AUTONOMOUS MULTI-ROBOT TOPOLOGICAL SPATIAL COGNITION

by

Hakan Karaoğuz

B.S., Electrical and Electronic Engineering, Koç University, 2007

M.S., Systems and Control Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronic Engineering

Boğaziçi University

2015

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my supervisor Prof. Işıl Bozma for her guidance, enthusiasm and patience. Her guidance helped me to keep my motivation high at all times. Her suggestions and ideas helped me to find my way through at difficult times and successfully complete my thesis.

I would like to thank Prof. Yagmur Denizhan and Prof. Hakan Temeltaş for being a member of my thesis committee as well as giving valuable feedback through out my research. I would like to thank Prof. Lale Akarun and Prof. Can Özturan for attending to my thesis defence as thesis committee members.

I would also like to thank Özgür Erkent and Haluk Bayram for their valuable suggestions and friendship. They were always with me during hard times which helped me a lot while struggling with difficulties.

I would like to express my sincerest thanks and appreciation to all ISL lab members but especially Mahmut Demir, Halil Samet Çıldır and Kadir Türksoy for their countless support during software development and real-time experiments.

Finally, I would like to express my deepest gratitude to my wife and my family for their invaluable support. During my entire PhD study, they always stood beside me.

ABSTRACT

AUTONOMOUS MULTI-ROBOT TOPOLOGICAL SPATIAL COGNITION

This thesis is concerned with topological spatial cognition in multi-robot systems. With topological reasoning, continuous space is discretized into a set of places that are spatially related. Thus, topological spatial cognition is associated with the acquisition, organization, utilization and revision of knowledge of places and their spatial relations. In turn, this can occur either through the direct experience of the robot or indirectly based on the knowledge of other robots. In this perspective, the problem is handled in three stages. First, efficient sensory data representation is studied and a model based on previously developed bubble space is presented. Next, the full range of spatial processing associated with direct experience is considered via introducing a topological spatial cognition (TSC) model. This model enables each robot to detect places, recognize them or learn them as necessary in a completely unsupervised, incremental and organized manner. The robot continually builds and utilizes its long-term spatial memory where knowledge of places and their spatial relations are retained in separate, but related parts. Finally, the problem of expanding each robot's spatial cognition based on other robots' knowledge is addressed and a model that enables each robot to merge its spatial memory with those of other robots is proposed. All of the proposed approaches are evaluated on benchmark data sets as well as on real robots. The experimental results demonstrate that robots are able to autonomously become cognizant of their surrounding through either their individual experience or that of other robots.

ÖZET

ÇOKLU ROBOTLARDA OTONOM TOPOLOJiK UZAMSAL ANLAMDIRMA

Bu tezde, çoklu robotlarda topolojik uzamsal anlamlandırma konusu ele alınmıştır. Topolojik muhakeme sayesinde, robotun etrafındaki uzay uzamsal olarak birbiriyle ilişkili yerlere bölünür. Bu çerçevede, uzamsal farkındalık terimi, bu yerler ve ilişkiler ile ilgili bilgilerin toplanması, organizasyonu, anlamlandırılması ve gerekirse düzeltilmesi ile ilgilidir. Bu çerçevede, problem için geliştirilen yaklaşım 3 aşamada ele alınmıştır. Birinci aşamada, algılayıcı verilerinin daha önce geliştirilmiş olan baloncuk uzayı kullanılarak verimli bir şekilde tanımlanması konusunda çalışılmıştır. Daha sonra, robotun tamamen kendi bilgisini kullanarak uzamsal bilgi işleme konusu üzerine çalışılarak, Topolojik Uzamsal Anlamlandırma modeli geliştirilmiştir. Bu model, robotun gezdiği yol boyunca ziyaret ettiği farklı ortamların algılanması, öğrenilmesi ve tanınmasını sağlamaktadır. Bunun yanı sıra öğrenilen yerler arasındaki hiyerarşik ilişkileri gösteren ağaç ve yerler arasındaki uzamsal ilişkileri gösteren topolojik haritadan oluşan uzun dönemli uzamsal hafıza önerilen bu model ile oluşturulmaktadır. Son olarak, robotun kendi bilgisini diğer robotlardan gelen bilgilerle zenginleştirmesi konusu çalışılmıştır. Bu sayede robotun kendi uzamsal hafızasını diğer robotların hafızalarıyla birleştirmesi sağlanmıştır. Önerilen tüm yaklaşımlar, hazır verisetlerinin yanı sıra gerçek-zamanlı olarak robotlar üzerinde test edilmiştir. Deneyler sonucunda, robotların kendi verilerini veya başka robotlardan aldıkları verileri kullanarak etraflarındaki ortamları anlamlandırabildikleri görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$a(N)$	Attribute vector of node N
$a_l(N)$	$l-$th attribute of $a(N)$
B	Bubble space
$c \in R^2$	Robot's position
$c_m(N)$	Centroid of N^{th} node of place memory m
d_N	Discriminant function of node N
d_ϕ	Kernel size of the visual filter
D_m	m^{th} Detected place
E	Edges of topological map
f	Robot's viewing direction
$\mathcal{F}$	Robot's viewing direction state space
$g_N(D_m)$	Cost of associating D_m with node N
G^m	Topological map (graph) of robot m
$I(x) \in R^{N_I}$	Descriptor vector at base point x
$\bar{I}_m$	Mean descriptor of detected place m
$\mathcal{K} = \{1, \ldots, k^*\}$	Index set of base points
L_i	Sensory Feature Set
$\mathcal{M} = \{1, \ldots, m^*\}$	Index set of detected places
n_L	Depth of place memory
M_p	Total number of base points of place p
N	A node of place memory
$N^\uparrow$	Parent of node N
$N^\downarrow$	Children nodes of N
$N_I \in Z^+$	Descriptor dimension
N_l	Number of sensory features
p^*	The number of learned places
$p(N) \subset \mathcal{P}$	Places associated with node N
$\mathcal{P} = \{1, \ldots, p^*\}$	Index set of learned places

$\mathcal{Q}^*(k)$	Maximal neighborhood of x_k
$\mathcal{Q}^i(k)$	i-th neighborhood of x_k
$S(c_m(N), \rho_m(N))$	Hypersphere of N^{th} node of place memory m
T^m	Place memory of robot m
$T^i(k)$	i^{th}-temporal window of x_k
$T^*(k)$	Maximal temporal window of x_k
x_k	k'th base point
$\mathcal{X}$	Base space
$\alpha \in S^1$	Robot's heading
$\beta \in \mathcal{P}$	Place index
$\beta(D) \in \mathcal{P}$	Place index of D_m
κ	Incoherency function
$\rho_m(N)$	Radius of N^{th} node of place memory m
τ_h	(Max.) Place memory level merge threshold
τ_l	(Max.) Place memory traversal parameter
τ_p	(Min.) Plenitude threshold
τ_r	(Max.) Recognition cost threshold
τ_n	(Max.) Incoherency extension threshold
τ_w	(Max.) Temporal window extent parameter
τ_η, τ_σ	(Max.) Informativeness thresholds
τ_κ	(Max.) Incoherency threshold
ς	Informativeness function

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BOW	Bag of Words
BuS	Bubble Space
HOG	Histogram of Oriented Gradients
NARF	Normal Aligned Radial Feature
RGB-D	Vision and Depth Sensor
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TSC	Topological Spatial Cognition

1. INTRODUCTION

This thesis is concerned with topological spatial cognition in multirobot systems. If robots are ever to have lifelong operation and thus play a bigger role in daily life, it is crucial for them to have spatial cognition. With topological reasoning, continuous space is discretized into a set of places that are spatially related - motivated by findings in humans' spatial cognitive abilities [1, 2]. Thus, the robot is able to have efficient spatial characterization and reasoning. In this framework, a 'place' is defined to be a collection of appearances sharing common perceptual signatures or physical boundaries. As such, topological spatial cognition is concerned with the acquisition, organization, utilization and revision of knowledge of places and their spatial relations [3–5].

Figure 1.1. Outdoor Jaguar Robot Team.

1.1. Problem Statement & General Approach

Consider a robot team shown in Figure 1.1. Assume that each robot is capable of navigating around. The aim is to have each robot be spatially cognizant of the places it visits. The building and updating of topological spatial knowledge is integral in this. This thesis considers this problem in three stages as shown in Figure 1.2:

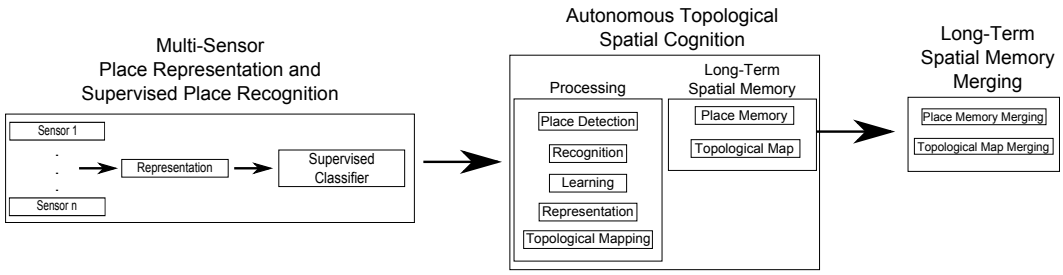


Figure 1.2. The three stages of the proposed approach.

(i) First, each place needs to be efficiently represented - taking all of the incoming sensory data into account. In particular, topological place representation based on sensory data from typical sensors and varying modalities is investigated. A model based on the previously developed bubble space is developed [6].

(ii) Second, the robot needs to have an accumulated knowledge base of places that it can refer to and update throughout its operation. The first means of acquiring this knowledge is through the direct experience of the robot. A topological spatial cognition (TSC) model is developed. An integral part of this model is the long-term spatial memory where knowledge of places and their spatial relations are retained in separate, but related parts. The processing part enables each robot to detect places, recognize them or learn them as necessary in a completely unsupervised, incremental and organized manner. Throughout this processing, it continually builds up and uses its long-term spatial memory simultaneously.

(iii) Alternatively, it can acquire this knowledge from other robots [7]. As such, topological spatial cognition problem from the perspective of a multirobot system is considered. In this case, the problem of expanding each robot's spatial cognition using other robots' knowledge is addressed. A model that enables each robot to merge its spatial memory with those of other robots is proposed.

1.1.1. Place Representation and Recognition

Place representation and recognition is a very active research area [6, 8]. The approach developed in this thesis is based on previous related work on bubble space and

attentive vision [9]. However, differing from that work, the robot is not assumed to look around attentively. Rather, it processes the whole scene. Furthermore, multiple sensory cues (laser,vision) are used instead of a single sensory cue. As such, the performance of different feature sets are analyzed to find optimal feature set that improves both computation and recognition performance.

1.1.2. Topological Spatial Cognition

An integrated model of autonomous topological spatial cognition is developed. This model consists of a continuum of representation, place detection, recognition, learning and mapping. The place memory organizes the learned places in a hierarchy based on their appearance-related similarities based on previously developed work while the topological map simply encodes their spatial relations. The processing modules operate together so that the robot builds its spatial memory or utilizes it in an organized, incremental and unsupervised manner. The developed approach is tested both on benchmark data sets and in real-time.

1.1.3. Merging of Spatial Memories

Once each robot builds up its own long-term spatial memory, it then considers expanding it via merging of spatial memories. A model that enables each robot to merge its spatial memory with those of its teammates is proposed. Once a robot receives the spatial knowledge of the other robot, it first merges its place memory with that of the other robot using a modified version of 'RACHET' algorithm for distributed hierarchical clustering [10]. It then incorporates other robot's map memory into its own via adding the new spatial relations as appropriately. This approach is implemented and tested on real robot data.

1.1.4. Contribution

The major contributions of this thesis are as follows:

(i) The existing work on place representation and detection is extended further to work without attentive vision and with multiple sensor modalities. Furthermore, an extensive evaluation of different feature sets are performed for finding optimal recognition performance [11].

(ii) A completely autonomous model of topological spatial cognition is developed. An integral aspect of this model is the concept of a 'place' that is is defined to be a collection of appearances sharing common perceptual signatures or physical boundaries.

- Places are detected using a novel place detection approach [12]. Its novelty is that ensuring sensory data reliability is integrated with place detection.
- A new approach to place recognition that explicitly considers long-term memory and the retrieval of knowledge from it is developed [13]. In this model, depending on whether has some knowledge of its whereabouts or not, its search in its long-term place memory proceeds differently.
- A model of long-term place learning that has been previously developed in [14] is completely integrated with representation, detection and recognition.
- A model of topological mapping using place memory is developed.

(iii) A novel approach for merging long-term spatial memories of multiple robots is developed. This approach allows each robot to merge its existing memory with the memories of any number of robots via considering them as wholes or in portions in a decentralized manner.

1.1.5. Organization of Thesis

The rest of this thesis is organized as follows.

The first problem is considered in Chapter 2. The previously proposed bubble space is adapted for place representation while processing the whole of the scene and with multi-modal sensory data. For the interested reader, a brief overview of bubble space is given in Appendix A. The performance of this model with varying feature sets is evaluated in a set of recognition experiments as to determine the optimal set of sensory features. Furthermore, recognition results with different combination of data

sets are given in Appendix B.

In Chapter 3, the topological spatial cognition (TSC) model is presented. Spatial knowledge that is comprised of two separate, but related parts - namely learned places and their spatial relations. The former is retained in the robot's place memory while the latter is encoded in the (topological) map memory. Places are detected while ensuring sensory data reliability via checking for informativeness, coherence and plenitude using only the bubble space representation of the incoming sensory data. They are recognized based on place memory with memory traversal varying depending on whether the robot knows where it is coming from or not. In case the robot does not recognize a place, it learns and maps it at the same time. The model is extensively evaluated using both benchmark data sets and real-robots. For the interested reader, the details of Jaguar Robot Team are explained in Appendix D.

In Chapter 4, a novel approach for merging long-term spatial memories coming from different robots. The problem is considered as the merging of appearance-based It is assumed that each robot has its individual spatial knowledge. In the merging process, each robot communicates with each of the other robots one-by-one and receives the spatial knowledge of the other robot . Once this is over, it first merges its place memory with that of the other robot using a modified version of 'RACHET' algorithm. It then incorporates other robot's map memory into its own via adding the new spatial relations as appropriately. The proposed approach is extensively tested on real-robot data.

The thesis concludes with a brief summary and a discussion regarding future work as presented in Chapter 5.

2. RGB-D BASED PLACE REPRESENTATION IN TOPOLOGICAL MAPS

2.1. Introduction

Place representation is an integral problem in topological map building [15]. Until recent years, research in this field has mostly focused on using visual or two-dimensional (2D) laser data. With the newly introduced integrated vision and depth (RGB-D) sensors such as Microsoft Kinect, it has now become much easier augment visual data with three-dimensional (3D) depth data.

In this chapter, we propose a new approach to RGB-D sensor based topological place representation - building on bubble space [9]. In bubble space, bubble surfaces encode all the features in a manner that is implicitly dependent on robot pose while preserving their local S^2-geometry. The associated feature vectors (bubble descriptors) are holistic representations of bubble surfaces in a rotationally invariant manner. Bubble descriptors have flexibility in integrating different features since its dimensionality is independent of the number of observations. Furthermore, no data association [16] is required for finding correspondences among observations taken at different times. The primary contribution of this chapter is of practical nature in regards to using bubble space representation for topological place recognition with RGB-D sensors. In this perspective, its contributions can be summarized as follows:

- First, while bubble space representation is in principal transparent to the type and number of sensory inputs employed, practically, it has been used only with visual data [6]. It has not been used with different sensing modalities including combined usage. In this chapter, it is shown that this framework can easily be used to integrate two disparate information – namely camera and depth features.
- Secondly, RGB-D sensors have some important drawbacks in comparison to the commonly used sensors such as the two degrees of freedom camera or the om-

nidirectional camera used in previous work [6]. In particular, their field of view ($\sim 60°$) is far more constrained than these sensors' field of views ($\sim 180°$) and their depth range is limited to typically less than 5 meters with noisy measurements [17]. This has a limiting effect on performance. In [18], it is shown that if multiple Kinect frames that cover 360° of each robot's field of view are used instead of a single frame, recognition performance increases nearly 30%. However, it may not be practical to turn the robot around itself for every visited base point or to deploy multiple number of Kinect cameras. In this work, we show that it is possible to obtain reasonable recognition rates even with limited field of view and resolution sensing.

- Finally, in many applications, the robot may be endowed with limited computational (memory and processing) resources. In order to be applicable in such scenarios, it should be possible to have acceptable recognition even with a very simple set of features. In this work, we consider this and show that the resulting performance is good - even with a limited set of features. Of course, as expected, a more comprehensive set will yield improved recognition rates. Hence, depending on the robot's computational capabilities, more extended and/or complex feature sets such as Harris-Laplace feature [19], SIFT descriptors [20], SURF descriptors [21], fingerprints [22] and biologically motivated filter responses [23] can easily be used instead.

In summary, we show that bubble space representation can easily be used to combine RGB and depth data while affording acceptable recognition performance even with limited sensing capabilities and simple features. The advantage of RGB-D sensing in bubble space is due to the fact that the associated feature vectors encode both sensory observations and their relative S^2 geometry. Hence, the rich 3D spatial information contained within the 3D depth data is not lost while allowing easy integration of data from different sensing modalities. For place learning and recognition, we use a standard supervised learning approach - support vector machines (SVM) in conjunction with bubble descriptors. During learning, as the robot collects RGB-D data at various viewpoints from each place, it simultaneously constructs the set of bubble surfaces and

then the associated bubble descriptors. These are then encoded in its memory using multi-class SVM. For recognition, it is given RGB-D data of one of these places – possibly with varying viewing conditions such as its viewpoint, illumination or a combination of both. Its decision making is based on comparing the currently constructed bubble descriptors with those of previously learned places in the SVM framework. Our experimental results with two datasets demonstrate that even with constrained field of view and resolution, relatively high recognition rates can be achieved with relatively low computational requirements.

The rest of this chapter is organized as follows. First, we give a review of the related literature in Section 2.2. In Section 2.3, we explain its usage with RGB-D data. Place learning and recognition based on bubble descriptors and multi-class SVM is explained in Section 2.4. Experimental results with benchmark RGB-D datasets are discussed in Section 2.5. The chapter concludes with a brief summary. A short overview of bubble space for completeness in Appendix A.

2.2. Related Literature

While geometric models for place representation enable explicit modeling of free-space [17], topological representations are believed to be "cognitively more adequate" [24] as they are more compact and amenable to communication. These representations can be categorized as purely vision based, depth based and vision-depth integrated approaches. In this section, we review related work with respect to each category.

2.2.1. Vision Based Representations

Vision based topological representations can be categorized into object-based and appearance based approaches. In object-based approaches, places are defined based on the occurrence statistics of objects in these places [25]. However, as object detection has remained to be challenging, these approaches have problems in generalizing to new environments [26]. The alternative appearance based approaches are based on global

configurations in the observed scenes [26].

Appearance based approaches are grouped into two categories. In context based appearance approaches, the incoming images are encoded directly such as using image patches [27], applying various transforms [28,29] or decomposing into eigen-images [30]. Another approach is to construct descriptors such as color histograms [15], histogram of oriented uniform patterns [26] or the colored pattern appearance model [31]. Alternatively, in landmark-based appearance approaches, a set of image features such as Harris-Laplace features [19], SIFT descriptors [20] and SURF descriptors [21] are first extracted. Robustness can be improved by using a mixture of local and/or global cues [32–34]. In some work, these features are used in generating the final representation [35,36] in order to minimize sensitivity to local variations in the incoming images. The most popular approach is bag-of-features representation [37]. Due to their high dimensionality [38, 39], approaches such as tree-structured Bayesian network and Chow Liu algorithm have been used for speeding the matching process [40, 41]. However, learning visual vocabularies remains to be challenging [40]. Moreover, only appearances of features are used and their relative spatial coordinates are discarded [42]. In [9], bubble space has been proposed as a novel representation for places in topological maps. In this work, we use bubble space for representing places based on RGB-D data.

2.2.2. Depth Based Representations

3D depth sensing based approaches for place recognition has remained very limited due to the complex nature of 3D sensing apparatus. From the mapping perspective, the three predominant data types for 3D depth data are Cartesian points, Cartesian point clouds and triangle meshes in either explicit or implicit forms [43] - which are all of metric type. Initial work has used the metric data directly [44], however the enormous number of points make the model reconstruction and the object recognition difficult. Compression schemes such as Gaussian regression are used to compute efficient surfaces [45] or feature-based sampling schemes [46] can be used to reduce the data considerably.

Alternatively, topological approaches attempt to alleviate this problem via converting the data into intermediate compact representations such as normal distribution transform used for appearance descriptors in [47], point feature histograms [48] , features based Laplacian of Gaussian [44], NARF features and bag of words (BOW) approach [49] and rotational invariant descriptors [50]. In all, while structural information is preserved locally around each cell of 3D depth data, the holistic relations among them are lost since these representations do not encode the relative geometry existing among the individual descriptors. These representations have been mostly used in 3D depth sensing based place recognition to detect loop closures for SLAM applications. Let it be noted that loop closure problems differ from place recognition problems as they rely on using additional information such as motion model and the sequentiality of incoming sensory data whereas such information is not used in place recognition problems.

2.2.3. Vision & Depth Based Representations

The integration of vision and depth data has been considered in order to combine their advantages while overcoming individual limitations [8]. Initially, this has been done with custom apparatus that combine cameras and 2D laser range finders. In these systems, feature vectors are constructed as to encode information from both sensing modalities [51–55]. With the recently introduced RGB-D sensors, the integration of the visual and depth data has become easier while allowing for richer representations of the environment. As this is a relatively new sensor, work is still in progress. One notable exception is associated with ImageClef 2012 Robot Vision Challenge that aims at place classification based purely on Microsoft Kinect data. Baseline results are presented via integrating HOG features obtained from visual data and NARF features obtained from depth data within OI-SVM cue integration framework [56]. A combination of SIFT features and color histograms based purely on RGB data is used by the third ranked approach [57]. The winner approach uses Fisher vector [58] obtained from SIFT features for both luminance and depth channels and linear classifiers [59]. The related, but slightly different problem of place categorization is considered in [60] where

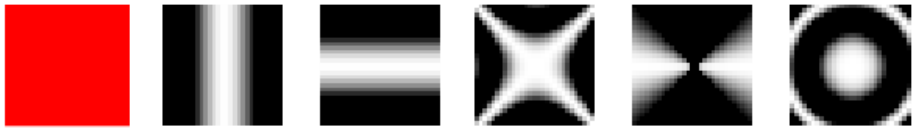


Figure 2.1. Visual filters. Left to right: Color, vertical, horizontal, hyperboloid, polar, cocentric filters. The first three are considered for the case of visual filters with $N_l = 3$. All are considered for the case of $N_l = 6$.

histograms of local binary patterns obtained from intensity and depth images are shown to achieve high categorization rates.

2.3. Features, Bubble Surfaces and Descriptors

The descriptor can be formed using one or several of the representations that have been developed. Here, we use bubble descriptors[1] [6]. However, the proposed model is in no way dependent on this particular choice and thus can be used with any other kinds of descriptors – as preferred.

The choice of sensory features $\mathcal{L}$ will vary depending on the task at hand. As discussed earlier, in many applications, the robot may be endowed with limited computational (memory and processing) resources. In order to be applicable in such scenarios, it is preferred to work with as simple set of features as possible. Of course, depending on the robot's computational capabilities, the feature set could be extended to have a greater number of features that are also more complex. In particular, we consider varying sets $\mathcal{L}_i$, $i = 1, \ldots, 5$ of sensory features as follows:

(i) $\mathcal{L}_1$ - Limited visual features only with $N_l = 3$: The sensory features $q_i(b, t)$, i=1,2,3 encode color (hue) and responses to vertical and horizontal filters as

[1]Bubble descriptors have been shown to be flexible and robust as they are able to incorporate any number of observations while preserving their relative S^2 geometry and being rotationally invariant. Furthermore, this representation can be used throughout all the remaining processing components as well as in the long-term spatial memory. For the interested reader, bubble space is explained briefly in Appendix A.

shown in Figure 2.1.

(ii) $\mathcal{L}_2$ - Extended visual features with $N_l = 6$: The sensory features $q_i(b,t)$, i=1,2,...,6 correspond to color (hue) and responses to Cartesian (vertical, horizontal) and Non-Cartesian filters (hyperboloid, polar, cocentric) filters as shown in Figure 2.1.

(iii) $\mathcal{L}_3$ - Depth only with $N_l = 1$: and $q_1(b,t)$ denotes the measured distance.

(iv) $\mathcal{L}_4$ - Integrated limited visual features and depth with $N_l = 4$: The sensory features $q_i(b,t)$, i=1,2,3, correspond to the three visual features while $q_4(b,t)$ is the depth value.

(v) $\mathcal{L}_5$ - Integrated extended visual features and depth with $N_l = 7$: The sensory features $q_i(b,t)$, i=1,2,...,6 correspond to the six visual features and $q_7(b,t)$ denotes depth value.

