.

[2] T. Greider and L. Garkovich, "Landscapes: The social construction of nature and the environment," *Rural sociology*, vol. 59, no. 1, pp. 1–24, 1994.

[3] L. Jean-Baptiste, *Ontologies with Python*, 1st ed. Apress, 2021.

[4] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *Int. J. Robot. Res*, 2021. [Online]. Available: arXiv:2101.06894

[5] R. Reid and T. Bräunl, "Large-scale multi-robot mapping in magic 2010," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, 2011, pp. 239–244.

[6] Y. Yue, C. Zhao, R. Li, C. Yang, J. Zhang, M. Wen, Y. Wang, and D. Wang, "A hierarchical framework for collaborative probabilistic semantic mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9659–9665.

[7] S.-H. Joo, S. Manzoor, Y. G. Rocha, S.-H. Bae, K.-H. Lee, T.-Y. Kuc, and M. Kim, "Autonomous navigation framework for intelligent robots based on a semantic environment modeling," *Applied Sciences*, vol. 10, no. 9, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/9/3219

[8] A. J. Hepworth, D. P. Baxter, A. Hussein, K. J. Yaxley, E. Debie, and H. A. Abbass, "Human-swarm-teaming transparency and trust architecture," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 7, p. 1281–1295, Jul 2021.

[9] R. Neware and A. Khan, "Survey on classification techniques used in remote sensing for satellite images," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1860–1863.

[10] J. Wang, G. Zhang, M. Cao, and N. Jiang, "Semi-supervised classification of hyperspectral image based on spectral and extended morphological profiles," in *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2016, pp. 1–4.

[11] J. Zhang and L. Jia, "A comparison of pixel-based and object-based land cover classification methods in an arid/semi-arid environment of northwestern china," in *2014 Third International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*, 2014, pp. 403–407.

[12] C. Qiao, J. Luo, Z. Shen, Z. Zhu, and W. Wu, "Spatial knowledge based complicated urban area classification from high-resolution remote sensing image," in *Proceedings 2011 IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services*, 2011, pp. 405–408.

[13] D. S. Kanade, V. S. K. Vanama, and S. Shitole, "Urban area classification with quad-pol l-band alos-2 sar data: A case of chennai city, india," in *2020 IEEE India Geoscience and Remote Sensing Symposium (InGARSS)*, 2020, pp. 58–61.

[14] S. Li, Z. Li, H. Wang, J. Wang, and L. Li, "Urban land cover classification using aerial lidar and ccd images," in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 1967–1970.

[15] M. Djamel and K. Djerriri, "Extraction of urban areas from multi-date landsat imagery using presence and packground learning one-class classification framework," in *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*, 2020, pp. 144–147.

[16] W. Li, H. Liu, Y. Wang, Z. Li, Y. Jia, and G. Gui, "Deep learning-based classification methods for remote sensing images in urban built-up areas," *IEEE Access*, vol. 7, pp. 36 274–36 284, 2019.

[17] X.-M. Li and G. Wang, "Land-cover hierarchical classification method study of tm image in loess hilly ravine area," in *2012 International Conference on Computer Science and Service System*, 2012, pp. 1691–1694.

[18] S. Garg, T. Fischer, and M. Milford, "Where is your place, visual place recognition?" *CoRR*, vol. abs/2103.06443, 2021. [Online]. Available: https://arxiv.org/abs/2103.06443

[19] T. Barros, R. Pereira, L. Garrote, and C. P. andUrbano J. Nunes, "Place recognition survey: An update on deep learning approaches," *CoRR*, vol. abs/2106.10458, 2021. [Online]. Available: https://arxiv.org/abs/2106.10458

[20] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018. [Online]. Available: 10.1109/TPAMI.2017.2711011

[21] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Robotics: Science and Systems*, vol. 11, 2015.

[22] P. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Doorslam: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 1656–1663, 2020. [Online]. Available: https://doi.org/10.1109/LRA.2020.2967681

[23] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188–1197, 2009, inside Data Association. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889009000876

[24] S. Yu, C. Fu, A. K. Gostar, and M. Hu, "A review on map-merging methods for typical map types in multiple-ground-robot slam solutions," *Sensors (Basel, Switzerland)*, vol. 20, 2020.

[25] Y. Yue, M. Wen, C. Zhao, Y. Wang, and D. Wang, "Cosem: Collaborative semantic map matching framework for autonomous robots," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3843–3853, 2022.

[26] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. [Online]. Available: 10.1109/TRO.2017.2705103

[27] A. Ilze, ""heterogeneous map merging: State of the art," *Robot*, vol. 8, no. 3, p. 74–103, 2019. [Online]. Available: https://www.mdpi.com/2218-6581/8/3/74

[28] Y. Kong, W. Liu, and Z. Chen, "Robust convnet landmark-based visual place recognition by optimizing landmark matching," *IEEE Access*, vol. 7, pp. 30 754–30 767, 2019. [Online]. Available: 10.1109/ACCESS.2019.2901984

[29] S. Garg, N. Suenderhauf, and M. Milford, "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 0278364919839761, 2019. [Online]. Available: https://doi.org/10.1177/0278364919839761

[30] D. P. Baxter, A. J. Hepworth, K. F. Joiner, and H. Abbass, "On the premise of a swarm guidance ontology for human-swarm teaming," in *65th Annual Meeting of the Human Factors and Ergonomics Society*, Baltimore, Maryland, 2021.