The color feature is computed based on hue value. The remaining visual features are obtained via applying a set of corresponding visual filters. The construction of visual filters is as explained in [23]. All the visual filters are placed into kernels of size $d_\phi \times d_\phi$ where d_ϕ is determined by considering the properties of foveal vision [61]. As foveal vision is associated with a narrow visual field, accordingly, we use a small kernel of size $d_\phi = 2.5°$. The corresponding filter size is determined via mapping the kernel size to the camera image plane based on its resolution and field of view. For example, with a 640×480 resolution and $54°$ horizontal field of view, the corresponding filters are of dimension 30×30.

For all features used, bubble surfaces are constructed with the number of harmonics $H_1 = H_2 = 9$. For a sample base point, the input visual and depth data are as shown in Figure 2.2(top-left) and Figure 2.2(top-center) respectively. As discussed earlier, Kinect sensor has a fairly narrow field of view in both the pan ($f_1 \in [-28.9°, 28.9°]$) and tilt ($f_2 \in [-21.6°, 21.6°]$) directions. Hence, in comparison to commonly used sensors such as the pan-tilt heads, omni-directional cameras or 2D wide angle depth scanners, much smaller portion of the bubble surface is deformed. This implies that each bubble surface encodes comparatively less amount of information.

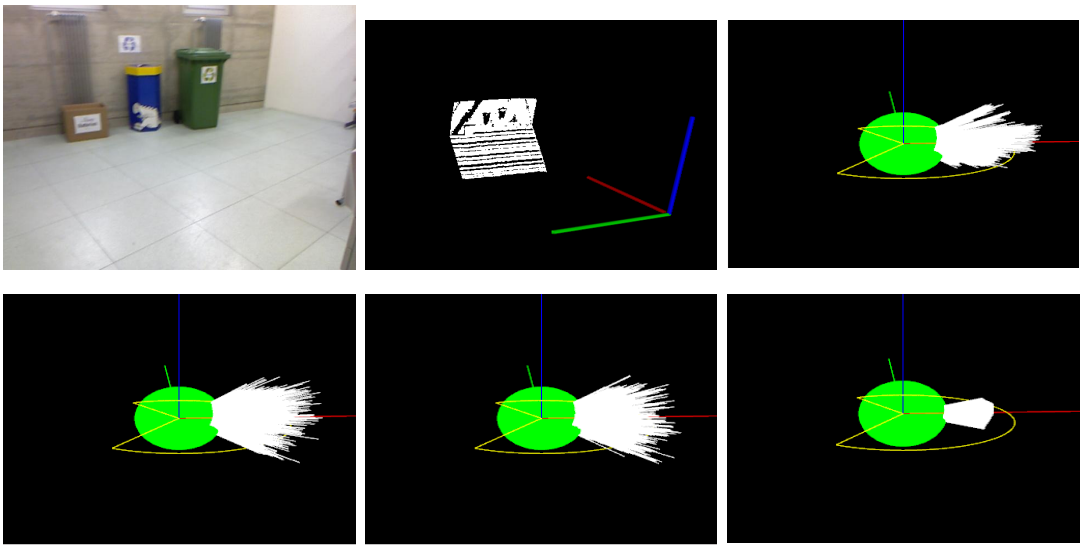


Figure 2.2. Sample sensory data at a base point. Top-left: Visual data; Top-center: Depth data; Bubble surfaces corresponding to different sensory features. Top-right: Visual feature 1 (color); Bottom-center: Visual feature 2 (Vertical filter response); Bottom-center: Visual feature 2 (Horizontal filter response); Bottom-right: Depth.

Bubble surfaces are then transformed into bubble descriptors. The dimension of each bubble descriptor varies depending on the size of feature set $\mathcal{L}_i$ and the number of harmonics associated with the bubble surfaces. When using only visual features $\mathcal{L}_1$ or $\mathcal{L}_2$, bubble descriptors $I(x,t)$ are of dimension $N_I = 300$ and $N_I = 600$ respectively. When using depth data only, $N_I = 100$. In case of integrated visual and depth data, bubble descriptors are of dimension $N_I = 400$ and $N_I = 700$ for $\mathcal{L}_4$ or $\mathcal{L}_5$ respectively.

2.4. Learning Places & Recognition

Let us suppose that the robot is to learn a set of places denoted by $\mathcal{P} = \{1, \ldots, p^*\}$. Suppose also that at each different place p, the robot goes to M_p different base points and generates a set of bubble descriptors $\mathcal{I}_p = \{I(x_j(p)) \mid j = 1, \ldots, M_p\}$. In learning mode, the bubble descriptors are organized in a supervised (manual) manner. In particular, we use one-against-one SVM for learning and recognizing places in which $N_K = \frac{p^*(p^*-1)}{2}$ binary SVM classifiers are trained to classify places.

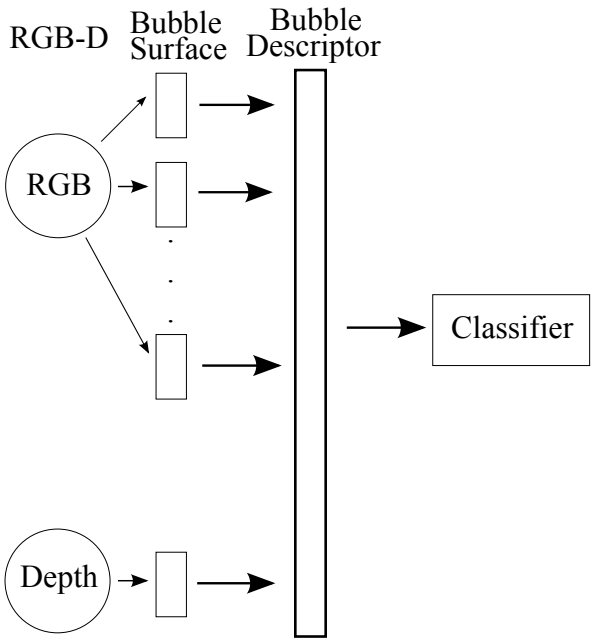


Figure 2.3. Recognition in bubble space.

In this framework, the problem of place recognition is defined as assigning a given representation (bubble descriptor in our case) to one of previously learned places $\beta \in \mathcal{P}$ or possibly designating it as a new (previously unknown) place based purely on this representation. Purely sensory data is used without any consideration of the robot's motion model, odometric data or the sequentiality of the incoming sensory data.

Suppose that the robot comes to a base point x_j, it starts to perceive the environment and constructs a bubble descriptor $I(x_j)$ as shown in Figure 2.3. Using the trained SVM classifiers, the probability $\xi_p(I(x_j))$ of being from each place $p \in \mathcal{P}$ is computed [62]. The decision rule is based on the computed probabilities:

$$\beta = \begin{cases} p' \in \arg\max_p \xi_p(I(x_j)) & p' \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

where τ is the confidence threshold that is selected by the user. If $\beta = 0$, this means that the current base point cannot be classified. In place recognition with only visual data [6], bubble space representation has been shown to have better recognition rates

compared to HIST [15] or classic Bag-of-Words (BoW) [37] while having comparable rates with extended-HCT (e-HCT) [63] and CPAM [31].

2.5. Experiments and Discussion

The proposed approach has been experimentally tested using two benchmark indoor RGB-D datasets. libSVM library is used for SVM implementation [64]. Optimal SVM parameters for the experiments are found using iterative 5-fold cross-validation. All the processing is done on a Linux based computer with 16 GB RAM and Intel Xeon 3.6 Ghz Quad Core Processor.

2.5.1. ImageClef 2012 Dataset

The first dataset is an indoor dataset recorded for ImageClef 2012 robot vision challenge. The dataset is recorded with a Microsoft Kinect sensor which is placed on a mobile robot that navigates through $p^* = 9$ different indoor places as shown from sample base points in Figure 2.4. The image frames are of size 640×480. The 3D depth data for each frame was encoded as a 2D image and converted to 3D point cloud by using supplied script [65]. There are 3 different learning sets. The first two of these sets have been obtained during daylight conditions with 2667 frames and 2532 frames respectively. Five frames from the first set are discarded as they do not have depth data. Their respective trajectories are the along the same route - although in reverse directions. The third set is collected in night conditions and has 1913 frames. While training conditions change, testing is always done with the same data that has been obtained at night conditions with 2441 frames. The optimal SVM parameters for this experiment are found as $\gamma = 2$, and $c = 8$ using iterative 5-fold cross validation. The problem is made more recondite by two data related issues: The discretized form of the depth data implies lower resolution and thus lower quality. This fact can be seen in Figure 2.2. In the camera image it can be seen that there are containers in front of the wall and the wall is not completely flat. However in 3D point cloud, these containers are barely detected and wall is seen as if it is completely flat. Secondly, the data collection is done with the Kinect sensor oriented towards the floor as shown in

Figure 2.4 which in turn tend to be flat and similar to each other and hence have lesser number of distinguishing features.

(a) Corridor (b) Elevator Area (c) Lounge Area

(d) Printer Room (e) Professor Office (f) Student Office

(g) Technical Room (h) Visio Conference (i) Toilet

Figure 2.4. $p^* = 9$ places in ImageClef dataset.

We first study the trade-off between recall and precision rates via varying the threshold parameter $\tau \in [0, 1]$. First, this is done via considering only limited visual features $\mathcal{L}_1$, only depth $\mathcal{L}_3$ and integrated feature set $\mathcal{L}_4$ for different combinations of training data. The results are as given in Figure 2.5. The same procedure is repeated for only extended visual features $\mathcal{L}_2$, only depth $\mathcal{L}_3$ and integrated features $\mathcal{L}_5$ - resulting in performance as shown in Figure 2.6.

With daylight training, using only limited visual features $\mathcal{L}_1$ has the worst performance with maximum achievable precision rate of 41% and the corresponding recall rate of 0.7%. The additional visual features clearly boost the success rate for vision-only sensing as the maximum achievable precision rate becomes 96% with 1.2% recall rate for $\mathcal{L}_2$. The results for depth only sensing $\mathcal{L}_3$ is low as expected due to low quality of the data as explained earlier. However, they are better compared to that obtained with using only visual features $\mathcal{L}_1$. This indicates that when there is a dramatic change

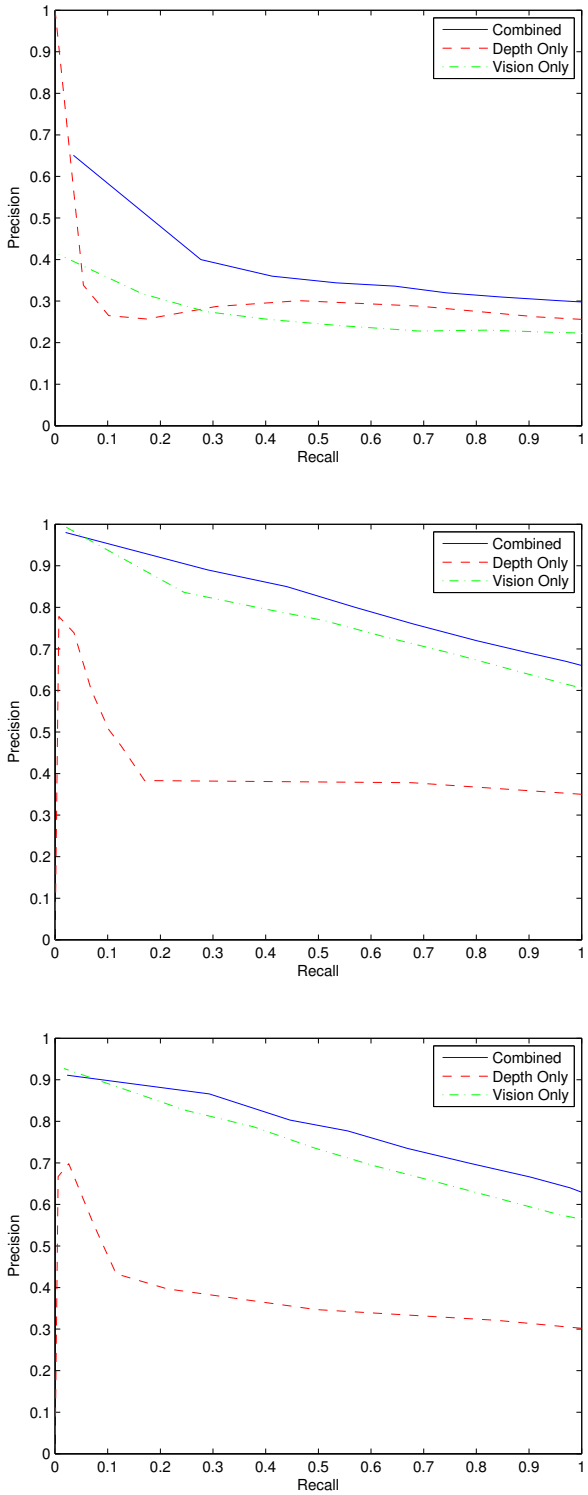


Figure 2.5. Precision-recall curves for ImageClef data with only limited visual features $\mathcal{L}_1$, only depth $\mathcal{L}_3$ and integrated set of features $\mathcal{L}_4$. Left: With daylight training; Middle: With night training; Right: With daylight-night training.

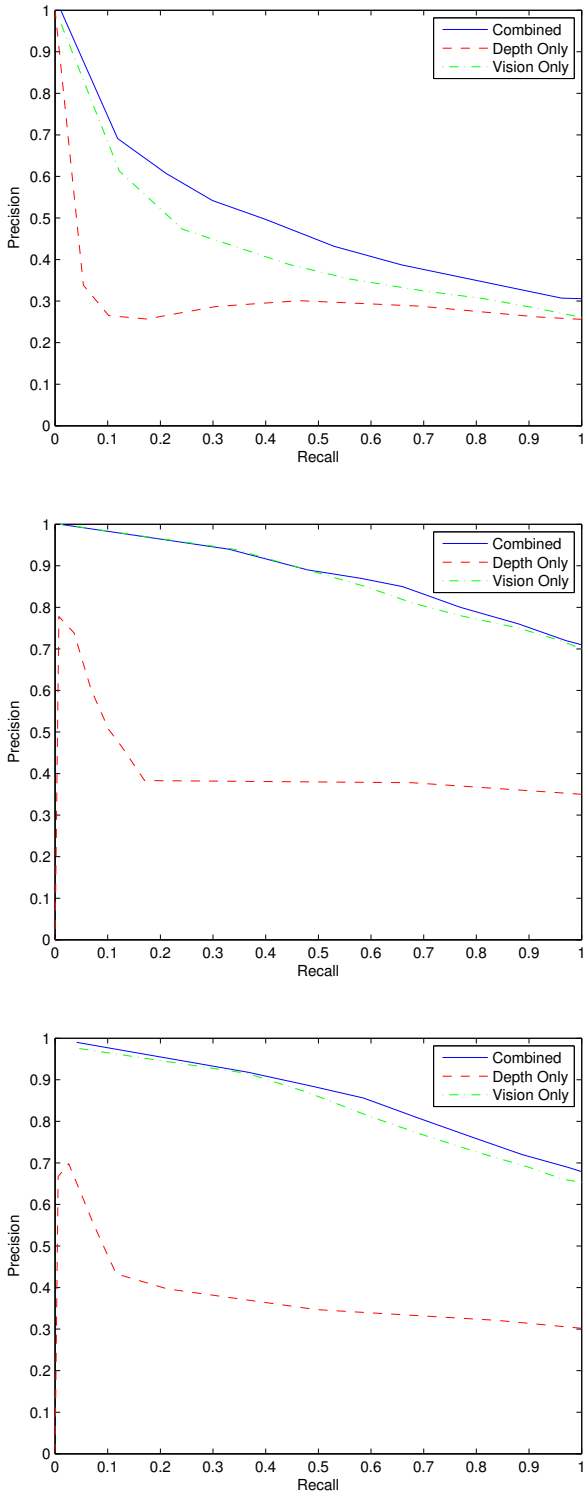


Figure 2.6. Precision-recall curves for ImageClef data with only extended visual features $\mathcal{L}_2$, only depth $\mathcal{L}_3$ and integrated extended set of features $\mathcal{L}_5$. Left: With daylight training; Middle: With night training; Right: With daylight-night training.

in lighting conditions, some visual features may become less reliable than depth data. The performance of $\mathcal{L}_4$ is also better compared $\mathcal{L}_1$ which both use the limited visual features. The best performance is attained with combined extended visual and depth data - namely $\mathcal{L}_5$. In this case, the maximum achievable precision rate is 100% with 1.1% recall rate.

If the robot is trained with night data, it is observed that 100% precision rate with 1.1% recall is achieved using visual features $\mathcal{L}_2$. If precision is reduced to 80%, the recall rate goes up to nearly 40% for visual features $\mathcal{L}_1$ and to 70% for extended visual features $\mathcal{L}_2$. Using only depth data $\mathcal{L}_3$ yields a lower success rate compared to only visual data case - as its maximum precision is 78% rate with a recall rate of 0.75%. The integrated vision and depth data is comparatively much better with a maximum precision rate of 98% with 2% recall for $\mathcal{L}_4$ and 100% with 1% recall for $\mathcal{L}_5$. At 80% precision, recall rate increases to 60% and 80% for each respectively.

For the daylight-night combined training, the results are better than the daylight only case but not as high as solely night training. With visual features only, a maximum precision of 92.7% with 1.7% recall is achieved for $\mathcal{L}_1$ while the corresponding performance for $\mathcal{L}_2$ is 98% precision with 3.6% recall. The depth-only $\mathcal{L}_3$ has much lower achievable precision rate of 67% with 0.7% recall rate. The integrated case achieve a maximum of 91.1% precision rate with 2.3% recall for $\mathcal{L}_4$ and 99% precision rate with 4.1% recall for $\mathcal{L}_5$. For this training set, the vision-only and integrated case achieved very similar results while depth-only case cannot get close.

As expected, the confidence threshold τ is critical to the performance of the system. When τ is high, the system behaves very conservative and leaves many samples as unclassified. While this leads to a significant increase in precision, recognition rate drops because of the high unclassification rate. If we decrease the threshold too much, then the system becomes too loose and as a result precision decreases. The best choice is to tune the parameter close to the middle range to make the system well-balanced such that the highest scores can be achieved most of the time.

Table 2.1. Comparative performance statistics (NA = Not available).

Feature Set		BuS		R&S
		$\mathcal{L}_4$	$\mathcal{L}_5$	
Max. ImageClef Score		874	1133	2071
Max. recognition Rate (%) (with precision=75%)		65	90	NA
Feature extraction time (ms/frame)		400	600	1320
Feature dimension		400	700	$> 80 \times 2048$
Learning time (min)	Daylight data	1.4	5	198
	Night data	0.25	1	
	Combined data	3	10	
Recognition time per frame (ms)	Daylight data	404.3	610	1340
	Night data	401.6	603.6	
	Combined data	406.5	613	

We also present a comparative study of bubble space representation (BuS) with the winner approach (R&S) [59] in the ImageClef 2012 competition as given in Table 2.1. The bubble descriptors are constructed using two alternative combined visual and depth feature sets – $\mathcal{L}_4$ and $\mathcal{L}_5$ respectively. The ImageClef scores are 874 and 1133 respectively for each case. Both are above baseline results given in [56]. While this is lower than the top score of 2071, the performance with feature set $\mathcal{L}_4$ ranks seventh while that with $\mathcal{L}_5$ ranks as fifth. However, for real-time applicability there are other aspects that need to be considered - particularly if the robots are endowed with limited computational (memory and/or processing) resources. It is observed that feature extraction step is significantly faster with less memory requirements. Furthermore, learning times are also much shorter with worst case of 10 minutes as opposed to 198 minutes. With the additional features used in $\mathcal{L}_5$, the average increase in the success rate is around 30% without much performance loss in feature extraction, learning and recognition times.

In summary, with RGB-D data, bubble space representation leads to acceptable performance in regards to recognition with comparatively less memory and processing requirements - which imply better applicability in cases of limited computational resources. As expected, combining visual features with depth clearly boosts performance compared to vision or depth only sensing. This makes sense since depth data complements visual data. While visual data is sensitive to illumination variations, depth data is not discriminating among places having geometrically similar structures. By integrating visual information with depth data, we can get more clues for recognizing each place. Of course, the quality of the learning data plays a key role in recognition performance. Our top scores are obtained with night training set which of course best suits the test set.

2.5.2. Experiment 2: Kyushu University Dataset

The second dataset is recorded in Kyushu University campus [60]. The data is captured using Microsoft Kinect sensor that is placed on a mobile platform. Based on the availability of data, we use 11 places consisting of 5 different corridors, 3 different offices and 3 different toilets. Figure 2.7 shows the office categories as an example. The data was recorded without temporal continuity which led us to measure the place-wise performance of our method instead of general classification performance. Optimal SVM parameters are found to be $\gamma = 0.5$ and $c = 8$ using iterative 5-fold cross validation. We again study vision only, depth only and integrated vision-depth sensing. The experiments are repeated 5 times for vision only, depth only and integrated vision-depth sensing. At each experiment, 45% of the available dataset of 817 is randomly selected for testing. The remaining 448 points are used for learning. $\mathcal{L}_1$, $\mathcal{L}_3$ and $\mathcal{L}_4$ feature sets are used for this dataset.

First, we present confusion matrices in Tables 2.3 and 2.4. Performance is very high with little confusion among different places for all sensor modalities. The small percentage of confusion is among different places of same type – such as different corridors.

The average of the obtained precision and recall rates are given in Table 2.5. With

Figure 2.7. Left to Right: Different offices in Kyushu University dataset.

Table 2.2. Confusion matrix for Kyushu University dataset: $\mathcal{L}_1$ feature set.

Place	Recognized As (%)										
	C1	C2	C3	C4	C5	O1	O2	O3	T1	T2	T3
C1	**99**	1	0	0	0	0	0	0	0	0	0
C2	1	**99**	0	0	0	0	0	0	0	0	0
C3	0	0	**96**	3	1	0	0	0	0	0	0
C4	0	0	0	**100**	0	0	0	0	0	0	0
C5	0	1	1	0	**97**	0	0	0	0	0	0
O1	0	0	0	0	5	**95**	0	0	0	0	0
O2	0	0	0	0	0	0	**100**	0	0	0	0
O3	0	0	0	0	0	1	0	**99**	0	0	0
T1	0	0	0	0	0	2	1	0	**94**	2	0
T2	0	0	1	0	0	0	0	0	3	**93**	4
T3	0	0	0	0	1	1	1	0	1	4	**92**

depth only $\mathcal{L}_3$ sensing, both the recall and precision rates are around 90%. These rates go up to 96% when integrated vision and depth sensing $\mathcal{L}_4$ is used. Likewise, vision only $\mathcal{L}_1$ sensing performs strongly which has nearly the same results as integrated sensing. Results using only single frame depth data are reported in [18] with an overall success rate around 60% for indoor place recognition. Although the dataset is not the same, our results are very promising in comparison.

Interestingly, these rates are much higher as compared to our results of first experiments due to 3 primary reasons. First, in contrast to the ImageClef dataset, the 3D point cloud data is available in its original format and thus has relatively high quality. Secondly, overlapping or transient scans between places is minimal or none since the

Table 2.3. Confusion matrix for Kyushu University dataset: $\mathcal{L}_3$ feature set.

Place	Recognized As (%)										
	C_1	C_2	C_3	C_4	C_5	O_1	O_2	O_3	T_1	T_2	T_3
C_1	**92**	5	0	3	0	0	0	0	0	0	0
C_2	7	**93**	0	0	0	0	0	0	0	0	0
C_3	0	0	**100**	0	0	0	0	0	0	0	0
C_4	0	0	0	**99**	0	0	0	0	1	0	0
C_5	0	0	1	5	**88**	0	0	0	5	0	1
O_1	0	0	0	0	0	**84**	1	9	6	0	0
O_2	0	0	0	0	0	2	**93**	0	0	0	5
O_3	0	0	0	0	0	7	1	**92**	0	0	0
T_1	0	0	0	0	0	4	0	0	**87**	4	5
T_2	0	0	0	0	0	2	3	0	9	**75**	11
T_3	0	0	0	0	0	2	1	0	6	10	**81**

Table 2.4. Confusion matrix for Kyushu University dataset: Integrated vision-depth sensing $\mathcal{L}_4$.

Place	Recognized As (%)										
	C_1	C_2	C_3	C_4	C_5	O_1	O_2	O_3	T_1	T_2	T_3
C_1	**99**	1	0	0	0	0	0	0	0	0	0
C_2	2	**98**	0	0	0	0	0	0	0	0	0
C_3	0	0	**100**	0	0	0	0	0	0	0	0
C_4	0	0	0	**100**	0	0	0	0	0	0	0
C_5	0	0	0	3	**96**	0	0	0	1	0	0
O_1	0	0	0	0	0	**98**	1	1	0	0	0
O_2	0	0	0	0	0	0	**100**	0	0	0	0
O_3	0	0	0	0	0	6	0	**94**	0	0	0
T_1	0	0	0	0	0	0	0	0	**94**	2	4
T_2	0	0	0	0	0	0	0	1	2	**89**	8
T_3	0	0	0	0	0	0	0	0	0	7	**93**

data was recorded without temporal continuity. Finally, the test and training samples are exactly in the same lighting conditions which simplifies the classification task. These reasons clearly indicate that the ImageClef dataset is more challenging compared to this dataset.