[31] A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho, *Ontological engineering*, 1st ed. Springer, 2010.

[32] A. J. Hepworth, D. P. Baxter, and H. A. Abbass, "Onto4mat: A swarm shepherding ontology for generalised multi-agent teaming," 2022, doi: 10.48550/ARXIV.2203.12955. [Online]. Available: https://arxiv.org/abs/2203.12955

[33] M. Tenorth, A. Clifford Perzylo, R. Lafrenz, and M. Beetz, "The roboearth language: Representing and exchanging knowledge about actions, objects, and environments," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1284–1289. [Online]. Available: 10.1109/ICRA.2012.6224812

[34] J. Y. C. Chen, S. G. Lakhmani, K. Stowers, A. R. Selkowitz, J. L. Wright, and M. Barnes, "Situation awareness-based agent transparency and human-autonomy teaming effectiveness," *Theoretical Issues in Ergonomics Science*, vol. 19, no. 3, pp. 259–282, 2018. [Online]. Available: https://doi.org/10.1080/1463922X.2017.1315750

[35] A. Hussein, S. Elsawah, and H. A. Abbass, "Trust mediating reliability–reliance relationship in supervisory control of human–swarm interactions," *Human Factors*, vol. 62, no. 8, pp. 1237–1248, 2020, pMID: 31590574. [Online]. Available: https://doi.org/10.1177/0018720819879273

[36] H. Abbass and R. Hunjet, *SHEPHERDING UXVS FOR HUMAN-SWARM TEAMING*, 1st ed. SPRINGER NATURE, 2022.

[37] A. Hepworth, "Activity recognition for shepherding," in *Shepherding UxVs for Human-Swarm Teaming*, H. Abbass and R. Hunjet, Eds. Springer, Cham., 2021, ch. 7, pp. 131–164.

[38] L. Riazuelo, M. Tenorth, D. Di Marco, M. Salas, D. Gálvez-López, L. Mösenlechner, L. Kunze, M. Beetz, J. D. Tardós, L. Montano, and J. M. M. Montiel, "Roboearth semantic mapping: A cloud enabled knowledge-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 432–443, 2015. [Online]. Available: 10.1109/TASE.2014.2377791

[39] T. Ben, D. Feras, C. Peter, and W. Gordon, "Robot navigation in unseen spaces using an abstract map," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 4, pp. 791–805, 2021. [Online]. Available: 10.1109/TCDS.2020.2993855

[40] M. A. Cornejo-Lupa, Y. Cardinale, R. Ticona-Herrera, D. Barrios-Aranibar, M. Andrade, and J. Diaz-Amado, "Ontoslam: An ontology for representing location and simultaneous mapping information for autonomous robots," *Robotics*, vol. 10, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/4/125

[41] A. R. Washburn, *Search and detection*, 1st ed. Operations Research Department, Naval Postgraduate School, 1981.

[42] A. J. Hepworth, K. J. Yaxley, D. P. Baxter, K. F. Joiner, and H. Abbass, "Tracking footprints in a swarm: Information-theoretic and spatial centre of influence measures," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 2217–2224.

[43] S. Duncany, G. Estrada-Rodriguez, J. Stocek, M. Dragone, P. A. Vargas, and H. Gimperlein, "Efficient quantitative assessment of robot swarms: coverage and targeting levy strategies," *IEEE Transactions on Robotics*, 2022.

[44] Y. Chen, S. Huang, and R. C. Fitch, "Active slam for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Transactions on Mechatronics*, vol. 25, pp. 1182–1192, 2020.

[45] H. Abbass and R. Hunjet, *SHEPHERDING UXVS FOR HUMAN-SWARM TEAMING*, 1st ed. SPRINGER NATURE, 2022.

[46] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in *ECCV*, 03 2020, pp. 687–694.

[47] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.

[48] W. Wenxin, G. Liang, G. Hongli, Y. Zhichao, L. Yuekai, and C. Zhiqiang, "Yolo-slam: A semantic slam system towards dynamic environment with geometric constraint," *Springer*, 2022. [Online]. Available: 10.1007/s00521-021-06764-3

[49] S. Schwertfeger and A. Birk, "Evaluation of map quality by matching and scoring high-level, topological map structures," *2013 IEEE International Conference on Robotics and Automation*, pp. 2221–2226, 2013.

[50] S. Garg, N. Sünderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," *CoRR*, vol. abs/1804.05526, 2018. [Online]. Available: http://arxiv.org/abs/1804.05526

[51] M. Poveda-Villalón, A. Gómez-Pérez, and M. C. Suárez-Figueroa, "Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 10, no. 2, pp. 7–34, 2014.

[52] A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho, *Ontological engineering*, 1st ed. Springer, 2010.

[53] Z. Qian, J. Fu, and J. Xiao, "Towards accurate loop closure detection in semantic slam with 3d semantic covisibility graphs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2455–2462, 2022. [Online]. Available: 10.1109/LRA.2022.3145066