Table 2.5. The average recall and precision rates for Kyushu University dataset with visual, depth and integrated data (Note: σ represents standard deviation).

	$\mathcal{L}_1$		$\mathcal{L}_3$		$\mathcal{L}_4$	
Place	Recall %	Precision %	Recall %	Precision %	Recall %	Precision %
C1	99	99	96	92	99	99
C2	96	99	91	93	99	98
C3	98	96	99	100	100	100
C4	99	100	97	99	99	100
C5	94	97	99	88	100	96
O1	95	95	82	84	94	98
O2	96	100	93	93	99	100
O3	100	99	90	92	96	94
T1	97	94	83	87	97	94
T2	91	93	79	75	90	89
T3	97	92	83	81	91	93
Overall	**97,** $\sigma = 3$	**97** $\sigma = 2$	**89** $\sigma = 5$	**90** $\sigma = 5$	**96** $\sigma = 3$	**97** $\sigma = 3$

2.5.3. Summary

Our experimental results on two different RGB-D data sets suggest that bubble space representation enables acceptable high recognition rates with acceptable precision under similar learning and testing illumination conditions. In ImageClef dataset, recognition rates are around 90% with 75% precision while with Kyushu University data set they are much higher with 97% recall rate at 97% precision. We attribute this difference in performance to two data related issues in the ImageClef data set - namely its compressed form and the orientation of Kinect sensor during data collection. It is also shown that, the performance can be improved via extending this set. The advantage of RGB-D based BuS representation is to offer extreme flexibility and simplicity in integrating different sensory features and observations while affording acceptable

performance even with limited sensing and simple features. Furthermore, results with ImageClef data sets demonstrate comparatively much lower memory and processing requirements - which suggest real-time applicability with robots having limited computational resources.

2.6. Conclusion

In this chapter, a new approach to RGB-D based topological place representation - building on previously proposed bubble space is proposed. In this framework, RGB and depth data are easily integrated via bubble descriptors. Bubble descriptors are feature vectors that simultaneously encode the associated visual and depth features as well as their relative S^2-geometry. Any number of feature observations associated with the two modalities can be added without changing its dimensions. Furthermore, with bubble descriptors, finding correspondences among observations from a single sensor taken at different times is no longer required. After a supervised learning stage, the robot can use the bubble descriptors computed based on current RGB-D data in order to recognize the place it is - among the predefined set of places. Our experimental results on two datasets along with comprehensive analysis demonstrate that even with very simple features and limited field of view, acceptable recognition rates can be achieved with very low computational (memory and processing)requirements. These results suggest real-time applicability of RGB-D based bubble space representation.

3. TOPOLOGICAL SPATIAL COGNITION

3.1. Introduction

With topological spatial approaches, the continuous world is viewed as a discrete set of places and their spatial relations. A 'place' is defined as a collection of appearances sharing common physical or perceptual boundaries [12, 66, 67]. The key motivation is that visual data from a single location will not encode all the place related knowledge. This definition differs appearance-based or topological SLAM methods where each location is considered separately as a place [35, 68, 69] or a representative location (key-place) is selected after grouping visual data from different locations [70]. In the former, key-frames do not necessarily represent distinct 'places' since their selections are typically arbitrary while in the latter, key-places may not encode all the place related knowledge since they are defined by the midpoint frames of the associated clusters. Furthermore, the resulting spatial knowledge is thought to be more consistent with human-like definitions that are associated with higher-level symbolic reasoning and semantic analysis [25]. However, topological spatial cognition has proven to be a challenging task as it encompasses a multitude of processing components - each with a complex functionality itself - that need to be working together in an integrated manner .

This chapter is focused on endowing a robot with topological spatial cognition. The contribution of the chapter is to present an integrated model as shown in Figure 3.1. In this model, a place inherently spans visual data from multiple locations sharing common perceptual boundaries. As the robot navigates around, distinct places are detected via monitoring coherent parts of the incoming visual data stream while pruning out uninformative or insufficient data. Detected places are then either recognized or learned as necessary and their spatial relations are mapped. The novelties of the model are two-fold:

- First, it explicitly incorporates a long-term spatial memory that stores two sep-

arate, but related types of knowledge: places and spatial relations. A place memory organizes all the learned places based on appearance in a hierarchical tree structure while the topological map (memory of spatial relations) encodes their spatial relations.

- Second, the processing modules operate together so that the robot is able to build its spatial memory in an organized, incremental and unsupervised manner.

Let us note that while there has been extensive work on spatial cognition, to the best of authors' knowledge, none of the existing models consider long-term spatial memory where each place is viewed as a collection of locations and where this memory evolves in an organized, incremental and unsupervised manner. As such, the robot is able to build an environmental representation that is amenable for higher-level symbolic reasoning and semantic analysis. Thus, the proposed model constitutes a step forward towards having robots that are capable of understanding their surroundings and hence interacting intelligently throughout (possibly life-long) operation.

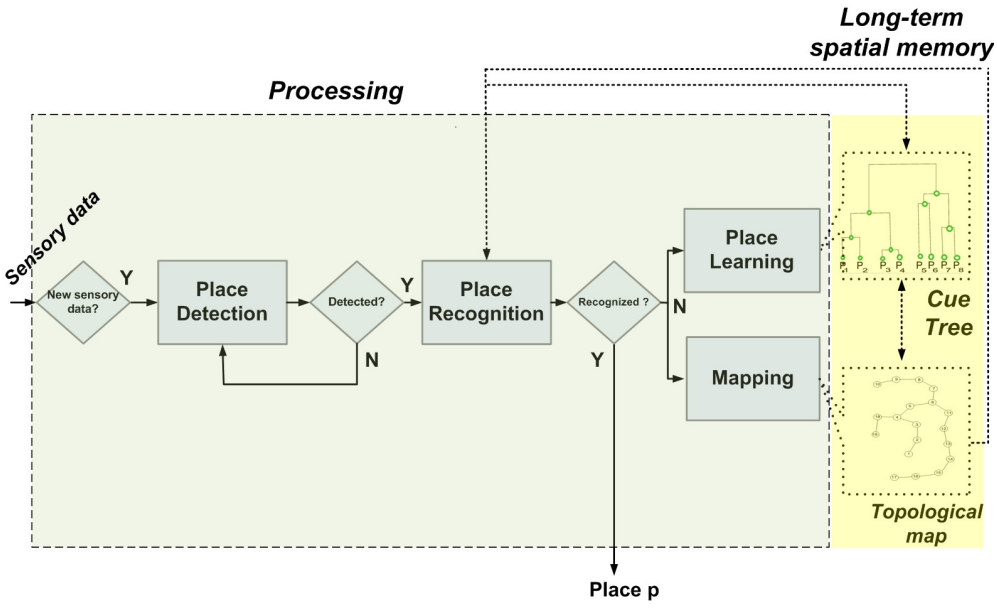


Figure 3.1. Topological spatial cognition model.

The outline of the chapter is follows: First, we review related work in Section 3.2. The long-term spatial memory is formulated in Section 3.3. Place detection is explained

in Section 3.4. This is followed by the remaining three components - namely place recognition in Section 3.5, learning in Section 3.6 and mapping in Section 3.7. The model is evaluated with an extensive sets of experiments in both indoors and outdoors settings along with discussion of comparative results with related work in Section 3.8. The chapter ends with a brief summary including future directions.

3.2. Related Literature

Spatial cognition has been receiving significant attention in robotics. It is observed that most work have focused on the processing aspect of spatial cognition. The proposed systems can be categorized depending on whether they use loop closure or recognition for recall. Loop closure methods are usually associated with SLAM approaches that use metric data and maps. In these approaches, each location is viewed as a place on its own and as such they do not provide a basis for semantic analysis and higher level, symbolic reasoning [71–73]. Furthermore, both memory and processing requirements become extremely high in large environments. On the other hand, recognition methods are used with topological reasoning as is the case in the proposed work. Topological reasoning is in general more efficient, however it needs to be accompanied by learning and recognition.

There are three aspects to learning: knowledge organization, incrementality and supervision. We humans are believed to organize the learned knowledge in a hierarchical manner using spatial or semantic attributes [74, 75]. Accordingly, knowledge representation is integral to the robot's spatial cognitive abilities [76]. The efficiency of knowledge storage and retrieval directly affects performance. While the proposed approaches all maintain some representation of spatial knowledge, these representations are usually specific and implicitly defined. For example, with loop-closure methods, if all the visited locations are recorded and need to be matched against, scalability may be difficult. Thus, the matching mechanisms rely on additional knowledge such as the robot's previous whereabouts in order to alleviate perceptual aliasing [40]. Similarly, recognition efficiency may be affected if the current place model needs to be compared with all the learned place models one-by-one [25, 67, 70, 77]. Again, the process is im-

proved by having the robot know its current whereabouts on the topological map [71] - possibly using additional odometric data [78]. Spatial Semantic Hierarchy is one of earlier models that addresses knowledge representation via a multitude of interacting qualitative and quantitative representations [79]. This model has been generalized to a spatial knowledge abstraction framework where local metric maps are related to higher level topological representation [80]. The different levels of abstraction have been expanded to allow spatial categorization [81]. A model consisting of sensory, place, categorical and conceptual layers enables the full abstraction of knowledge while considering uncertainties and knowledge fusion from multiple sources [76]. The second aspect pertains to how the data is presented. Batch methods assume all the learning data to be available [9, 52, 81–84]. Learning needs to be repeated from scratch if the robot needs to learn new places. Alternatively, with incremental methods, the knowledge base is gradually built and thus new places are naturally accommodated [40, 67, 78, 85]. For example, a probabilistic place model with incremental update is proposed for learning places and detecting loop closures [40]. An SVM classifier pre-trained with a number of visual categories is incrementally updated with new training examples [85]. Finally, supervised approaches rely on semantic information such as place labels being externally provided [9, 52, 77, 81, 83, 84, 86] In contrast, unsupervised approaches do not rely on external guidance and aim to learn from unlabeled data [66, 67, 70, 78, 87–90]. Fewer work consider incremental and unsupervised learning simultaneously. For example, GMMs that combine visual and odometric data are used to build place models incrementally without any supervision [78]. In [67], places are detected and recognized in an unsupervised and incremental manner as the robot explores the environment. Dirichlet Process Mixture Models are used to learn place models in an incremental and unsupervised fashion in [90]. An approach that integrates dynamic vocabulary building, incremental topic modeling and topic space clustering is proposed to detect and recognize places in an incremental and unsupervised manner [70]. Interestingly, none of these work consider explicitly knowledge organization together with learning.

3.3. Long-Term Spatial Memory

The long-term spatial memory is a record of past experience and thus plays a key role in the effective storage and retrieval of knowledge. There are two types of learned knowledge - places and their spatial relations. Since they are of different nature, each is stored in a different part of the memory.

A place memory organizes the set of learned places $\mathcal{P}$ in an hierarchy based on appearance-related similarities [14]. The place memory hierarchy is defined by a nested sequence of partitions of $\mathcal{P}$ as inferred from the visual data and is viewed as associating a set of appearance-related attributes. As there are no externally provided labels expressed in natural language such as "kitchen" or "Saar building", it is not possible to expect such explicit label assignments [91]. with the learned places. The first level corresponds to the first attribute and the last level corresponds to the n_L-th attribute. Each node N is associated with a set of places $\mathcal{P}(N) \subseteq \mathcal{P}$ that are viewed as sharing a set of semantic attributes $a(N)$. As parent attributes are propagated to the children, $a(N)$ is iteratively defined $a(N) = \begin{bmatrix} a(N^{\uparrow})^T & a_l(N) \end{bmatrix}^T$ where $a(N^{\uparrow})$ denotes attribute vector associated with its parent $N^{\uparrow}$ and $a_l(N)$ is the distinguishing attribute of node N. As the level increases, so does its specificity. As such, each terminal node is associated with one maximal set of attributes and is viewed as corresponding to a distinct place. Thus, each node N except the root node with a discriminant function d_N which measures the likelihood of a detected place D_m that is associated with $a_l(N)$ as its $l-$th attribute given that its first $(l-1)$ attributes are given by the parent node $a(N^{\uparrow})$. Each discriminant function is constructed based on one-class SVM [92]. As such, the place memory enables efficient retrieval and update of learned knowledge. Furthermore, the hierarchical propagation of attributes provides a basis for semantic analysis and understanding.

In parallel, topological map stores any observed spatial relations among different places - similar to [66, 93]. It is defined by an evolving undirected graph $G = \{\mathcal{P}, E\}$ with nodes $\mathcal{P}$ and edges E. The nodes correspond to the different places while edges represent the adjacency relations between the different places. As places are already

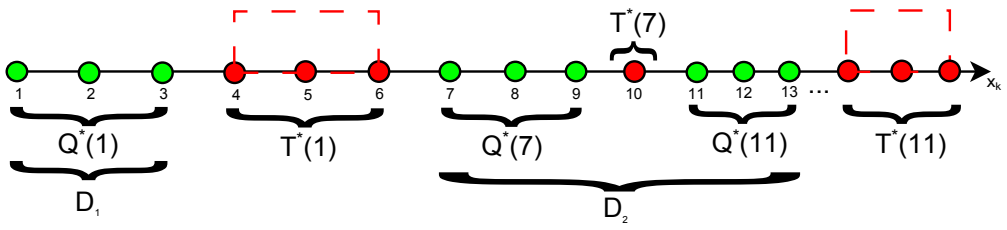


Figure 3.2. Place detection - Partitioning the base points into places. Each collection D_m of base points corresponds to one distinct detected place.

stored in the place memory, the map simply consists of edges as defined by place tuples $ij \in E$. Namely $ij \in E$ if and only if the robot has navigated from the place i to the place j for $i, j \in \mathcal{P}$.

3.4. Place Detection

Consider a robot at position $c \in R^2$ with a heading $\alpha \in S^1$. Let $x = \begin{bmatrix} c^T & \alpha \end{bmatrix}^T$ be referred to as the 'base point'. The set $\mathcal{X} \subseteq R^2 \times S^1$ is the base space (all possible locations and headings). If odometric information is not available or is unreliable, the coordinates of the base point x will not be known explicitly - as is assumed here. As the robot goes through a sequence of base points x_k with $k \in \mathcal{K}$, it forms a sequence of descriptors $I(x_k) \in R^{N_I}$.

Places are detected via the iterative partitioning of the set $\mathcal{K}$ so that each cell corresponds to one distinct place. The partitioning is based on the coherence of the associated visual data while pruning out data is not informative or of sufficient plenitude [12]. The partition evolves as the robot visits new base points. Let the partition be denoted by $\{D_0, ..., D_{m^*}\}$ with cells $D_m \subset \mathcal{K}$ indexed by $\mathcal{M} = \{1, \ldots, m^*\}$. As the partition evolves, the index set $\mathcal{M}$ expands. The first cell D_0 contains all discarded base points due to low informativeness, low coherency or low plenitude of the data from them. Each other D_m corresponds to one distinct 'detected place'. While the distinctness of each place can be enunciated in a set of spatial and/or semantic attributes that are often expressed in natural language, these attributes are unknown to the robot –

since it has no external guidance or natural language capability.

The partitioning process is based on identifying maximal neighborhoods and temporal windows. Each maximal window encodes a cluster of base points that are associated with one detected place while each temporal window corresponds to a transition region that is not considered as a part of any place. Each base point is added to the maximal window or temporal window considering the informativeness, coherency and plenitude of the associated descriptors $I(x_k)$, $k \in \mathcal{K}$ [12]. The descriptor $I(x_k)$ may not be informative due to problematic environmental conditions such as low illumination or viewpoint (robot looking at a large object or being very close to one). We assume that this can be measured by a binary valued function $\varsigma : \mathcal{X} \rightarrow \{0, 1\}$ that depends on a priori set informativeness thresholds $\tau_\mu > 0$ and $\tau_\sigma > 0$. Coherency is related to the consistency of sensory data from two from two or more consecutive base points. It is assumed to be measured by a binary valued function $\kappa : \mathcal{X} \rightarrow \{0, 1\}$ that depends on a priori determined incoherency threshold τ_κ. The final check is for plenitude. The extent of each detected place as compared to plenitude threshold τ_p is an indication of the amount of information. The specific formulations of these functions will depend on the descriptors used. With our descriptors, they are as given in Appendix C.

Suppose the robot is at a particular base point x_k. Its maximal neighborhood $\mathcal{Q}^*(k) \subset \mathcal{K}$ is determined by clustering consecutive base points that are both informative and coherent. First, let $\mathcal{Q}^1$ denote 1-neighborhood of base point x_k if the succeeding base point x_{k+1} is informative and coherent:

$$\mathcal{Q}^1(k) \stackrel{\triangle}{=} \{k + 1 \mid \varsigma(x_{k+1}) = 0 \text{ and } \kappa(x_{k+1}) = 0\} \tag{3.1}$$

Now, iteratively define the $(i + 1)^{th}$ base neighborhood as:

$$\mathcal{Q}^{i+1}(k) \;:=\; \left(\bigcup_{j \in \mathcal{Q}^i(k)} \mathcal{Q}^1(j) \right)$$

According to this definition, each $(i + 1)^{st}$ neighbor of base x_k is informative and

coherent wrt to some i^{th} neighbor of base k. Once the extent of $\mathcal{Q}^{i+1}(k)$ exceeds τ_p, the robot has sufficient sensory data for detecting a place while the recognition module is activated periodically as explained in Section 3.5. Hence the robot can recognize its whereabouts - if it is revisiting a previously learned place. Meanwhile, the iterative process continues (is in progress) until uninformativeness or incoherency is detected - which implies that $\mathcal{Q}^{i+1}(k) = \mathcal{Q}^{i}(k)$. Let i^* be the corresponding index. Finally the maximal neighborhood is defined as:

$$\mathcal{Q}^*(k) \;\; := \;\; \mathcal{Q}^{i^*}(k)$$

A temporal window $T^*(k) \subset \mathcal{K}$ clusters consecutive base points that are uninformative or incoherent. Once such a base point is found, a temporal window is initiated as $T^1(k)$:

$$T^1(k) \stackrel{\triangle}{=} \{\zeta_1(k)\}$$

where $\zeta_1(k)$ denotes the index of uninformative or incoherent base point closest to x_k.

$$\zeta_1(k) \stackrel{\triangle}{=} \inf(k' > k \mid \varsigma(x_{k'}) = 1 \text{ or } \kappa(x_{k'}) = 1) \tag{3.2}$$

As the robot navigates to new base points, repeated consecutive detection of uninformativeness or incoherencies extends the temporal window:

$$T^{i+1}(k) := T^i(k) \bigcup_{j \in T^i(k)} \{k' \leq \zeta_2(j)\} \tag{3.3}$$

where $\zeta_2(j)$ to be the smallest index that is at most τ_n distant to x_j while still being either not informative or incoherent:

$$\zeta_2(j) \stackrel{\triangle}{=} \inf(k' > j \mid \varsigma(x_{k'}) = 1 \text{ or } \kappa(x_{k'}) = 1, j < k' \leq j + \tau_n) \tag{3.4}$$

The incoherency extension threshold τ_n defines the number of succeeding base points that will be checked. If there is at least one uninformative or incoherent base point in the next τ_n base points, then the temporal window is extended to include the index of the corresponding base point. Thus, each $T^i(k)$ th temporal window contains uninformative or incoherent base points to some i^{th} neighbor of base k. This process is repeated as long as uninformativeness or incoherency is detected in the next τ_n base points. When it is stopped, then the extent of the temporal window is finalized. Let i^* be the corresponding index.

$$T^*(k) := T^{i^*}(k)$$

Once a temporal window terminates, the extent of $T^*(k)$ as compared to temporal window extent parameter τ_w is used to decide how to use this knowledge. A short extent indicates sensing problems which suggests that the associated data simply needs to be ignored. On the other hand, a long extent signals transition regions which suggests that the regions before and after the transition region need to be detected as two different places. In the current system, the parameters τ_w and τ_n are set based on the manual inspection of the sample data as to obtain optimal detection of transitions.

With these two definitions, the partitioning process is defined as an iterative process with an initialization as follows:

$$r_1 := k_1 \text{ where } k_1 \in \arg\min_{k \in \mathcal{K}} \{k \mid \varsigma(x_k) = 0 \text{ and } \kappa(x_k) = 0\}$$
$$D_0 := \{k' \mid k' < r_1\}\,; \qquad D_1 := \mathcal{Q}^*(r_1)$$
$$m = 1 \qquad T^*(r_1) = \emptyset$$

where $r_1 \in \mathcal{K}$ is the start index of D_0. The iterative steps are expressed as:

$$
\begin{aligned}
r_m &:= \max(D_0, \max\left(\bigcup_{j \leq m-1} D_j\right)) + 1 \\
&\text{if } |T^*(r_m)| < \tau_w, \text{ then } D_{m-1} := D_{m-1} \bigcup \mathcal{Q}^*(r_m) \\
&\text{else if } |T^*(r_m)| \geq \tau_w \\
&\quad D_m := \mathcal{Q}^*(r_m) \\
&\quad \text{if } |D_m| \leq \tau_p, \text{ then } D_0 := D_0 \bigcup D_m \\
&\quad \text{else } m = m + 1 \\
&D_0 := D_0 \bigcup T^*(r_m)
\end{aligned}
$$

Here, $r_m \in \mathcal{K}$ is the start of each maximal neighborhood. The process continues as long as the robot navigates to new base points x_k. A sample place detection scenario is given in Figure 3.2 using parameters $\tau_n = 2$, $\tau_w = 2$ and $\tau_p = 2$. There are 3 neighborhoods $\mathcal{Q}^*(k)$ detected - namely $\mathcal{Q}^*(1)$, $\mathcal{Q}^*(7)$ and $\mathcal{Q}^*(11)$. In this case, all contain 3 base points. The temporal windows $T^*(1)$, $T^*(7)$ and $T^*(11)$ are shown in red dashed rectangles. The first and third contain 3 base points while $T^*(7)$ contains only 1 base point. As such, correspond to transition regions that separate two distinct places while $T^*(7)$ is interpreted as noisy data from a single place. At the end, 2 places - D_1 and D_2 are detected. The first detected place is $D_1 = \mathcal{Q}^*(1) = \{1, 2, 3\}$ while the second place is $D_2 = \mathcal{Q}^*(7) \bigcup \mathcal{Q}^*(11)$.

Finally, each detected place D_m is represented distinctly by the corresponding set of descriptors $I(x_j)$, $j \in D_m$. The mean descriptor $\bar{I}_m$ is defined as:

$$
\bar{I}_m = \frac{1}{|D_m|} \sum_{j \in D_m} I(x_j) \tag{3.5}
$$

3.5. Place Recognition

In recognition, the robot attempts to map the detected place D_m to a place index $\beta(D_m) \in \mathcal{P}$ via relating to its long-term spatial memory. Recall that each detected place D_m is represented by a set of descriptors $I(x_j)$ with $j \in D_m$ and the mean descriptor $\bar{I}_m$. Place memory plays a key role in this process. The robot traverses through the place memory in order to find a terminal node that relates to the detected place. It will be able to find such a node only if the detected place has been previously visited and learned. Otherwise, there will be no recognition. There are three aspects to recognition – namely the decision-making at each non-terminal node N of the place memory, how to traverse the place memory and the updating of place knowledge in case of recognition.

The decision-making at each node N is based on minimizing a cost function g_N while ensuring that the minimum cost is below the recognition cost threshold τ_r. The cost function measures how unlikely the detected place D_m is to be associated with one of children nodes $N^{\downarrow}$. First, let $\tilde{\gamma}(D_m, N)$ measure the dissimilarity of the detected place to the places associated with the node based on the respective descriptors:

$$\tilde{\gamma}(D_m, N) = \left\| \bar{I}_m - \frac{1}{|\mathcal{P}(N)|} \sum_{p \in \mathcal{P}(N)} \bar{I}_p \right\|^2 \tag{3.6}$$

The greater this value is, the more dissimilar the detected place to the places associated with node N. If $N^{\downarrow}$ denotes the children nodes of N respectively, let $N^1(D_m), N^2(D_m) \in N^{\downarrow}$ to be the two offspring nodes that are most similar to D_m as:

$$N^1(D_m) \in \arg\min_{N' \in N^{\downarrow}} \tilde{\gamma}(D_m, N') \tag{3.7a}$$

$$N^2(D_m) \in \arg\min_{N' \in N^{\downarrow} - N^1(D_m)} \tilde{\gamma}(D_m, N') \tag{3.7b}$$

Finally, the function g_N is defined as:

$$g_N(D_m) = \tilde{\gamma}(D_m, N^1(D_m)) + \frac{\tilde{\gamma}(D_m, N^1(D_m))}{\tilde{\gamma}(D_m, N^2(D_m))} + (1 - V_{N^1}(D_m)) \tag{3.8}$$

The first term encodes the lowest dissimilarity between the detected place D_m and children nodes of N. The second term indicates the reliability of this. If D_m is equally similar to the two nodes $N^1(D_m)$ and $N^2(D_m)$, then this term increases. Otherwise, it decreases. The third term measure the vote percentage that does not associate D_m with the node. The vote percentage $V_N(D_m)$ is computed after considering all the base points D_m and checking whether each is likely to be in any of the places associated with this node as:

$$V_N(D_m) = \frac{1}{|D_m|} \sum_{j \in D_m} v_N(j) \tag{3.9}$$

where $v_N(j) \in \{0, 1\}$ is the vote for each base x_j - depending on the discriminant function value $d_N(x_j)$ as:

$$v_N(j) = \begin{cases} 1 & \text{if } d_N(I(x_j)) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The cost value is checked against a recognition cost threshold τ_r.

$$g_N(D_m) \leq \tau_r \tag{3.11}$$

If τ_r is too small, then the number of places may be too large - thus making the place memory structure unnecessarily complex. Conversely, if τ_r is too large, then place knowledge may be incorrectly updated. In case this condition is satisfied, depending on whether $N^1(D_m)$ is a terminal node or not, either place assignment can be made or the robot moves to another node in the place memory and the process is repeated. Otherwise, the place is declared to be unrecognized. When a place is recognized, the associated terminal node $N^1(D_m)$ is updated including member base points, the mean bubble descriptor $\bar{I}_{N^1}$ and the corresponding discriminant function.

The robot uses two alternative strategies for place memory traversal: top-down depth-first or hybrid (integrated bottom-up and top-down). It switches between the two depending on whether it has any knowledge of where it is coming from - namely the last node it was at in the topological map. If the robot has just started operating or has been kidnapped, then it will have no prior knowledge about the environment. In this case, all it can do is to employ a top-down, depth-first strategy that starts from root node and propagates down the place memory until a decision can be reached. At each node N, $g_N(D_m)$ is computed. In case $g_N(D_m) \leq \tau_r$, the closest child node $N^1(D_m)$ is selected and the process is repeated. If the closest child is a terminal node, D_m is said to be recognized as $\beta(D_m)$ where $\beta(D_m) = p$ where $p \in \mathcal{P}(N^1(D_m))$. In case $g_N(D_m) > \tau_r$, the traversal comes to an halt and the detected place is deemed to be not recognized. If the robot has been operating for a while and has been relating to the topological map, it can utilize this knowledge and traverse the place memory in a hybrid manner. Let D_{m-1} be the previously detected place that is associated with a learned place $\beta(D_{m-1}) \in \mathcal{P}$. In this case, a good node N to start search in the place memory is the parent of the node associated with place $\beta(D_{m-1})$. The robot can then consider other offspring nodes and see if it can recognize the detected place D_m. Recognition is based on computing the cost $g_N(D_m)$ and checking it against a recognition cost threshold τ_r. Depending on whether $g_N(D_m) \leq \tau_r$ or not, two alternative decisions are made. In case this condition is satisfied, D_m is said to be recognized if the most similar child N^1 of N is a terminal node which implies that D_m is recognized as $p \in \mathcal{P}(N^1(D_m))$. Otherwise, the process needs to be repeated at another node. The next node is selected via going up one level and considering the parent $N^{\uparrow}$ of N. This process is repeated until either it goes up by an a priori defined search level τ_l or D_m is recognized. In case of the former, the robot starts traversing the place memory in a top-down manner starting from the last visited node. Hybrid strategy is particularly advantageous if the structure of the place memory is such that spatially nearby places are organized closer together in the hierarchy. The place memory traversal parameter τ_l affects how many levels will be searched in a bottom-up manner. If set to a low number, then only a small part of the place memory will be potentially searched and the robot may not be able to recognize a learned place. Otherwise, a larger part of the place memory will be

potentially searched. While this will possibly increase recognition performance, it will also increase the computational requirements as it may be possibly searching in vain - even if the place is really unknown.

3.6. Place Learning

Learning aims to store the knowledge associated with visited places for subsequent referral. It occurs when the robot does not recall the detected place D_m. Suppose that the robot has learned p^* places – namely $\mathcal{P} = \{1, \ldots, p^*\}$. Initially, the set of learned places $\mathcal{P} = \emptyset$ with $p^* = 0$. Through learning, the set $\mathcal{P}$ grows with each new place as:

$$\begin{aligned} p^* &= p^* + 1 \\ \mathcal{P} &= \mathcal{P} \cup \{p^*\} \end{aligned}$$

The place label of the detected place is set as $\beta(D_m) = p^*$.

Places are learned via updating the place memory tree via the appropriate insertion of a terminal node associated with the new place $p^* + 1$. The update is based on hierarchical single link clustering method SLINK [94] as presented in [14]. Edges between two different nodes N and N' are constructed based on the similarity of the associated descriptors as measured by $\gamma(N, N')$:

$$\gamma(N, N') = \left\| \frac{1}{|\mathcal{P}(N)|} \sum_{p \in \mathcal{P}(N)} \bar{I}_p - \frac{1}{|\mathcal{P}(N')|} \sum_{p \in \mathcal{P}(N')} \bar{I}_p \right\|^2 \tag{3.12}$$

In the SLINK algorithm, the evolving place memory tree is described based on two iterative functions $\upsilon_p : \mathcal{P} \to [0, \infty]$ and $\psi_p : \mathcal{P} \to \mathcal{P}$. The value $\upsilon_p(i)$ is the lowest distance at which place i is no longer the last member in its cluster while $\psi_p(i)$ is the index of the last place that joins this cluster at the lowest distance $\upsilon_p(i)$. When a new place $p^* + 1$ is inserted into the place memory, the functions υ_{p^*+1} and ψ_{p^*+1} are

computed incrementally:

$$\begin{aligned} \upsilon_{p^*+1}(i) &= \begin{cases} \infty & i = p^*+1 \\ \min\{\mu_p(i), \upsilon_p(i)\} & i < p^*+1 \end{cases} \\ \psi_{p^*+1}(i) &= \begin{cases} p^*+1 & i = p^*+1 \\ p^*+1 & \mu_p(i) \leq \upsilon_p(i) \\ p^*+1 & \mu_p(\psi_p(i)) \leq \upsilon_p(i) \\ \psi_p(i) & \text{otherwise} \end{cases} \end{aligned} \tag{3.13}$$

Here, μ_p is defined as:

$$\mu_p(i) = \min\left\{\gamma(N_i, N_{p^*+1}), \min_{\psi_p(j)=i} \max\{\mu_p(j), \upsilon_p(j)\}\right\} \tag{3.14}$$

where N_i and N_{p^*+1} are the terminal nodes associated with places i and p^*+1 respectively. Thus, $\mu_p(i)$ is defined for $i = 1, \ldots, p^*$ and since $\mu_p(i) \leq \gamma(N_i, N_{p^*+1})$, it is finite for all i. Note that for number of places p^*, the computation of distance function considers $\frac{1}{2}p^*(p^*-1)$ pairs of clusters with an required storage of order $O(p^{*2})$. SLINK algorithm reduces this dependence to order $O(p^*)$ - in fact $3p^*$ [94]. The operation complexity is shown to be of order $O(log(p^*))$. Hence, the theoretical order-of-magnitude bounds for compactness of storage and operation complexity both make the approach feasible even if the number of places p^* approaches well in the range of $10^3 - 10^4$.

After the update, the robot re-organizes its memory via simplifying the place memory. The efficiency of storage and processing of the place memory can be improved if the resulting tree structure consists of fewer nodes and levels. If the similarity of a node N to its parent $N^{\uparrow}$ is high, this may indicate that there is a level redundancy. In this case, it would be preferable to combine the parent and the children on the same level. The simplification procedure starts from the bottom and is repeated until root node. The place memory level threshold τ_h controls the extent of merging. When the value of τ_h is high, the tree structure will be simpler with fewer levels. However,

terminal nodes that are associated with distinct places may be wrongly associated with the same parent node which can affect the recognition performance adversely. Conversely, if τ_h is low, then the place memory structure will be more complex as there will be many split levels and corresponding internal nodes. As a result, the computational cost of learning and recognition will increase. After the place memory structure is modified, SVM models are updated at the nodes associated with structural changes.

3.7. Mapping

Mapping aims to store the learned spatial relations that exist among different places. Initially, when the set of learned places $\mathcal{P} = \emptyset$ with $p^* = 0$, the topological map is also empty with $G = \emptyset$. As the robot explores the environment, while $\mathcal{P}$ and the place memory expands, so does the topological map. This is achieved via adding edges between the consecutively learned places to the map as:

$$E = \bigcup_{m=2}^{m^*} \{\beta(D_{m-1})\beta(D_m)\}$$

While updating the map, it uses the map and its place memory to determine whether current detected place D_m was previously visited or not. If the detected place has been previously visited and learned as $\beta(D_m) \in \mathcal{P}$, this indicates that a corresponding node already exists in the topological map. The robot checks whether an edge between the previous place $\beta(D_{m-1}) \in \mathcal{P}$ and current place $\beta(D_m)$ exists or not and updates the topological map accordingly. In case there is no edge (which implies that the robot is learning the transition from place $\beta(D_{m-1})$ to place $\beta(D_m)$, the topological map is updated as:

$$E = E \bigcup \{\beta(D_{m-1})\beta(D_m)\}$$

Note that each edge corresponds to a transition region $T^*(r_m)$ as detected in place detection. On the other hand, if it is not recognized, a new node corresponding to the

new place p^* is added to the topological map along with a new edge that links the node associated with the previously visited place and the newly added node:

$$E = E \bigcup \{\beta(D_{m-1})p^*\}$$

3.8. Experimental Results

We evaluate the proposed model via studying how a mobile robot's spatial cognitive abilities evolve in two different sets of experiments. In the first set of experiments, we use the combined benchmark indoors COLD dataset [95] and outdoors New College dataset [96] - in order to have our evaluation as extensive as possible. In the second set, we conduct experiments with the data of our Jaguar robot operating outdoors. In both, the robot has only visual data from a sequence of base points along a given path. It is not given any other sensory data. At each base point, the robot encodes the incoming visual data by a descriptor having dimension $N_I = 600$. The details of this descriptor are provided in Appendix A for the interested reader. Initially, long-term spatial memory is empty – namely both the place memory and the topological map are initialized to be empty. Recall that the model has nine predefined parameters - six being associated with place detection, one with recognition and the remaining two with place memory learning. All the parameter values are determined manually prior to experimentation based on the robot's camera type and remain fixed once it starts operating. In particular, the detection parameters are selected as to provide reliable detection of different space units in the environment. The hardware setup and the environmental effects such as illumination are taken into consideration while selecting these parameters. The recognition parameters are selected as to have high precision - thus some detected places may not be recognized even if they have been previously visited learned. Once in operation, all the processing is autonomous with no external intervention.

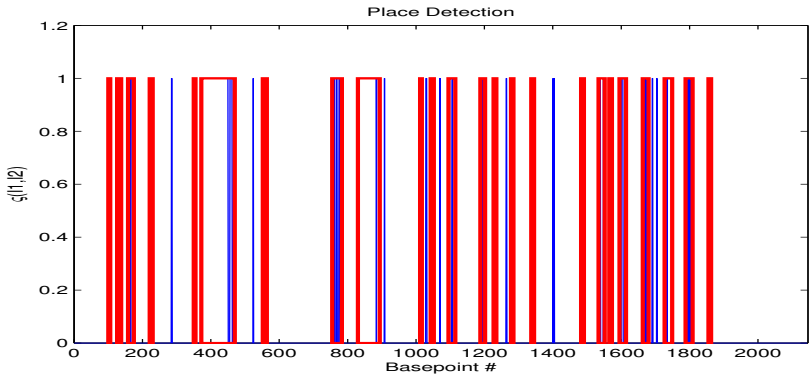
Table 3.1. Parameter settings for COLD+NC dataset.

τ_μ	τ_σ	τ_κ	τ_n	τ_w	τ_p	τ_h	τ_r	τ_l
0.2	0.01	0.4	10	5	20	0.25	1.1	1

3.8.1. Combined COLD and New College Datasets

The COLD dataset consists of visual data from three different sites - Freiburg Lab (Fr), Ljubljana Lab (Lj) and Saarbrucken Lab (Sa) - with sample scenes as shown while New College (NC) dataset is from a college campus with scenes as also shown in Figure A.1a. Both are recorded with perspective cameras. Using perspective camera adds more challenge in terms of recognition since the camera has a limited field of view. Thus, a scene may possibly look very different depending on the viewing angle and hence it may not be possible to identify two such views as being from the same place. The parameter settings are as given in Table 3.1. Their values have been selected manually as to visually optimize the performance of the associated module. After the robot's spatial cognition is activated, it starts processing visual data along its route in each site consecutively under cloudy illumination.

3.8.1.1. First-Time Visits. The robot's first-time tour in the Fr site consists of 2146 base points from a route of 100 meters. The place detection module processes these base points as shown in Figure 3.3a where bases associated with uninformative/incoherent data or short extents are indicated by blue, transition regions (incoherent regions with long extents) are indicated in red and detected places are indicated in white. As there is no metric available, we evaluate the detection performance qualitatively based on evaluating whether the resulting places correspond to accurate depictions of places. The robot finds the data from 480 base points to be either uninformative as seen in Figure 3.3b or incoherent (mostly due to abrupt heading changes) as seen in Figure 3.3c. The remaining 1666 base points are partitioned into 25 places. As 4 of these are pruned as their extents are less than the plenitude threshold τ_p, there are 21 detected places as shown in Figure 3.4a. Visual inspection of the site plan reveals that there



(a) Place detection results. The x-axis corresponds to consecutive bases. Blue base points indicate uninformative or incoherent regions having short extents, red base points correspond to transition regions and white base points designate detected places.

(b) Sample uninformative data.

(c) Sample sequence of incoherent data.

Figure 3.3. Place detection in Fr site after first-time visit.

are 8 space units in this site. The robot associates each space unit with about 2-3 places with the exception of the corridor area which is associated with 6 places. In the Sa site, the robot goes through a sequence of 997 base points along a 50 meters route. Data from 23 base points are labeled as uninformative while data from 128 base points are labeled as incoherent. The remaining 846 base points are partitioned

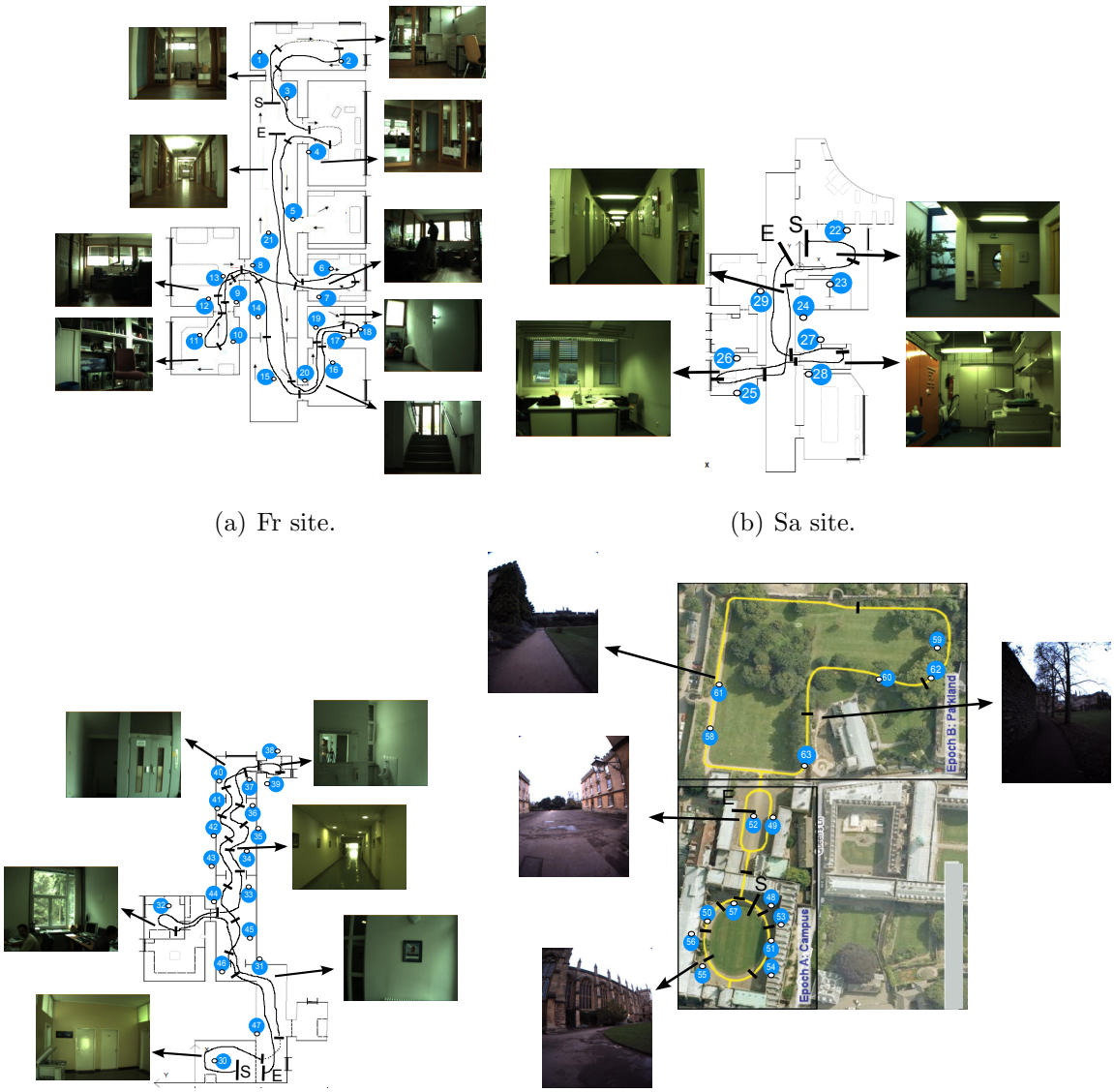
into 9 places – however, one place is ignored due to the plenitude threshold. Thus, 8 places are detected in this site as shown in Figure 3.4b. Again, a visual inspection of the site plan indicates 4 space units with 1-2 places per space unit. In the Lj site, the robot goes through 2112 base points from a route of 180 meters. Data from 72 base points are labeled as uninformative while data from 617 base points are labeled as incoherent. The remaining 1423 base points are partitioned into 25 places – only 18 pass the plenitude condition as shown in Figure 3.4c. This site has many detected places considering there are only 4 associated space units. This is due to the zig-zag motion of the robot during its navigation along the corridor. Finally, in the NC site, the robot goes through 5800 base points as it loops a route of 600 meters multiple times. The robot determines 44 base points to be uninformative while 778 base points are labeled as incoherent. With the remaining 4978 base points, 16 places are detected[2] as shown in Figure 3.4d which are all valid. These results are as summarized in Table 3.2.

Table 3.2. Number of base points and detected places for COLD+NC dataset.

Site	Base points			Detected Places		
	Uninformative	Incoherent	Total	Valid	Pruned	Total
Fr	37	443	2146	21	4	25
Sa	23	128	997	8	1	9
Lj	72	617	2112	18	7	25
NC	44	778	5800	16	0	16

Once the robot detects a place, it attempts to recognize it based on its long-term spatial memory. As the place memory is initially empty, it cannot do so and starts to learn the detected places one by one. When the Fr site tour is complete, it is observed that there are 20 places learned as seen in Figure 3.5a. This is because detected places D_{10} and D_{12} are recognized as a single place (place 10) - which is determined to be correct via visual inspection. After visiting Sa site, the number of learned places increases to 28 as seen in Figure 3.5b. Note that in this site, each detected place is learned as a

[2]Note that in a related work [70], there are 28 key-places detected with the same dataset.



(a) Fr site.

(b) Sa site.

(c) Lj site.

(d) NC site.

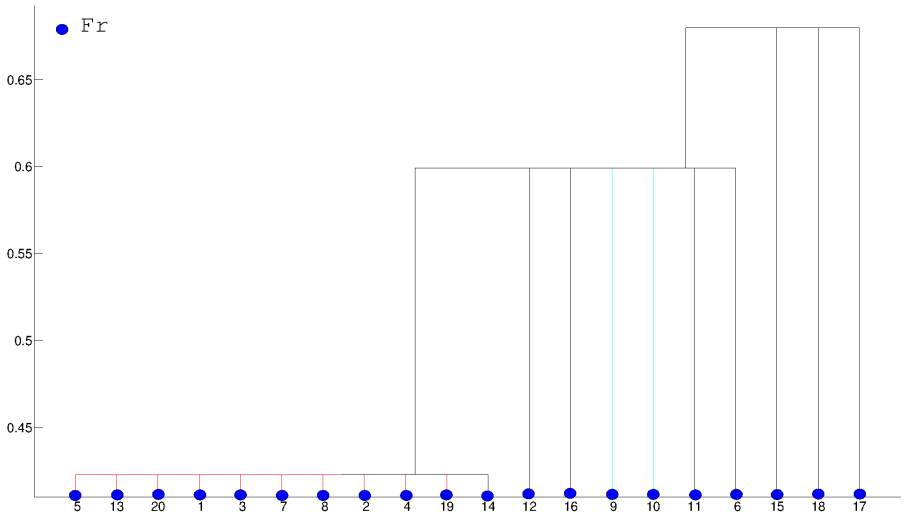
Figure 3.4. Detected places along the robot's path where the sequence of base points corresponding to each detected place is indicated by the cutting short black lines and robot's heading is as shown by the white dot. 'S' and 'E' represent the start and end points of the tours respectively.

distinct place as expected. Next, in Lj site, the place memory expands to have 45 terminal nodes - as detected places D_{31} and D_{46} are recognized as a single place (place 30) as seen in Figure 3.6a. Interestingly, the recognition of these two detected places is a nice surprise since they are geometrically close but have opposite viewing angles. This is attributed to the fact that symmetric places such as corridors and doorways even if

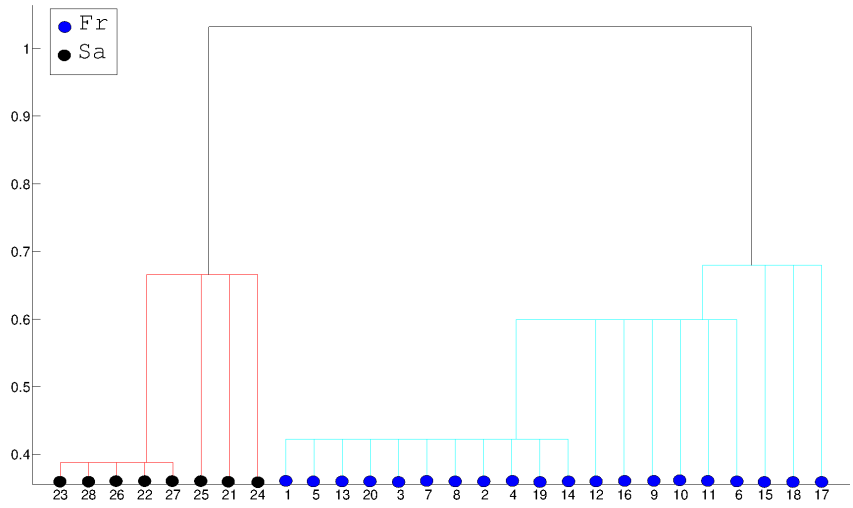
acquired with perspective cameras, can nevertheless look similar. After the robot visits the NC site, the place memory as seen in Figure 3.6b has 59 terminal nodes. Some of the detected places (D_{49}, D_{52} and D_{58}, D_{61}) are recognized as single places. Thus, the associated terminal nodes are updated accordingly. Normally, more node updates are expected for NC site since the robot revisits the same places during its tour. The lack of recognition and node updates are probably due to the changing illumination conditions during robot's tour and the low overlap between detected places due to the changes in robot's viewing direction at consecutive revisits.

The resulting place memory after first-time visits is as seen in Figure 3.6b and has 4 levels. Recall that the attributes are defined by the hierarchical structure. The top level has two children nodes. The left node represents the Sa and Lj sites while the right node represents Fr and NC sites. This suggests that the places in the Sa and Lj sites share a common attribute value as well as those of Fr and NC sites. This finding is reasonable as the appearances of the places in the Fr and NC sites have a more colorful texture and natural illumination as seen in Figure A.1 while those in the Sa and Lj have more similar structure and artificial illumination. At the second level, for the right side the outlier places of Fr (places $6, 15-17$) and NC (places 54, 56, 57, 58) sites are grouped together while at the left side outlier places of Sa (place 21) and Lj (places 36, 43) sites are grouped. At the third level, on the left hand site, six places from the Sa site are grouped together (places 22, $23, 25-28$) as well as a smaller group is formed with a place from Sa site (place 24) and two places from Lj site (places 31 and 45). This indicates that spatial attributes are taken into account surprisingly - even though the robot is not provided with any externally provided labels. On the right side, a mixture of places from both Fr (places $9-12$ and 16) and NC site (places $50, 52$ and 53) are grouped at this level. This is due to the similarity of the texture and color attributes between the visited room of the Fr site and the walls of the building at the NC site. At the last level, on the left side places belonging to the Lj site are grouped while on the right side, the places mostly belong to the corridor part of Fr site (places $1-5$, 7, 8, 13, 14 and 19, 20) are grouped together. Places belong to the NC site (places $46-49$, 51, $55, 59$) are also grouped together at this level. According to these results, we can see that the places of the Fr and NC site are differentiated with

respect to three different attributes while those of Sa and Lj sites are mostly grouped with respect to single or at most two attributes. This is expected since the robot's path in Fr and NC sites have more scene related variability in terms of visited places in comparison to the Sa and Lj sites.



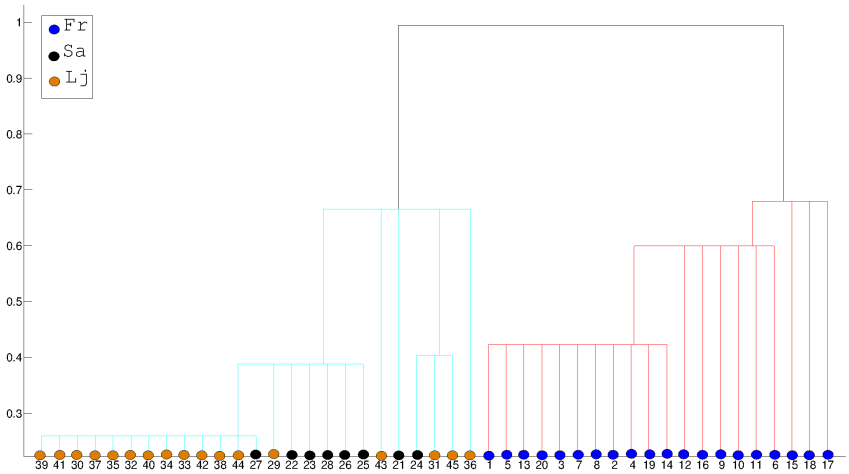
(a) Fr site.



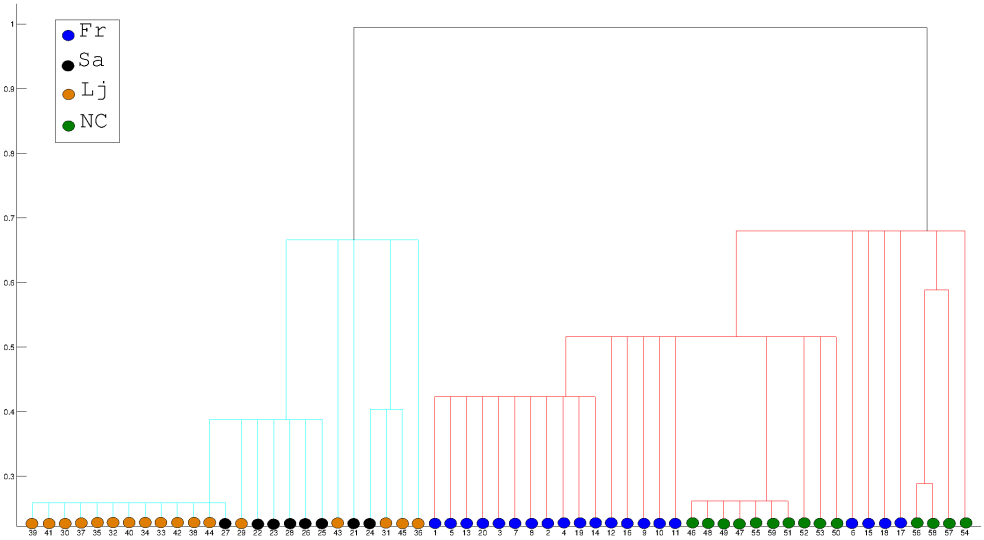
(b) Fr + Sa sites.

Figure 3.5. Long-term spatial memory: The evolution of the place memory after visiting Fr and Sa sites.

The topological map of each site is learned in parallel with the learning of the



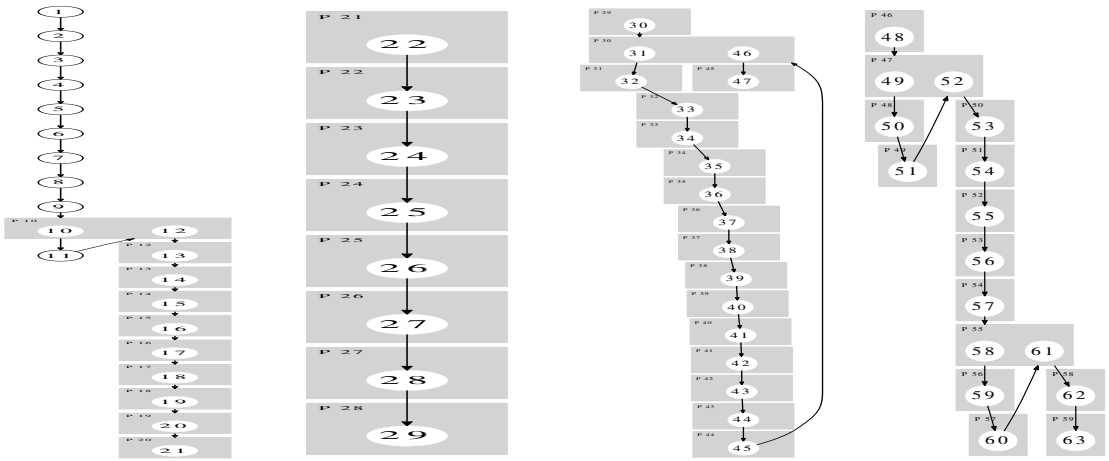
(a) Fr + Sa + Lj sites.



(b) Fr + Sa + Lj + NC sites.

Figure 3.6. Long-term spatial memory: The evolution of the place memory after visiting each site.

place memory. The Fr site map consist of 20 nodes as seen in Figure 3.7a where some detected places are viewed as belonging to the same place. For example, detected places 10 and 12 correspond to one single learned place (namely place 10) that is represented by a terminal node in the place memory. The topological maps of the resulting three sites are given in Figure 3.7b, Figure 3.7c and Figure 3.7d respectively. In cases when a detected place is recognized, no new node is added to the topological map. The



(a) Fr site. (b) Sa site. (c) Lj site. (d) NC site.

Figure 3.7. Long-term spatial memory: The topological maps of each site. Each node corresponds to one distinct place. Nodes whose place and detected place indices are identical are shown by simple circles where as nodes where that is not the case are indicated by shadowed boxes.

existing node is simply updated to include this association. It is observed that the most complex maps belong to the Fr and Lj sites - as Fr has the maximum number of different space units (those corresponding to distinct terminal nodes) while Lj has over-partitioning of the corridor area due to the zig-zag motion of the robot. The Sa site map has the least number of nodes. In the map corresponding to the NC sites, some of the detected places are recognized and no new node is added and it is observed that the whole site roughly divided into three parts. Altogether, the topological maps contain 59 nodes.

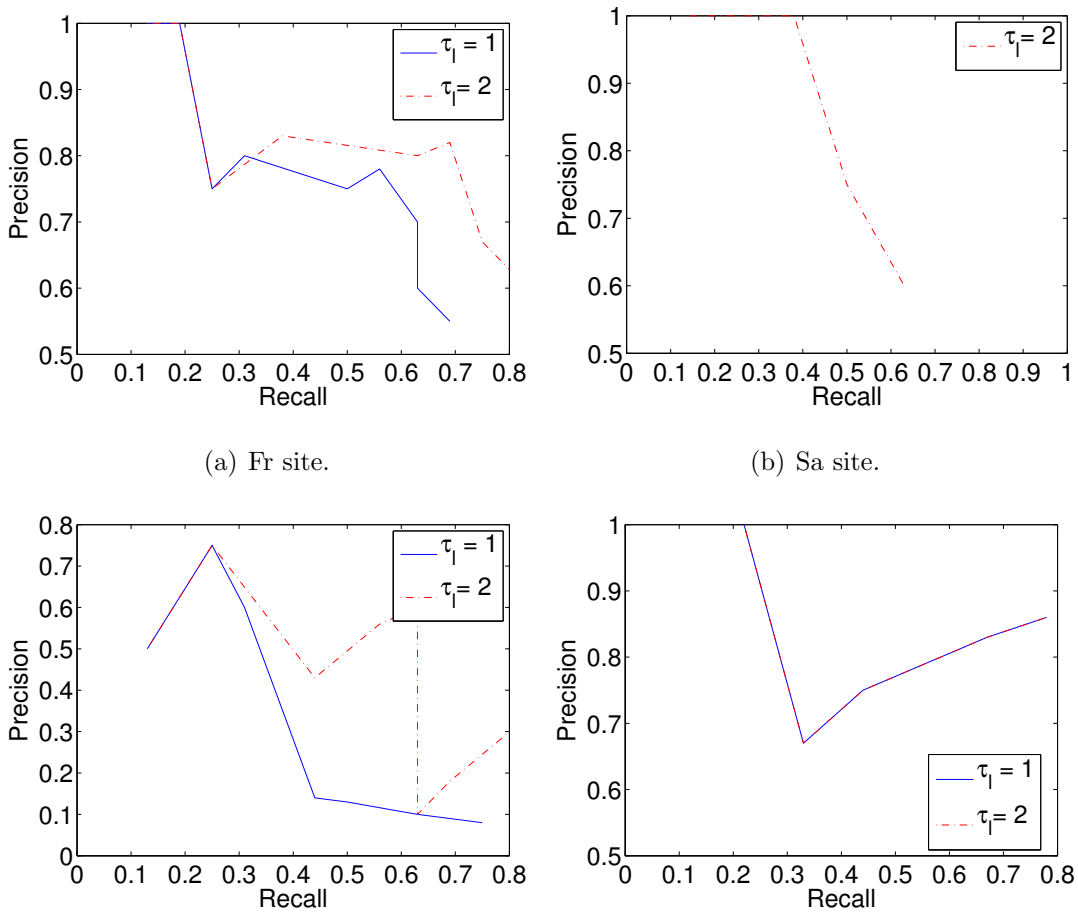
3.8.1.2. Second-Time Visits. The experiment continues with the robot revisiting each of the sites – again under cloudy illumination. In the first part of this experiment, the robot revisits only one of the learned sites. This experiment is repeated separately for each of the four sites. In this case, while the robot goes through (mostly) the same places, the exact path as well as the illumination conditions differ. Furthermore, at some sites, the robot visits some places for the first time. Of course, now the robot's long-term spatial memory is not empty. Rather, it contains knowledge that was learned

Table 3.3. Detected places in the second-time visits with the COLD+NC dataset.

Site	Detected Places	
	Total	Revisited
Fr	23	16
Sa	10	8
Lj	18	16
NC	9	9

in the first-time visit as shown in Figure 3.6b. As the robot revisits a site, in cases when the robot recognizes a detected place, its correctness is verified via manual inspection. The experiments are repeated with varying $\tau_l \in \{1, 2\}$ and recognition cost threshold $\tau_r \in [1, 2]$.

In revisiting the Fr site, the robot detects 23 places as given in Table 3.3 with indeces between $(60 - 82)$. Visual inspection reveals that 7 detected places are new while the remaining 16 places have been previously visited and should be recognized. The recall precision curves are given in Figure 3.8a. It is observed that the recognition rates are much better with $\tau_l = 2$ as compared to $\tau_l = 1$. This is expected since this site is grouped into 3 levels in the place memory. Thus, a full coverage of the place memory is obtained when $\tau_l = 2$. Our robot operates at 100% precision region in order to to minimize erroneous recalls. Thus, the robot - correctly - does not recognize the 7 places and adds them to the place memory. At the same time, only 2 of the possible 16 places are recalled while the remaining 14 are viewed as new and added to the place memory. Visual inspection reveals it is very difficult to relate these places to those previously learned based purely on appearance as there are major variations in the appearances due to viewpoint or illumination differences that are challenging even for a human. After revisit is complete, the place memory evolves with 21 new places added as seen in Figure 3.9 while the topological map is updated as in Figure 3.10. It is observed that place memory structure evolves such that the spatially nearby places are grouped closer. For example, newly detected places in the Fr site are stored close



(a) Fr site.

(b) Sa site.

(c) Lj site.

(d) NC site.

Figure 3.8. Recall-precision curves - Revisiting each site individually for the second-time after having learned Fr, Sa, Lj and NC sites.

to previously learned places from the same site. The topological map is updated so that only nodes associated with the newly learned places are added.

Alternatively, if the robot revisits Sa, the robot detects 10 places - 8 of which have been previously learned while the remaining 2 detected places are visited for the first time. With $\tau_l = 2$, with 100% precision, the recall rate is 40%. The recall rate becomes 65%. with 60% precision as shown in Figure 3.8b. For $\tau_l = 1$ the recall - precision rates turn out to be fixed at 38% with 100% precision for all τ_r values. The in-depth analysis of this situation reveals that new places in the Sa site are very similar to the previously learned places of Lj site. As a result, with $\tau_l = 2$, a confusion occurs

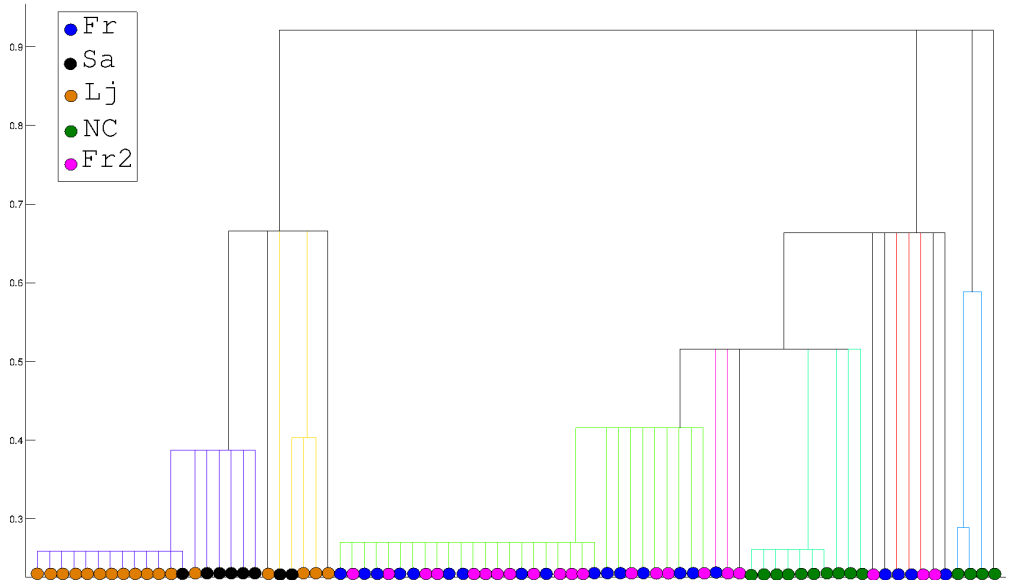


Figure 3.9. Long-term spatial memory: The evolved place memory after revisiting the Fr site. Color codes indicate different sites.

at the upper levels of place memory. With $\tau_l = 1$, the robot does not confuse places as it doesn't traverse the upper levels. When the robot revisits Lj, 18 places are detected - 16 of which are from previously learned places. The performance is lower in comparison to the previous sites as seen in Figure 3.8c. The maximum achieved precision rate is around 70% with a corresponding recall rate of 25%. This result is expected since the robot mostly navigates along the corridor with zig-zag motion. Thus, the corridor is split into many places that look similar to each other. Thus, the intra-place confusion is higher for this dataset and as a result precision rate is low.

Finally, the robot detects 9 places in the second tour of NC site - all of which have been previously visited. The NC site has very successful recognition results which are shown in Figure 3.8d. For both curves, we have a recall rate of nearly 25% at 100% precision. With 85% precision, the recall rate is around 80%. τ_l parameter do not affect the recognition. This is probably due to the detected places of the second tour of the NC site are similar to second and third level terminal nodes of the NC site. Thus, with $\tau_l = 1$ most of the associated nodes are traversed during recognition.

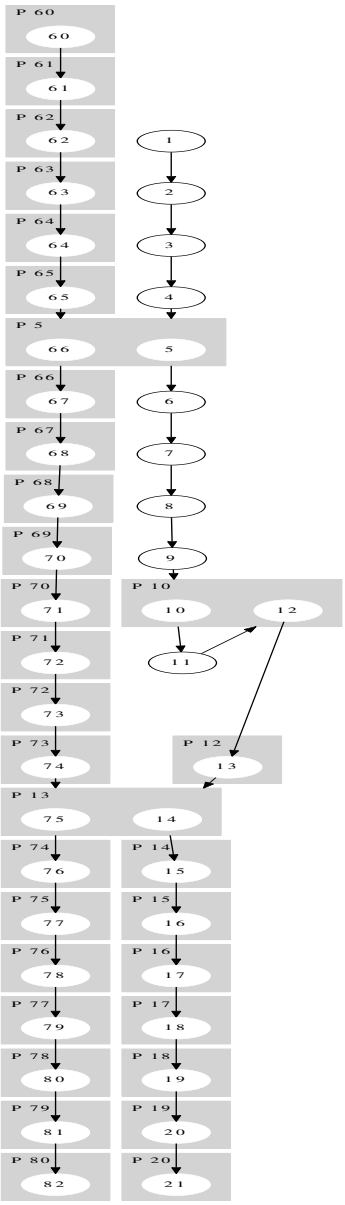


Figure 3.10. Long-term spatial memory: The updated topological map of Fr site after revisiting the Fr site.

Moreover, detected places of NC site are comprised of more base points and they are more distinct compared to indoor datasets. Therefore, the recognition rates are better compared to indoor datasets. Due to the space constraints, the resulting place memories and updated maps of these sites are not given graphically. These results indicate that performance is closely related to the informativeness of the incoming sensory data. In the Sa and Lj sites, the variability among the different places is low

compared to Fr and NC sites. As such, inter-site confusion occurs mostly between Sa and Lj sites.

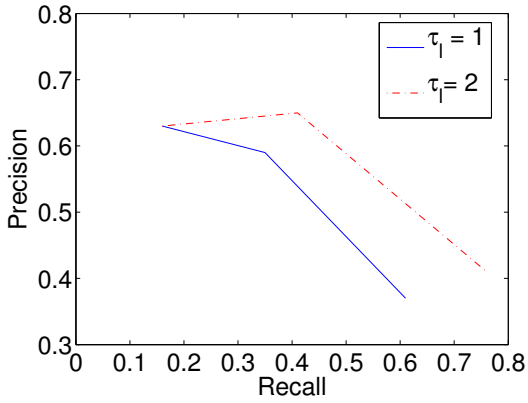


Figure 3.11. Recall-precision curves for combined revisit.

In the second part of this experiment, the robot revisits all the sites one after another instead of only one site. The recall precision curves are given in Figure 3.11. We can see that, the maximum obtainable precision rate is around 65% with a corresponding recall rate of 15%. As the robot revisits each site, the place memory structure becomes more complex with increased number of levels and terminal nodes. A more in-depth analysis reveals that most of the incorrectly learned places occur at the Lj site which decrease the overall recognition performance. Indeed this was also the case in the first part of the experiment and was attributed to the zig-zag motion of the robot that causes over-partitioning of the corridor area. Thus, the precision and recall rates are lower to compared to those of the first part where the place memory has a smaller structure. Of course, increasing the place memory traversal parameter τ_l clearly boosts the recall rate as it increases the chance of finding the correct terminal node for each newly introduced place.

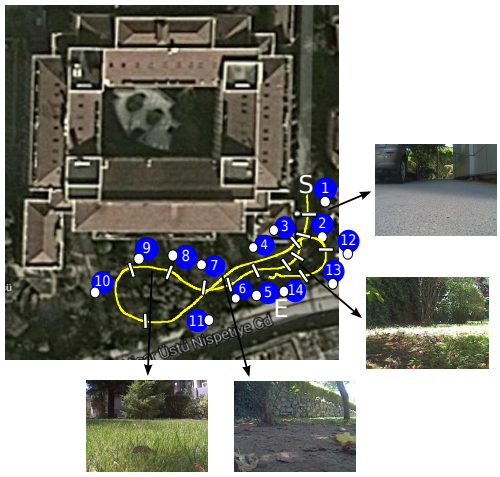
Table 3.4. Parameter settings for Jaguar experiments.

τ_μ	τ_σ	τ_κ	τ_n	τ_w	τ_p	τ_h	τ_r	τ_l
0.15	0.003	0.56	5	3	20	0.25	1	2

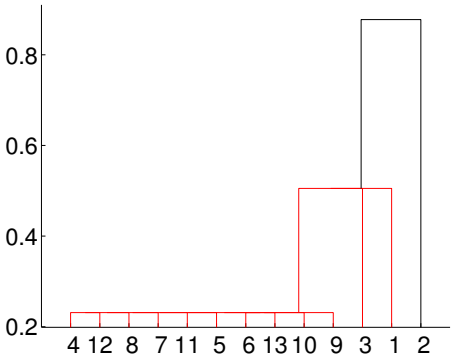
Figure 3.12. Jaguar robot.

3.8.2. Jaguar Robot

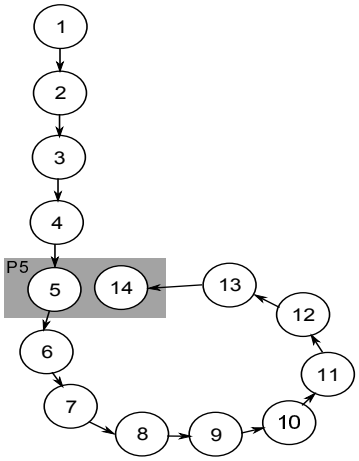
The second set of experiments are done with our Jaguar robot as shown in Figure 3.12. The spatial cognition system operates on the visual data collected while navigating outdoors. As visual sensing (both the hardware and the acquisition geometry) is different from that of the first experiment, the parameters are adjusted manually as shown in Table 3.4. In the first tour, the robot follows a path of approximately 175 meters as shown in Figure 3.13a and collects data at 3200 base points. Scenes from sample bases along the robot's path are given in Figure 3.13a with noticeable elevation differences along the route. The robot finds data from 28 base points to be uninformative. Of the remaining, 14 places are detected as shown in Figure 3.13a. The learned place memory as seen in Figure 3.13b. consists of three levels. It is observed that all places except the first three are clustered together. This is expected as they share similar spatial and/or semantic attributes as shown in Fig 3.13. On the other hand, places 1, 2 and 3 are from the car park area and thus are clustered separately. Finally, the detected place D_{14} is recognized as place 5 and the place 5 is updated accordingly. This update is indeed correct since the robot goes through the same place twice - once at the onset and once at the end of its tour. The resulting topological map is as given in Figure 3.13c. As expected, detected places D_5 and D_{14} are represented by a single node in the topological map.



(a) Robot's path and detected places.

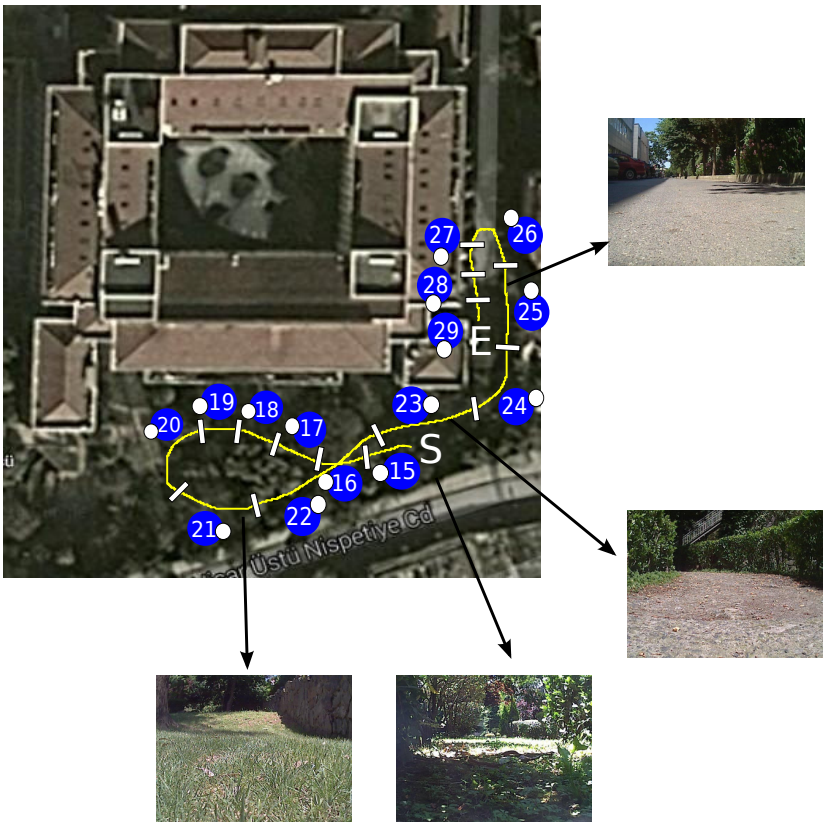


(b) Long-term spatial memory: place memory.

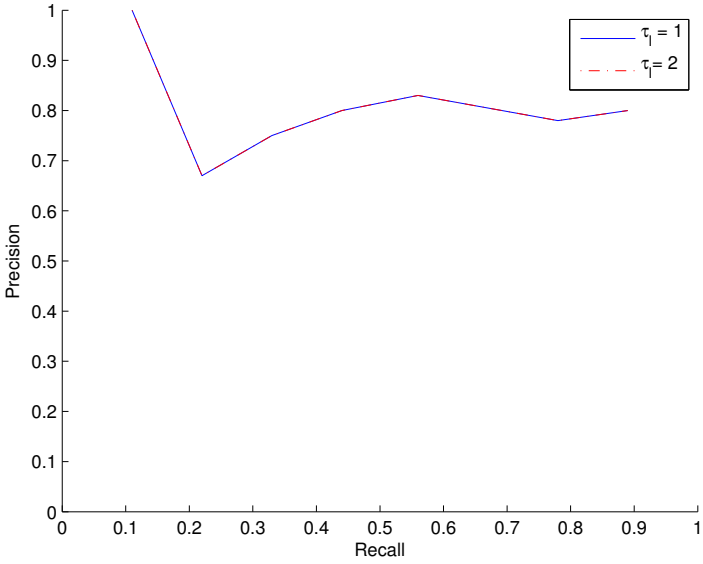


(c) Long-term spatial memory: topological map.

Figure 3.13. First-time tour results with the Jaguar robot.

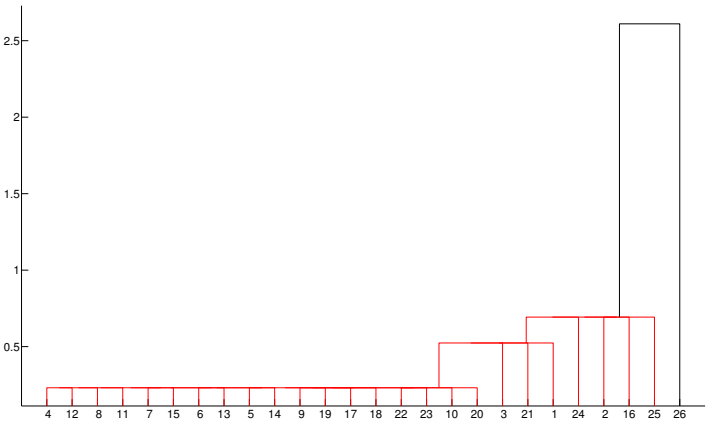


(a) Robot's path and detected places.

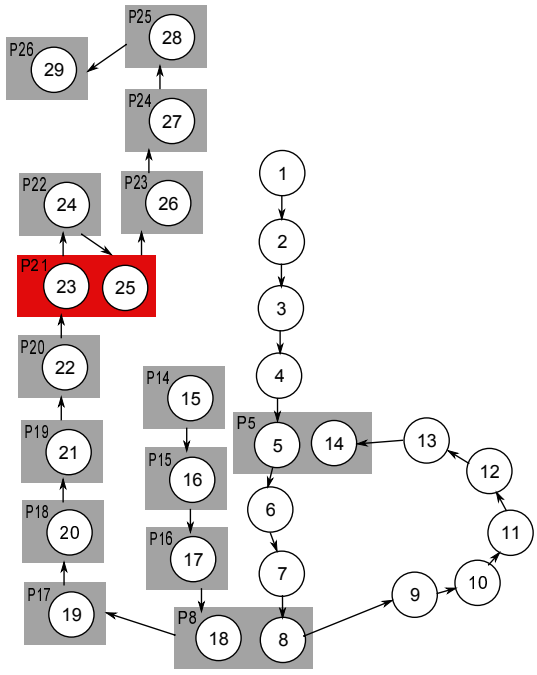


(b) Recall-Precision Curves.

Figure 3.14. Second-time tour results with the Jaguar robot. Detected places and recall-precision curves.



(a) Long-term spatial memory: Updated place memory.



(b) Long-term spatial memory: Updated topological map.

Figure 3.15. Second-time tour results with the Jaguar robot. Long-term spatial memory.

Following, the robot goes through a second-time tour of approximately 150 meters as shown in Figure 3.14a. This time, the robot goes through 2550 base points along this tour - of which 82 base points are found to be uninformative. In this case, the robot detects 15 places as shown in Figure 3.14a. As this path has certain overlapping regions with the first tour, we expect the robot to recognize 9 of these places – as they have been previously visited. The precision-recall curves with $\tau_r \in [0.9, 1.7]$ are given in Figure 3.14b. The recognition performance is evaluated for $\tau_l \in \{1, 2\}$. It is observed that with 100% precision, the recall rate is 10%. With 80% precision, recall increases to 60%. The effect of level parameter τ_l is negligible for this dataset due to the low number of levels in the place memory tree.

The updated place memory and topological map are as presented in Figure 3.15a and Figure 3.15b respectively. The place memory now has four levels. At the root node, place 26 is split from the other places like an outlier. Indeed it is the case since the robot only sees the wall of the building from this place. The places that belong to car park area ($1-3$, 21, 24 and 25) are placed at second and third levels. The places at last level are mostly belong to the garden area. Interestingly, a place that belongs to car park area 23 also reside at this node. This can be due to the ground color and vegetated area at the car park which is similar to the color of walkway and vegetation in the garden area. From the topological maps, we observe that there is a terminal node update at place 8 since detected place D_{18} from Tour2 is recognized as place 8 which is spatially correct. Unfortunately, there is a false node update at detected places D_{23} and D_{25} which is shown in red. This false node update is probably due to the similarity of their appearances as observed in Figure 3.14a. Other than that, the obtained results are indeed satisfactory since whole the process is handled in an autonomous manner without any human intervention.

3.8.3. On-Robot Implementation

The proposed model has been implemented and tested on Jaguar robot platform for real-time operation. The robot has a threaded motion system with a maximum speed of 1.4 meters/second. The robot has on-board encoders, Hokuyo laser range

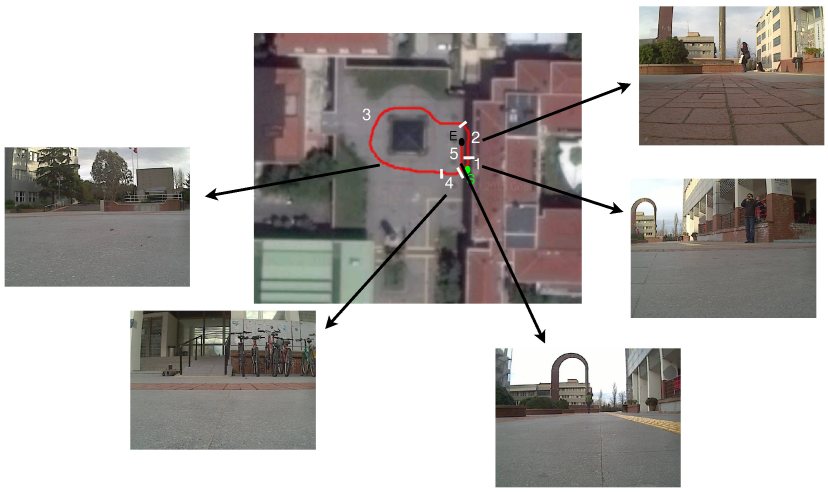
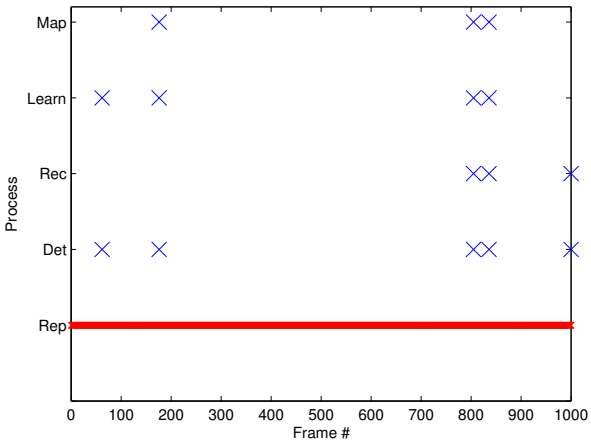


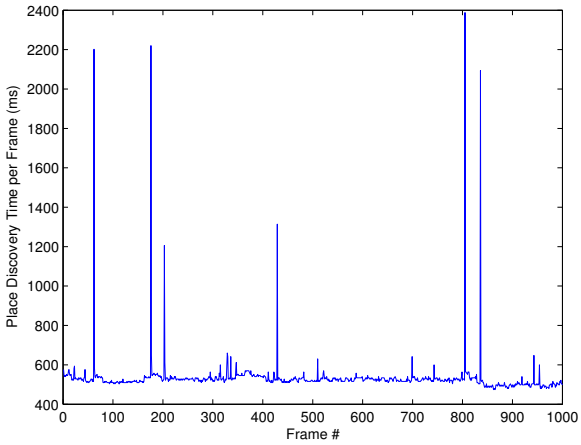
Figure 3.16. Robot's path with S (start) and E (end) points with the detected places 1-5 as shown by their extents as indicated by short white markers and sample scenes.

scanner with 30m field of depth, standard front camera, PTZ camera, GPS, IMU and on-board computer. In this implementation, only the front camera of the robot is used as sensory input. The on-board computer is a nettop computer with AMD E450 CPU. It has a fairly low processing power (5 times slower compared to a modern i7 processor) but consumes very low power (around 15 Watts on the average) and can be placed on-board the robot easily thanks to its compact form factor. A remote computer is used for observing the robot's current state and give commands if necessary. Notice that there is a wired unidirectional communication between the sensors and the robot while on-board computer has wired bi-directional and the remote computer has wireless bi-directional communication.

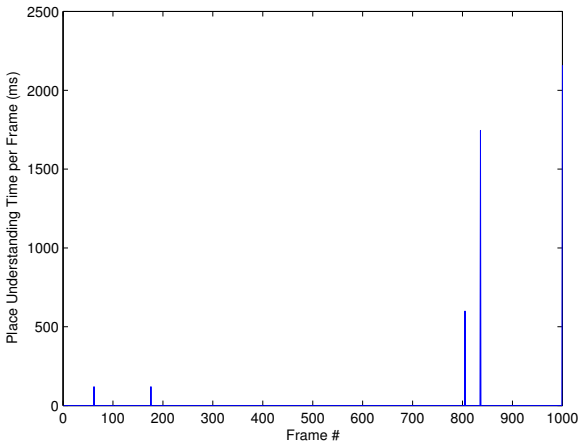
The experiments are done in one of the campuses of our university. The robot follows a path of 100 meters as shown in Figure 3.16 with an average speed of 22 cm/sec while simultaneously running the TSC model on its on-board processor. The motion commands are sent by a human operator using remote laptop computer. The start and end of the tour overlap to some extent in order to test whether the robot is able to understand that they are actually same places. The progression of events with each incoming frame throughout robot's navigation is given in 3.17a. It acquires 1000 image frames during its tour. It is observed that representation events occur with each



(a) Progression of events.



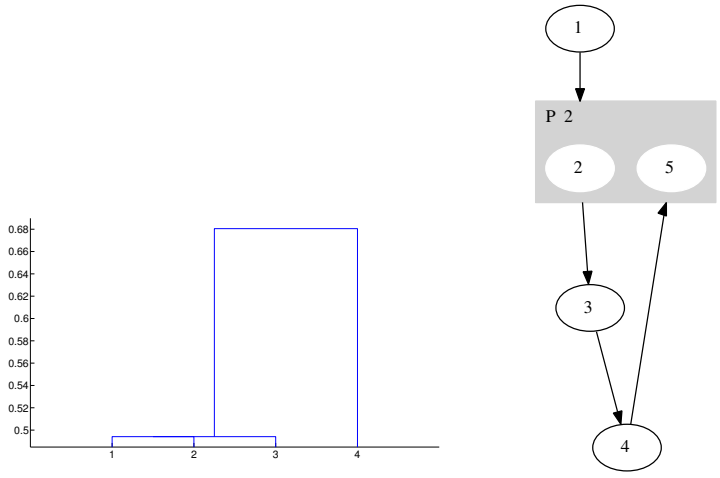
(b) Place Discovery.



(c) Place Understanding.

Figure 3.17. Spatial cognition events and the processing times (msec) per frame.

frame. The remaining events occur more occasionally. For example, place detection event occurs 5 times. The extents of the detected places are shown overlaid in Figure 3.16, but of course these are unknown to the robot as the start of its operation. The robot detects them completely on its own. The robot recognizes the 5^{th} detected place as being the same that of the second detected place - as indeed is the case. Therefore, only recognition event is generated and the robot updates its long-term spatial memory accordingly. The place memory evolves to contain 4 learned places as seen in Figure 3.18a while the topological map evolves to encode the spatial relations among these places as shown in Figure 3.18b. As determined, the detected place 5 is associated with the 2^{nd} learned place. Furthermore, the hierarchical structure of the place memory is meaningful. Places 1, 2 and 3 are close to each other as the robot sees the horizon without any occlusion. Therefore they are placed on the same branch. On the other hand, in place 4, the robot mainly sees the exterior of a building. Thus it is placed further away in a different branch.



(a) Place memory. (b) Topological map.

Figure 3.18. Robot's long-term memory after the experiment.

We have also analyzed the robot's computational performance. The per frame processing times vary depending on the spatial cognition activities of the robot. In the place discovery node, processing times are found to be in the range of 500-550 ms on average with occasional spikes of around 2.2 seconds as seen in Figure 3.17b.

These spikes correspond to frames where a place detection event occurs as seen in Figure 3.17a. It is observed that the maximum spike is observed when the 3^{rd} place is detected - which suggests that the extent of the detected place affects the processing time as expected. Notice also that while there are more than 5 spikes in the figure, only 5 places are detected. This is because place detection checks for the informativeness, reliability and plenitude, some sensory data may be treated as problematic and ignored as explained in [12].

In the place understanding node, processing times are much lower with the exception of 5 spikes as seen in Figure 3.17a. These spikes are attributed to recognition, learning and mapping processes that take place as the robot detects places. Notice that the processing times associated with the first two detected places are very small (around 100 ms) as only the place memory is initialized. As more places are detected, other modules become more involved. For 3^{rd} and 4^{th} detected places, all the modules are activated as these are first time visits and the long-term memory is available for recognition, learning and mapping. For the 5^{th} place, as it is recognized as 2^{nd} learned place, only the recognition module is invoked at this node. As the robot continues its tour and more places are detected, the processing time in the place understanding node increases. This is expected as long-term memory expands accordingly.

In the resulting statistics as given in Table 3.5, it is observed that place discovery node takes on average 532.5 ms of processing per frame while place understanding node takes only 4.75 ms per frame. This is because place understanding is activated only when a place is detected as seen in Figure 3.17c. Thus, per frame basis, its computational overhead is very low. Altogether, real-time performance is around 2 frames per second. These results are very promising – as they have been obtained with a fairly low processing power and no software optimization and indicate that the developed approach could be used in real-time applications.

Table 3.5. Processing time statistics of ROS nodes.

ROS Nodes	ms/frame		
	Mean	Min	Max
Place Discovery	532.5	474	2388
Place Understanding	4.75	0	2160

3.8.4. Summary

Our experimental results on three different data sets (COLD, New College and Jaguar) suggest that the robot is able to develop its knowledge of places and their spatial relations as it visits different environments. Detected places are observed to enclose groups of base points having coherent appearances after visual inspection. As expected, they are affected by robot's sudden or jerky movements. For example, robot's zig-zag motion in Lj dataset causes redundant partitioning of the corridor area. In the place memory, appearance-wise similar places are close to each other while structurally different sites such as Lj and NC are further away. The topological maps also evolve to be meaningful since edges are setup between places that are geographically adjacent. The resulting recognition performance is quite comparable with state-of-the-art approaches. For example, with the NC dataset, we obtain around 23% recall with 100% precision with our concept of 'places'. Localization results for the same data set are reported to be 16% with 99.5% precision in [69] where each location is viewed as a distinct place. In another related work based on key-places as determined from the same dataset, the recall rate is around 12% with 100% precision as reported in [70]. All indicate that robot is able to develop its topological spatial cognition abilities reliably - even if its long-term spatial memory is evolving in an organized, incremental and unsupervised manner.

Computational performance-wise, the first module takes on average 340 milliseconds per base point and is always activated. The activations of the remaining modules are intermittent since their operations are conditional. The recognition module is

invoked only if a place detection occurs and takes around 10 milliseconds with the current place memories. Thus, the activation of the first two modules requires about 350 milliseconds. The remaining two modules (learning and mapping) are activated less frequently – as their activation depends on the robot detecting a place, but not recognizing it. It is observed that a place is composed of 170 base points or so on average. In this case, adding a new place to the place memory and incorporating it into topological map take around 6.8 seconds which implies 40 milliseconds per base point. Thus, with all 4 modules activated, the robot requires on average 390 milliseconds per base point to complete its processing. Of course, this time will be as short as 350 milliseconds for most of the base points in case of recognition. All the other modules are implemented in MATLAB and there is no software optimization. This suggests that even without any software optimization, assuming the distance between two consecutive base points to be around 50 cm, the robot is able to navigate with a reasonable speed (4.6 km/hour) with all modules being active in real-time.

3.9. Conclusion

In this chapter, we consider the topological spatial cognitive abilities of mobile robots. We present an integrated model in which the concept of a 'place' is defined as a collection of locations sharing common perceptual boundaries based on appearance coherency. The novelties of this model are two-fold: First, it explicitly incorporates a long-term spatial memory where two separate, but related types of knowledge are stored. The place memory organizes the learned places in a hierarchy based on their appearance-related similarities while the topological map simply encodes their spatial relations. It enables the robot to efficiently store and retrieve the information that was learned. It also provides the framework to which the robot is able to link new knowledge by association. Second, the processing modules operate together so that the robot builds its spatial memory or utilizes it in an organized, incremental and unsupervised manner. In particular, the robot detects places via partitioning the sequence of the associated descriptors based on their coherency while pruning out uninformative or scanty data. It then attempts to recognize each detected place via relating to its long-

term spatial memory. In case of recognition, its already existing knowledge is updated accordingly. In case of no recognition, it invokes place learning and mapping as to incorporate the new place and its spatial relation. A series of experiments demonstrates the proposed model to be highly effective for topological spatial cognition. The hierarchical structure of the place memory is observed to be related to semantic attributes where similar places end up in the same branch of the place memory. In parallel, the memory of spatial relations encode how to reach the learned places. Together, they contribute to the robot's awareness of its surroundings.

Moreover, this knowledge is quite effective for recognition purposes in future visits. Results show that similar precision and accuracy performance is obtained in both indoor and outdoor environments with performance comparable with the state-of-art approaches. As the robot's long-term spatial memory evolves completely on its own while learned knowledge is organized in a manner that is amenable for higher-level semantic reasoning, this is a step forward towards having robots that are capable of interacting with their environments throughout (possibly life-long) operation. Furthermore, the on-robot experimental results on Jaguar robot equipped with inexpensive processors demonstrate that the robot is build its long-term spatial memory with acceptable computational performance. As such, the robot becomes aware of its surroundings as it is moving around.

There are several ways in which the work presented in this chapter can be extended. First, we will consider the adaptation of the associated parameters throughout robot's operation. Second, we plan to extend the proposed model to incorporate semantic reasoning and categorization. This will require expanding both the processing and long-term spatial memory parts of the model. The processing part will expand to include a semantic analysis module that is capable of in-depth parsing of a given place and a categorization module that is capable of extracting common semantic properties as to infer abstractions of spatial units. In parallel, the long-term spatial memory will be be expanded to include the learned semantic knowledge including levels of abstraction. Finally, we plan to unite robot's spatial cognition and motion capabilities so that the robot will be able to coordinate spatial reasoning and navigation.

4. MERGING APPEARANCE-BASED SPATIAL KNOWLEDGE

4.1. Introduction

This chapter considers the problem of merging appearance-based spatial knowledge of multiple robots operating independently. The goal is to have the robots expand their spatial knowledge accordingly so that they are able to reason about places - even if they have not been actually visited. Most related work approach the problem as how to merge multiple maps that have been independently built by different robots [97,98]. With appearance-based SLAM or topological maps, merging is formulated as identifying edges that connect nodes belonging to different maps via finding pairs of similar images. As such, the scalability of these methods with respect to both spatial extent and the overlap of these maps turns out to be problematic.

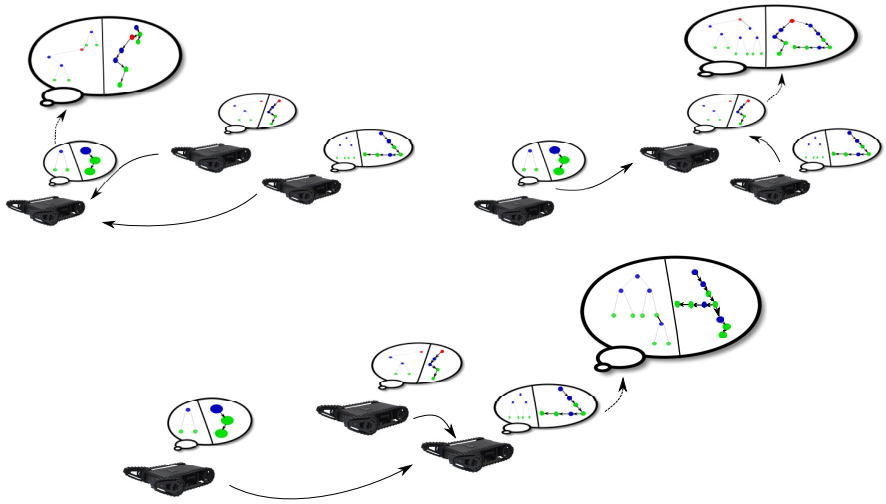


Figure 4.1. Merging of spatial knowledge in a team of 3 robots. Each robot communicates with each of the remaining robots and exchanges its spatial knowledge. Next, it merges its place memory with that of the other robot. This is followed by merging of map memories. As a result, each robot's spatial knowledge expands accordingly.

In this chapter, we present a novel approach where the problem is considered as the merging of appearance-based spatial knowledge that is comprised of place and map memories. It is assumed that each robot has its individual spatial knowledge. In the merging process, each robot communicates with each of the remaining robots one-by-one and receives the spatial knowledge of the other robot as shown in Figure 4.1. Once this is over, it first merges its place memory with that of the other robot. It then incorporates other robot's map memory into its own via adding the new spatial relations as appropriately. There are two aspects that differ from previous related work on map merging:

- First, a 'place' is defined to be a collection of appearances sharing common perceptual signatures or physical boundaries. This is in contrast to viewing each appearance as a single place.
- Second, each robot's place memory is processed as a whole or in portions. This is in contrast to processing each appearance individually.

The advantages of such an approach are three-fold: First, it scales easily with respect to the amount and overlap of the appearance data. This is because it does not require matching pairs of appearances from two different maps. Furthermore, the expanded place knowledge continues to be organized as a semantic hierarchy that is amenable to human-like interpretations and higher-level symbolic reasoning. Finally, it can be applied in a decentralized manner by all the robots individually.

4.2. Related Literature

One of the paradigms commonly used in distributed intelligence is the knowledge-based paradigm. The focus in these approaches is on knowledge sharing between robots with the objective of easily allowing them to share and understand knowledge from disparate sources. This problem is different from cooperative mapping where multiple robots concurrently and continuously contribute their data to a single map [99, 100]. Previous related work on multi-robot spatial knowledge sharing have primarily focused on how to merge maps that have been generated [101]. The proposed method vary

depending on whether the maps are either of metric or topological nature. In direct metric approaches, the initial poses of the robots are assumed to be known [102,103] or it is assumed that the robots can identify, rendezvous and communicate with each other when in line of sight [100,104–110]. Thus, these approaches extend the existing SLAM methods by using the pose and other information coming from each robot. When the robots are aware of each other, these approaches turn into cooperative mapping using Bayesian filters such as Kalman Filter, Particle Filter, Extended Kalman Filter etc. for reasoning. Each robot maintains and updates its local map by observing landmarks and maps are merged in a centralized fashion by matching observed features and applying geometric transformations. Afterwards, the global map is shared among robots and the robots start to update the map cooperatively. The computational requirements of these methods increase tremendously when the number of observed and tracked landmarks increases. Moreover, SLAM methods require the knowledge of other robots' locations and synchronous sensory data updates at all times in order to generate a global map [97, 100]. These methods are also susceptible to local inconsistencies and data association problems. Therefore, mapping large places can be problematic.

In order to overcome these problems, alternative methods such as Expectation Maximization that are robust against data inconsistency have been proposed [104,111]. While large places can be efficiently mapped, initial positions of the robots are required to be close to each other in order to have maps merged successfully. In contrast, with indirect metric methods, individual maps are merged via aligning them with any positional knowledge using a variety of approaches such as random walks, laser scan integration methods or genetic algorithms [112–121]. However, as the search process is costly, methods that aim to alleviate this have been proposed such as such as Hough transformation based matching [101,122,123] or pose-graph matching [119,121]. While computational requirements are considerably reduced, the complexity and scalability of metric maps often prohibits efficient application in large-scale environments [124]. Furthermore, even with appearance-based or topological SLAM methods, each location is considered separately as a place [35, 68] or a representative location (key-place) is selected after grouping visual data from different locations [70]. However, such a representation does not correspond to a concept of a place defined a specific semantic

entity such as 'being in X lab'.

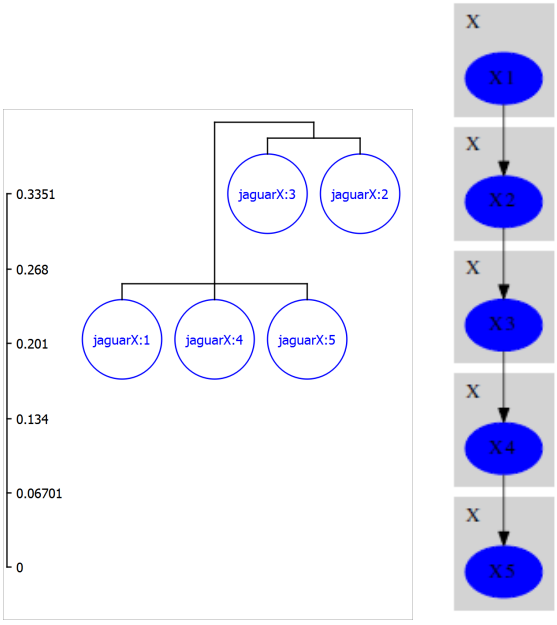
Alternatively, appearance based maps are merged by finding correspondences among them with the approaches varying with respect to the information used. Feature based approaches use the local features for this [120, 125, 126] while appearance-based approaches use the whole scene appearance in order to find correspondences among nodes via comparing the respective image pairs [98]. The quality of a merged map is measured via a metric based on algebraic connectivity that encodes entanglement – namely the amount of effort needed to split the merged map back into two separate maps. This is sensible in all this work since each node (place) of the topological map corresponds to one single appearance. However, again scalability with respect to the number of nodes in the map is problematic. Furthermore, again such a representation does not correspond to a concept of a place defined as a collection of appearances sharing common perceptual signatures or physical boundaries. Thus, it is much harder to have a robot that can be aware of its surroundings. Furthermore, these approaches do not cover all degrees of overlap between maps and knowledge. In this work, we present an approach that addresses both issues simultaneously.

4.3. Spatial Memory

Spatial memory retains the learned knowledge of places and their spatial relations. Since they are of different nature, each is stored in a different part of the memory. The knowledge of places is retained in place memory while that of spatial relations is encoded in the map memory.

4.3.1. Place Memory

Suppose that the robot has learned a set of places as indexed by $\mathcal{P}$. There are three aspects to a robot's place memory: how a place is internally represented, the organization of the knowledge associated with the learned places and how this knowledge is used in recognition.



(a) Place memory. (b) Map memory.

Figure 4.2. Long-term spatial memory example.

As discussed, a 'place' consists of a set of appearances sharing common visual features or physical boundaries. Thus, it does not correspond to a single image (and thus single location), but rather a particular spatial area. Places are detected using a place detection algorithm [12,66]. It should be noted that in general detecting a place does not imply its recognition - with few notable exceptions such as [67]. In all, the learned knowledge must be retained for future referral.

The place memory T organizes the learned places $\mathcal{P}$ in a tree hierarchy - as developed previously in [14]. This hierarchy is defined by a nested sequence of partitions of $\mathcal{P}$ based on the associated descriptors. The place memory T evolves as the robot learns new places. This structure is also viewed as encoding a semantic hierarchy.

The robot uses its place memory for recognition - using an approach as presented in [13]. When in a newly detected place, it scans through its place memory in order to find a terminal node that relates to the current place. It will be able to find such a node only if the detected place has been previously visited and learned. Otherwise, there will

be no recognition. There are three aspects to recognition – namely the decision-making at each non-terminal node N of the tree structure, how to traverse the its place memory and the updating of place knowledge in case of recognition. The decision-making at each node N is based on minimizing a cost function g_N. The cost function measures how unlikely the given place P is to be associated with one of children nodes $N^{\downarrow}$.

$$g_N(P) = \tilde{\gamma}(P, N^1(P)) + \frac{\tilde{\gamma}(P, N^1(P))}{\tilde{\gamma}(P, N^2(P))} + (1 - V(N^1, P)) \tag{4.1}$$

The first term $\tilde{\gamma}(P, N)$ measures the dissimilarity of the given place to the places associated with the node based on the respective descriptors. The greater this value is, the more dissimilar the detected place to the places associated with node N. The second term indicates the reliability of this. The terms $N^1(P)$ and $N^2(P)$ are the two offspring nodes of N that are most similar to P as: If P is equally similar to the two nodes $N^1(P)$ and $N^2(P)$, then this term increases. Otherwise, it decreases. The third term measure the vote percentage that does not associate P with the node. The function V computes the overlapping volume between two hyperspheres. The cost value is checked against a recognition cost threshold τ_r.

$$g_N(P) \leq \tau_r \tag{4.2}$$

In case this condition is satisfied, the detected place can be recognized. Otherwise, the place is declared to be unrecognized and learning is invoked. When a place is recognized, the associated terminal node $N^1(P)$ is updated including member base points, the mean bubble descriptor $\bar{I}_{N^1}$. As such, place memory enables efficient storage and retrieval of learned place knowledge. Furthermore, the hierarchical structure provides a basis for semantic analysis and understanding.

4.3.2. (Topological) Map Memory

In parallel, map memory stores the spatial relations among the learned places - similar to [66, 93]. It is defined by an evolving undirected graph $G = \{\mathcal{P}, E\}$ with

nodes $\mathcal{P}$ and edges E. The nodes correspond to the different places that are stored in place memory while the edges represent the adjacency relations between the different places. As places are already stored in the place memory, the map memory simply consists of edges as defined by place tuples $ij \in E$. Namely $ij \in E$ if and only if the robot has navigated from the place i to the place j for $i, j \in \mathcal{P}$.

4.4. Merging of Spatial Knowledge

Now consider a set of N_r robots as defined by the index set $R = \{1, \ldots, N_r\}$. Suppose that the robot $m \in R$ has learned $\mathcal{P}^m$ places so far and its place memory is denoted by T^m while its map memory is denoted by G^m. The merging process is done in a decentralized manner by each robot m separately. It consists of three steps that are repeated until it communicates with every other robot one-by-one and merges its spatial knowledge with that of the other.

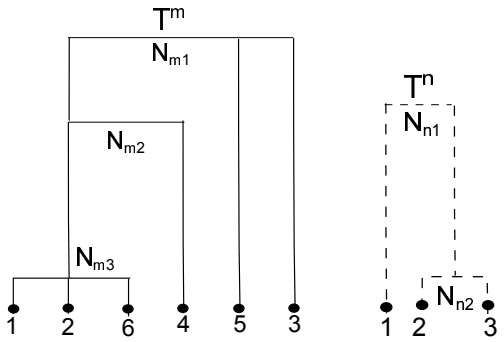
(i) It communicates with another robot $n \neq m$ and receives the other robot's spatial memory T^n and G^n.
(ii) Once communication with robot n is over, it first merges its place memory T^m with T^n.
(iii) This is followed by merging its map memory G^m with G^n.

4.5. Merging of Place Memories

Suppose that robot m has communicated with robot $n \in R$, $n \neq m$ and has received its place memory as represented by T^n. A sample scenario is shown in Figure 4.3. From the perspective of robot m, the goal is to update its place memory as to incorporate the new knowledge as

$$T^m(k^m + 1) \leftarrow T^m(k^m) + T^n(k^m)$$

where k^m denotes the update index of robot m and $+$ denotes the merging operator. In the sequel, we will omit it for notational simplification. There are two requirements



(a) Place memory T^m. (b) Place memory T^n.

Figure 4.3. Place memories of robots m and n respectively. Robot m knows 6 places while robot n knows 3 places. Note that an identical index across both memories will not necessarily refer to the same place.

for the merging process:

- First, the resulting place memory $T^m(k^m + 1)$ should incorporate the learned places $\mathcal{P}^m \cup \mathcal{P}^n$ of both of the robots.
- Second, $T^m(k^m + 1)$ should have a structure that is similar to the place memory generated directly from $\mathcal{P}^m \cup \mathcal{P}^n$ as much as possible.

Such a problem is considered as a case of distributed hierarchical clustering problem. Our proposed approach is based on a modified version of 'RACHET' algorithm [10].

First, two descriptive statistics are associated with each node N in the tree structure T^m. The first is the centroid $c_m(N)$ defined as:

$$c_m(N) = \frac{1}{\|\mathcal{P}^m(N)\|} \sum_{p \in \mathcal{P}^m(N)} \bar{I}_p \tag{4.3}$$

where $\mathcal{P}^m(N) \subseteq \mathcal{P}^m$ is the set of places associated with the subtree of T^m having node N as its root. The second one is the radius $\rho^m(N)$ - defined as the average distance

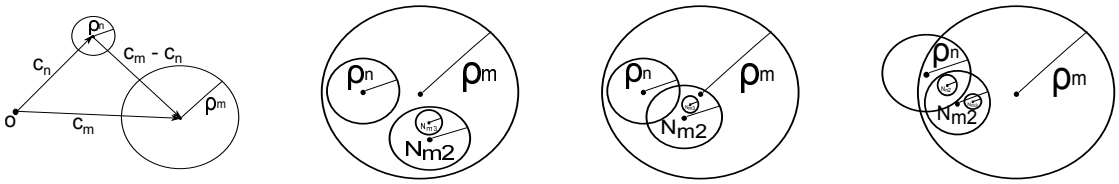
between its centroid descriptor and the places (terminal nodes):

$$\rho_m^2(N) = \frac{1}{\|\mathcal{P}^m(N)|} \left\| c_m - \sum_{p \in \mathcal{P}^m(N)} \bar{I}_p \right\|^2 \tag{4.4}$$

The centroid and the radius together define a N_I dimensional hypersphere $S(c_m(N), R_m(N)) \subset R^{N_I}$ be with center $c_m(N)$ and radius $\rho_m(N)$. If N is the root node, then the hypersphere $S(c_m(N), R_m(N))$ is a covering for the place memory T^m. Let this hypersphere be denoted by $S(c_m, \rho_m)$. The hyperspheres are related to $S(c_m, \rho_m)$ as:

$$S(c_m(N), R_m(N)) \subseteq S(c_m, \rho_m)$$

The merging is based on the extent and nature of the overlap of the two hyperspheres $S(c_m, \rho_m)$ and $S(c_n, \rho_n)$.



(a) Case 1. (b) Case 2-1. (c) Case 2-2. (d) Case 3.

Figure 4.4. Relation between $S(c_m, \rho_m)$ and $S(c_n, \rho_n)$.

4.5.1. Hyperspheres Non-Overlapping

The first case occurs when robots have visited different sites so that the places $\mathcal{P}^m$ in the place memory T^m are well separated from the places $\mathcal{P}^n$ in the place memory T^n. In this case, the corresponding hyperspheres do not intersect at all as shown in Figure 4.4a:

$$S(c_m, \rho_m) \cap S(c_n, \rho_n) = \emptyset$$

As such, the case is checked as follows:

$$\|c_m - c_n\|^2 > (\rho_m + \rho_n)^2 \tag{4.5}$$

4.5.2. One Hypersphere Contained

The second case occurs when the two robots visit places that appear mostly similar. As a result, one hypersphere is a subset of the other. Note that Without loss of generality, assume that the place memory T^m of robot m is associated with the larger memory:

$$S(c_n, \rho_n) \subset S(c_m, \rho_m)$$

Two subcases are possible depending on whether the hyperspheres associated with the hyper-terminals intersect or not. Hyper-terminal consist of inner nodes that have terminal nodes as children or the union of all children nodes of an inner node that are terminal nodes themselves. For example, there are three hyper-terminals in T^m in Figure 4.3 of the place memory of robot m while T^n has two hyper-terminals.

- **Case 2-1:** There is no hyper-terminal N of T^m such that the associated hypersphere intersects with $S(c_n, \rho_n)$ as shown in Figure 4.4b.

$$\forall\, N \text{ of } T^m \text{ s.t. } S(c_n, \rho_n) \cap S(c_m(N), \rho_m(N)) = \emptyset$$

- **Case 2-2:** There is an hyper-terminal N of T^m such that the associated hypersphere intersects with this hypersphere as shown in Figure 4.4c. Namely,

$$\exists N \text{ of } T^m \text{ s.t. } S(c_n, \rho_n) \cap S(c_m(N), \rho_m(N)) \neq \emptyset$$

The merging is done so that the smaller memory is incorporated into the larger one. Namely,

$$\begin{aligned} T^m(k^m+1) &\leftarrow T^m(k^m)+T^n(k^m) \\ T^n(k^n+1) &\leftarrow T^m(k^n)+T^n(k^n) \end{aligned}$$

Thus, the merged place memories of the two robots turn out to be the same.

4.5.3. Intersecting Hyperspheres

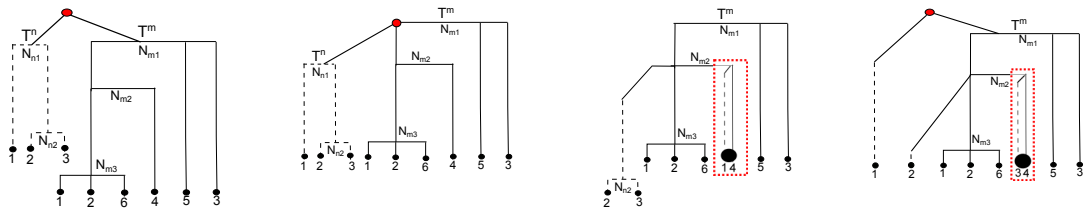
The third case occurs when appearance-wise some of the places learned separately by the two robots are overlapping while some are completely different. As a result, the corresponding hyperspheres intersect with none being a subset of the other as shown in Figure 4.4d:

$$\begin{aligned} S(c_m,\rho_m)-S(c_n,\rho_n) &\neq \emptyset \\ S(c_n,\rho_n)-S(c_m,\rho_m) &\neq \emptyset \end{aligned}$$

This can be checked as:

$$\rho_n^2 < \|c_m - c_n\|^2 < (\rho_m+\rho_n)^2$$

It should be remarked that case 3 is a more general case of the preceding case.



(a) Case 1. (b) Case 2-1. (c) Case 2-2. (d) Case 3.

Figure 4.5. Merging of place memories T^m and T^n.

T^m, T^n, N, N'
if Case1 & N and N' are roots of T^m and T^n **then**
  $create\_parent(T^m, T^n)$
else if Case 2-1 & N and N' are roots of T^m and T^n **then**
  $add\_child(T^m(R), T^n)$
else if Case 2-2 & N is a hyper-terminal T^m and N' are hyper-terminals of T^n **then**
  if N and N' do not intersect **then**
    $add\_child(T^m(N), T^n(N'))$
  end if
  if N does not have terminal nodes as children **then**
    $add\_sibling(T^m(N), T^n(N'))$
  else
    $\forall P' \in T^n(N')$
    if $\exists P \in T^m(N^*)\ g(P, P') \leq \tau_r$ **then**
      Update P
    else
      $add\_sibling(T^m(N^*), T^n(P'))$
    end if
  end if
else if Case 3 & N and N' are hyper-terminals of T^m and T^n **then**
  if N does not have terminal nodes as children **then**
    $add\_sibling(T^m(N), T^n(N'))$
  else
    $\forall P' \in T^n(N')$
    if $\exists P \in T^m(N^*)\ g(P, P') \leq \tau_r$ **then**
      Update P
    else
      $add\_sibling(T^m(N^*), T^n(P'))$
    end if
  end if
end if

Figure 4.6. Merging process $T^m + T^n$.

4.5.4. Merged Place Memory

In the merging process, parts of place memories that are overlapping are determined. In case 1, since there is no overlap, the merged place memory simply contains the union of individual knowledge of the two robots. This is realized by the expanding current place memories so that the two respective place memories are incorporated as two subtrees as shown in Figure 4.5a. From the perspective of the individual robots, the resulting place memories are identical - namely

$$T^m(k^m + 1) \;\; = \;\; T^n(k^n + 1)$$

In the remaining three cases, the robot needs to determine the overlap among the two place memories. There may be many overlap areas depending on the overlap of the robots' knowledge. This is achieved via comparing all the hyper-terminals of T^n against those of T^m. Given $N \in T + n$ and $N' \in T^m$, their overlap can be checked as: overlap as:

$$\exists \, \|c_n(N) - c_m(N')\|^2 < (\rho_n(N) + \rho_m(N'))^2 \tag{4.6}$$

If an overlap is determined, the amount of overlap is computed. The goal is to find node N^* where the hyperspheres are maximally overlapping - namely:

$$N^* \in \arg max_{N' \in T^n} S(c_n(N), \rho_n(N)) \cap S(c_m(N'), \rho_m(N')) \tag{4.7}$$

Once each hyper-terminal N of T^n is matched with N^* of T^m, then the robot attempts to recognize the respective terminal nodes associated with $T^n(N)$. In case of recognition, the associated place knowledge is updated. Otherwise, it is learned as a new place via adding it as a terminal node of N^*.

The merging algorithm is summarized as given in Algorithm 4.5.3. The hyperspheres associated with the merged memories are shown to have identical descriptive

statistics [10]. However, this does not necessarily imply that $T^m = T^n$. The following proposition addresses this:

Proposition 4.1. *Let* $T^m \leftarrow T^m + T^n$ *and* $T^n \leftarrow T^n + T^m$, $T^m = T^n$ *iff* $S(c_m, p_m) = S(c_n, p_n)$ *and* $\forall N \in T^m, \exists N \in T^m S(c_m(N), p_m(N)) = S(c_n(N'), p_n(N'))$ *and* $\forall N' \in T^n, \exists N \in T^m S(c_m(N'), p_m(N')) = S(c_n(N), p_n(N))$

Proof. Suppose $T^m = T^n$ and $\exists N' \in T^n, \exists N \in T^m$ s.t. $S(c_m, p_m) \neq S(c_n, p_n)$, $\exists T^m(N) \neq T^n(N')$. This implies that $T^m \neq T^n$. Hence this contradicts with Theorem 4.1. □

4.6. Topological Map Merging

The topological mapping algorithm works in parallel with place memory matching algorithm. Suppose that robot m has communicated with robot $n \in R$, $n \neq m$ and has received its (topological) map memory as represented by T^n. From the perspective of robot m, the goal is to update its map memory as to incorporate the new knowledge as

$$G^m(k^m + 1) \leftarrow G^m(k^m) + G^n(k^m)$$

Again, there are two requirements for the merging process:

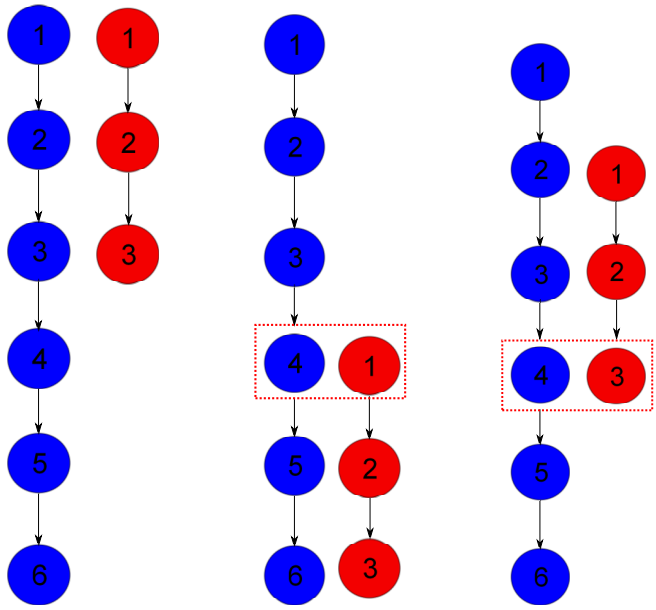
- First, the resulting map memory $G^m(k^m + 1)$ should incorporate the learned places $\mathcal{P}^m \cup \mathcal{P}^n$ of both of the robots.
- Second, $G^m(k^m + 1)$ should contain the learned spatial relations of both of the robots $E^m \cup E^n$.

The merging process is based on the nature of merging in the place memory. If a place $P' \in \mathcal{P}^n$ is recognized to be also in $\mathcal{P}^m$, no node addition is done. The recognition algorithm uses the place memory $T^m(k_m)$ and is as described in Section 4.3.1. Otherwise, a new node is added to G^m. The associated edges are checked to exist in $E^m(k_m)$

and added if necessary. The merging process is summarized in Algorithm 4.6.

Map memories G^m and G^n
if Case1 or Case2.1 **then**
  $G^m = G^m \cup G^n$
else if Case2.2 or Case 3 **then**
  $G^m = G^m \cup G^n$
  if $\forall P' \in T^n(N')\ \exists P \in T^m(N^*)\ g(P, P') \leq \tau_r$ **then**
    $P = P \cup P'$ {Merge corresponding nodes}
  end if
end if

Figure 4.7. Merging map memories G^m and G^n.



(a) Case 1 & 2.1. (b) Case 2-2. (c) Case 3.

Figure 4.8. Merging of topological maps G^m and G^n.

The topological maps of T^n and T^m which are in red and blue colors respectively are shown in Figure 4.8. for Cases 1 & 2.1, since there is no overlap, the topological maps are left as it is as shown in Figure 4.8a. For case 2.2, we see that there is a place merging between places 1 & 4 which is shown in Figure 4.5c. Thus, the corresponding

nodes of the topological maps are merged as shown in Figure 4.8b. For case 3, a place merging between places 3 & 4 which is shown in Figure 4.5d. Therefore, the corresponding nodes of the topological map are merged as shown in Figure 4.8c.

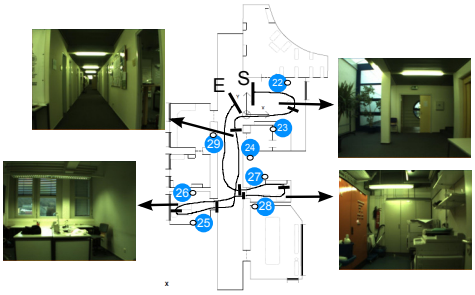
4.7. Complexity Analysis

The complexity analysis of the overall approach is conducted. It is assumed that each robot has adequate processing and bandwidth capabilities. For transmitting data before memory merging, the algorithm's complexity is calculated as $\mathcal{O}(N_I \bar{p^*})$ where N_I is the size of the bubble descriptor vector while $\bar{p^*}$ which is defined in Equation 4.8 is the maximum number of learned places among the robot team. The complexity of the place memory merging algorithm is calculated as $\mathcal{O}(\bar{p^*}^2)$ while merging map memories has a complexity of $\mathcal{O}(\bar{p^*})$. Since the number of learned places is very small compared to the number of visited base points, the complexity of the approach is feasible compared to appearance-based SLAM approaches such as FAB-MAP [68].

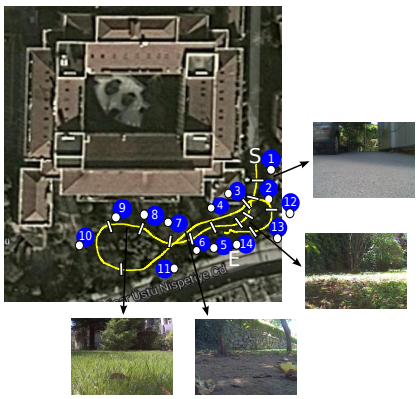
$$\bar{p^*} = \arg\max_{p^*}(|P^m|,\ m \in R) \tag{4.8}$$

4.8. Experiments

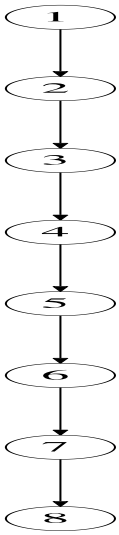
We conduct a series of experiments using the data collected by a team of robots as well as the COLD data set [95]. Each robot has only visual data from a sequence of base points along a given path and is not given any other sensory data. At each base point, the robot encodes the incoming visual data by a descriptor having dimension $N_I = 600$. The details of this descriptor are as explained in [6]. We consider each merging case separately and investigate the structure and performance of the merged memories. We also compare the proposed approach with straightforward one-by-one learning of places after either the robot visiting them or receiving their knowledge from other robots.



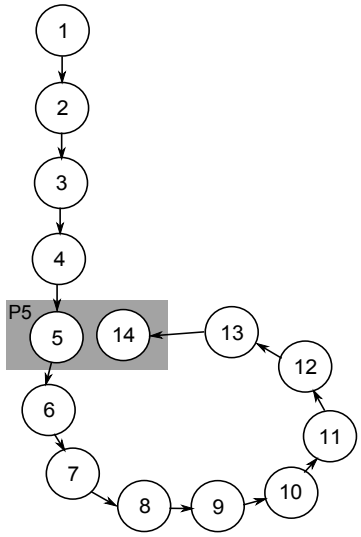
(a) Robot 1 learns Sa site.



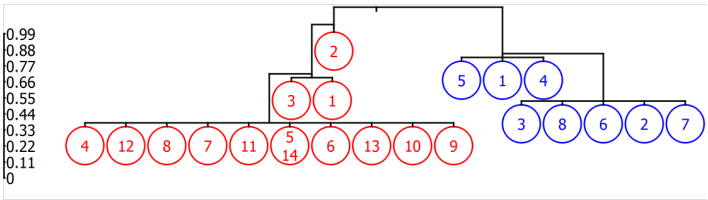
(b) Robot 2 learns campus site.



(c) Map memory of Robot 1.



(d) Map memory of Robot 2.



(e) Merged place memories $T^1 = T^1 + T^2$ and $T^2 = T^2 + T^1$.

Figure 4.9. Case 1: Learned places nonoverlapping. Blue nodes indicate places from robot 1's place memory while red nodes indicate those from robot 2's place memory.

4.8.1. Case 1: Learned Places Nonoverlapping

In this case, two robots visit two different sites without any overlap of the places that are learned. Of course, the robots do not know this. The first robot visits the Saarbrucken (Sa) site from the COLD data set, collects data at 997 base points in a

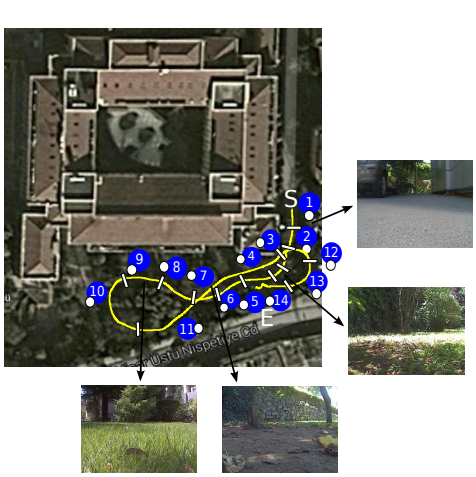
50 meters tour and learns all of the 8 places based on this data. The place memory T^1 has radius $\rho_1 = 0.607$. The map memory is as shown in Figure 4.9a. The second robot navigates along a 175 meters path, collects data at 3200 base points and detects 14 places. Since the second robot returns to where it is started at the end of the tour, the last detected place is recognized as 5^{th} place which is correct. Thus, there are 13 learned places in its place memory in the place memory T^2. The associated hypersphere has radius $\rho_2 = 0.478$. The resulting map memory is as shown in Figure 4.9b. The squared Euclidean distance between the centroids of two respective hyperspheres $\|c_1 - c_2\|^2 = 1.6118$ is larger than the squared sum of radii $(\rho_1 + \rho_2)^2 = 1.1772$ which satisfies the condition for Case 1. The memories $T^1 \leftarrow T^1 + T^2$ and $T^2 \leftarrow T^2 + T^1$ resulting from their merging are identical and are as given in Figure 4.9e. It is observed that the places of both sites are separated from the top root node. The right subtree corresponds to the place memory of the first robot prior to merging while the left subtree corresponds to that of the second robot again prior to merging. The merged topological map is simply the union of individual maps as given in Figure 4.9c and Figure 4.9d.

Table 4.1. Correspondence between places $\mathcal{P}^X$ and $\mathcal{P}^Y$.

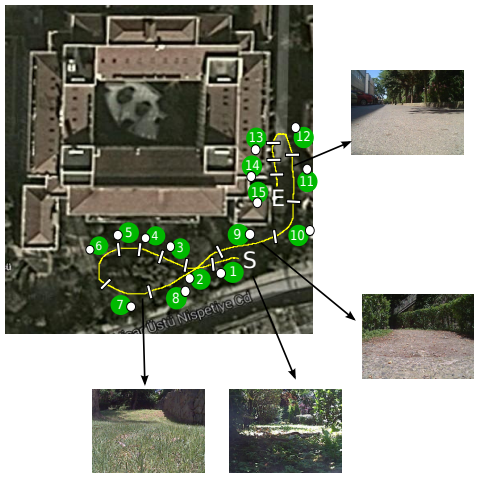
$\mathcal{P}^X$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Manual $\mathcal{P}^Y$	15	15	3,4	3,4	1	2	3	4,5	4,5	6,7	8,9	9	9	1
Approach $\mathcal{P}^Y$	-	-	-	4	1	2	2	4	7	7	4	4	2	1

4.8.2. Case 2: Learned Places Mostly Overlapping

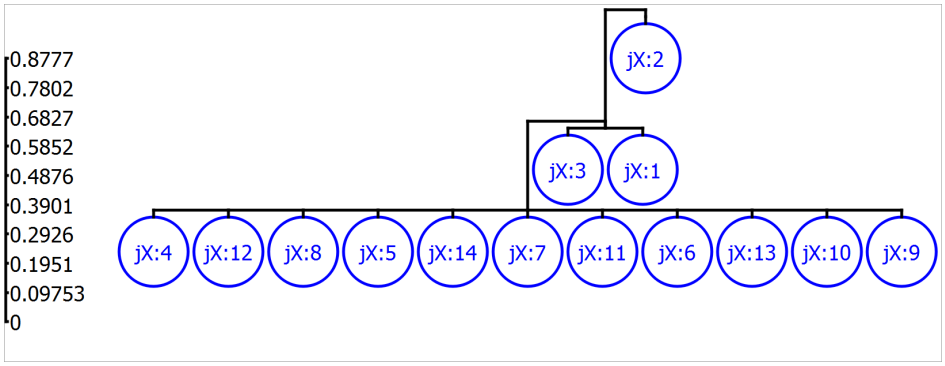
Next, we consider case 2 when the learned places are mostly overlapping, but one of robots covers a larger area. This case is experimented with a team of two robots, robot jX and robot jY. The robots navigate in the university campus in a teleoperated manner. The first robot jX moves along a path of approximately 175 meters, collects visual data at 3200 base points and detects 14 places as shown in Figure 4.10. The resulting place memory T^X of robot jX is as shown in Figure 4.10c with $\rho_X = 0.478$. The second robot jY navigates along a longer path similar to robot jX's, but also covering some additional places. As such, the second robot jY not only



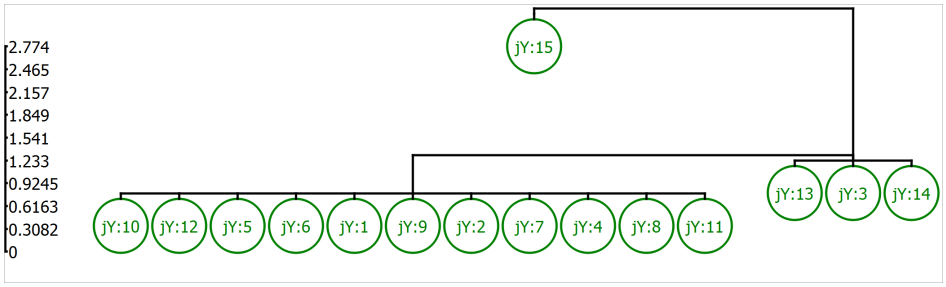
(a) Tour of robot jX .



(b) Tour of Robot jY .



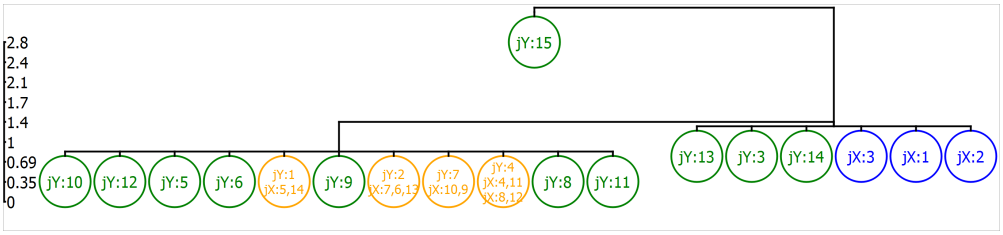
(c) Place memory of robot jX .



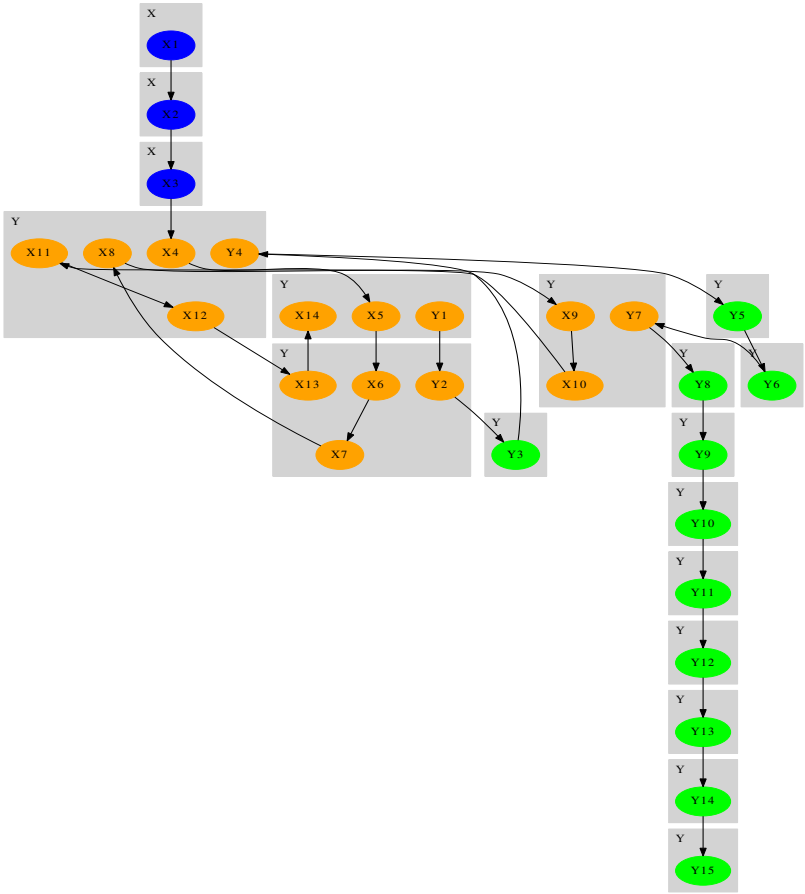
(d) Place memory of robot jY.

Figure 4.10. Case 2: Learned places mostly overlapping: Tour of robot jX and jY.

visits all the places visited by robot jX, but goes to some new places. Altogether, it collects data at 2550 base points and detects 15 places. Note that as the robot acquires its visual data using a perspective camera, its field of view is limited. Thus, it may see completely different appearances if it is going through the same base point, but in opposite directions. For example, robot jX learns place 13 while moving down while



(a) Merged place memory $T^Y \leftarrow T^Y + T^X$.



(b) Merged map memory $G^Y \leftarrow G^Y + G^X$.

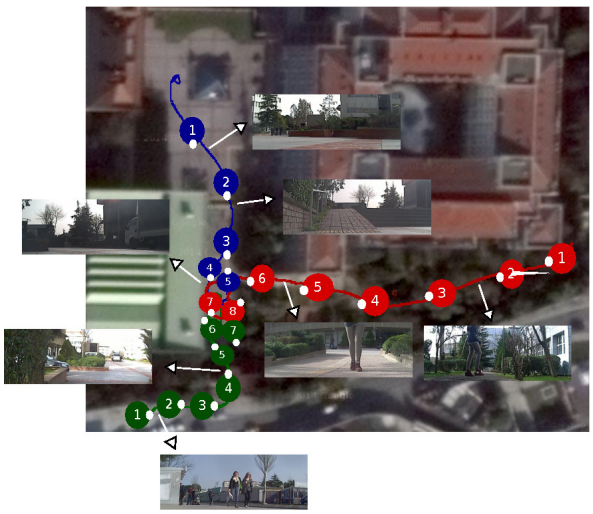
Figure 4.11. Case 2: Learned places mostly overlapping: Merged memory.

robot jY learns place 10 while moving up. While these two places geographically are the same, as their appearances are different, they will be perceived as two different places by each of the robots. Its place memory T^Y contains 15 places with radius $\rho_Y = 0.8445$. The squared Euclidean distance between the centroids of two trees $\|c_X - c_Y\|^2 = 0.2150$ is smaller than ρ_Y. As such, this is an example for Case 2. From the perspective of robot jY, the resulting spatial memory is as shown in Figure 4.10.

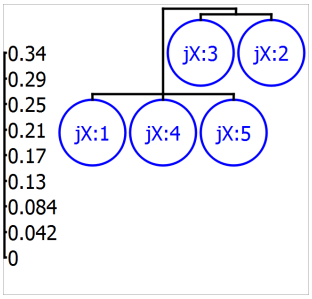
The actual correspondences between the learned places $\mathcal{P}^X$ of robot jX and and $\mathcal{P}^Y$ of robot jY based purely on appearance are manually determined to be as in Table 4.1 via visual inspection. Note that while most of the appearance-wise corresponding places are actually at the same geometric locations, there are also few exceptions where this doesn't hold. For example, while places jX:3 and jX:4 of the robot jX appear similar to jY:3 and jY:4 of robot jY, they actually are different at physical locations. On the other hand, places jY:10, jY:11, jY:12, jY:13 and jY:14 that are visited only by robot jY do not have any appearance-wise corresponding places in $\mathcal{P}^X$. As such, we expect the merged memories to contain about 15 places. The merged place memory $T^Y \leftarrow T^Y + T^X$ of robot jY is as shown in Figure 4.11b with recognition parameter $\tau_r = 2$. It contains 18 places. Note that this is the same for robot jX since its place memory is updated as $T^X \leftarrow T^Y + T^X$. It is observed that six of these places are correctly merged. Some of them are merged with nearby places whose appearances are very similar. As such, these mergings are also acceptable. Examples include jX:7 and jX:9. The robot wrongly views places jX:1,jX:2 and jX:3 as new places while this is not case. When these places are observed in detail, it is seen that because of view point differences between two paths, these places are seen as new. On other hand, it confuses places jX:12 and jX:13 to be the same as jY:4 and jY:2 respectively. These confusions are inevitable since the views of the garden area are very similar because of vegetation and trees. The resulting topological map $G^Y \rightarrow G^Y + G^X$ is as shown Figure 4.11b. The topological map merging is executed in parallel to place memory. We see that the combined nodes jX:11 and jY:4 is wrongly merged.

4.8.3. Case 3: Learned Places Partially Overlapping

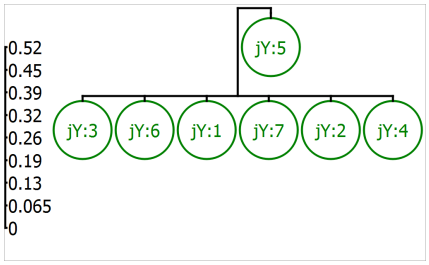
Next, we consider the case when robots visit places that are only partially overlapping. The experiment is conducted with a team of three robots - jX, jY and jZ in outdoors settings. Robots start at different parts of the campus and converge to a final destination at the end of their respective tours as shown in Figure 4.12a. Each collects images with its perspective camera along its respective path with sample images as shown. Robot jX traverses a path of approximately 150 meters, collects data at 1050



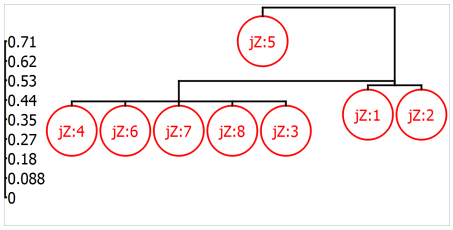
(a) Robot jX blue path, jY green path, jZ red path.



(b) T^X.



(c) T^Y.



(d) T^Z.

Figure 4.12. Case 3: Learned places are partially overlapping: robots' paths and place memories.

base points and learns 5 places. The resulting place memory T^X is as shown in Figure 4.12b. It is seen that jX has a memory structure composed of 2 levels. The places 2 & 3 are in one level while places 1,4 & 5 are on the other. Robot jY traverses a path of 80 meters, collects visual data at 370 base points and learns 7 places. Its place memory has structure T^Y as shown in Figure 4.12c consisting of two levels, where place 5 is

$\mathcal{P}^X$	1	2	3	4	5
Manual $\mathcal{P}^Y$	-	4	4	7	4,5
Manual $\mathcal{P}^Z$	-	4,5	4,5	7	8

(a) $\mathcal{P}^X$ to $\mathcal{P}^Y$ and $\mathcal{P}^Z$.

$\mathcal{P}^Y$	1	2	3	4	5	6	7
Manual $\mathcal{P}^X$	-	-	-	2,3	5	5	3,4
Manual $\mathcal{P}^Z$	-	-	-	4	-	8	7

(b) $\mathcal{P}^Y$ to $\mathcal{P}^X$ and $\mathcal{P}^Z$.

$\mathcal{P}^Z$	1	2	3	4	5	6	7	8
Manual $\mathcal{P}^X$	-	-	-	2	-	-	4	5
Manual $\mathcal{P}^Y$	-	-	-	4	-	-	7	6

(c) $\mathcal{P}^Z$ to $\mathcal{P}^X$ and $\mathcal{P}^Y$.

Figure 4.13. Case 3: Learned places are partially overlapping: place correspondences.

at the first level and the remaining places are the second level. Here place 5 serves as a transition region between the street and the campus entrance. Thus, it is placed further away from the other places. The remaining places are on the same level probably due to similarity of their appearances. Finally, robot jZ traverses a path of 130 meters, collects visual data at 710 base points and learns 8 places. Its place memory has structure T^Z as shown in Figure 4.12d. T^Z has a more complex structure with 3 levels. This can attributed to the fact that environmental changes along the robot jZ's path are more than those of robots jX and jY. The appearance-wise correspondences between robots' learned places are manually determined to be as given in Table 4.13a, Table 4.13b and Table 4.13c. It is observed that appearances associated with places jX:4, jY:7 and jZ:7 are very similar. They also geometrically correspond to the same area. This is also the case for jX:5 and jZ:8. On the other hand, while some places are geometrically nearby, their appearances are different as the respective robots move through them in opposite directions. For example, places jY:6 and jZ:7 are in this category. Similarly, this holds for jX:5 and jZ:6 as well as jZ:4, jX:2 and jY:4. As such, we expect them to be learned as different places. Finally, there are also places whose appearances are similar while they are geometrically distant such as jX:1 and jY:6.

Table 4.2. Descriptive statistics of hyperspheres S^X, S^Y and S^Z.

(a) Pairwise distances.

Robot	jX	jY	jZ
jX	0	0.367	0.22
jY	0.367	0	0.218
jZ	0.22	0.218	0

(b) Radii.

Robot	ρ
jX	0.244
jY	0.307
jZ	0.415

Pairwise distances between the centroids of the respective hyperspheres and their radii are given in Table 4.2(a) and Table 4.2(b). These are used to verify that that conditions for Case 3 prevail. The merged place memories vary depending on the order of merging as seen in For each robot, there are two alternatives in regards to the order of merging. It is observed that the merged place memories are different for each robot as seen in Figures 4.14, 4.15, 4.16 - even if the descriptive statistics of the associated hyperspheres are independent of the learning order. This is expected by Prop. 4.1 since those associated with the inner nodes vary. Places that are only learned by one robot are indicated by corresponding color (blue - robot jX, green -robot jY and red -robot jZ) while places shown by orange nodes indicate merging of knowledge regarding places that are determined to be overlapping.

Table 4.3. Case 3 - Place mergings.

(a) $T^X + T^Y + T^Z$.

Updated Places	jX:5	jX:1	jX:4	jZ:2
Updated With	jZ:4, jY:4	jZ:6	jZ:7	jY:6

(b) $T^X + T^Z + T^Y$.

Updated Places	jX:5	jX:1	jX:4	jZ:2
Updated With	jZ:4, jY:4, jZ:8	jZ:6	jZ:7	jY:6

(c) $T^Y + T^X + T^Z$.

Updated Places	jY:4	jY:7	jY:2	jY:1	jY:6
Updated With	jX:5, jZ:7	jX:3, jZ:3	jZ:4	jZ:6	jZ:2

(d) $T^Y + T^Z + T^X$.

Updated Places	jY:4	jY:6	jY:7
Updated With	jZ:4, jX:2	jZ:2, jX:1, jX:5	jX:4, jX:3

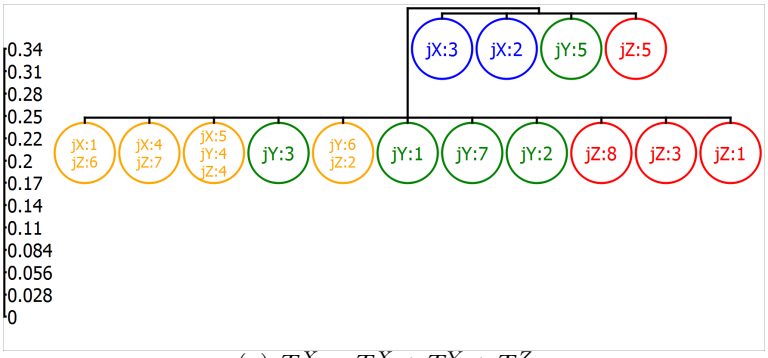
(e) $T^Z + T^X + T^Y$.

Updated Places	jZ:6	jZ:7	jZ:3	jZ:4	jZ:2
Updated With	jX:1, jX:5	jX:4	jX:3	jX:2	jY:6, jY:2

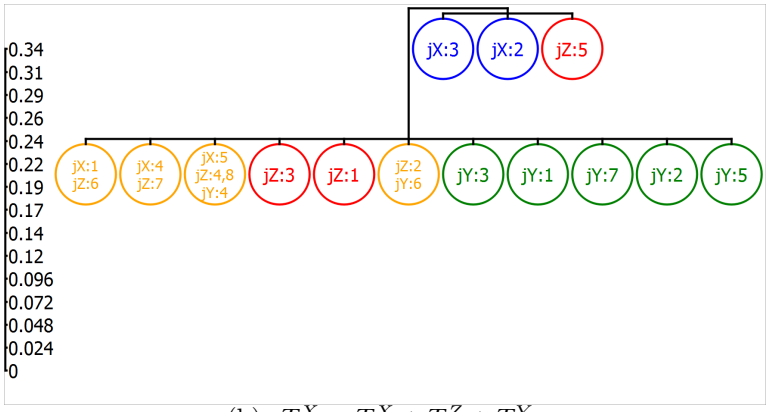
(f) $T^Z + T^Y + T^X$.

Updated Places	jZ:8	jZ:4	jZ:6	jZ:7
Updated With	jY:3, jY:1, jY:2	jY:4, jX:2	jX:1, jX:5	jX:4, jX:3

For robot jX, with the merged place memory $T^X + T^Y + T^Z$, the robot increases its place knowledge from 5 to 15 places. Note that considering Table 4.13a, we expect this number to be around 13. Closer inspection reveals that while we expect jZ:5 to be merged with jX:2 or jX:3, this is not the case. Similarly, we expect jY:7 to be merged with jX:4. Again they are determined as different places. In addition, while some mergings occur as expected, that is not the case with others as analyzed in Table 4.3(a). For example, the merging of jX:4 with jZ:7 is expected. This is also geometrically correct. On the other hand, while places jX:1 and jZ:6 are appearance-



(a) $T^X = T^X + T^Y + T^Z$.

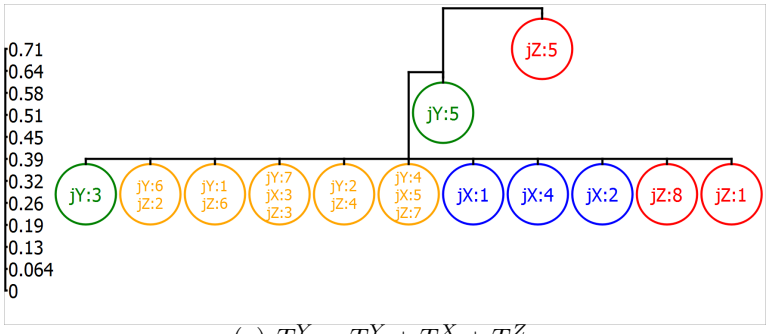


(b) $T^X = T^X + T^Z + T^Y$.

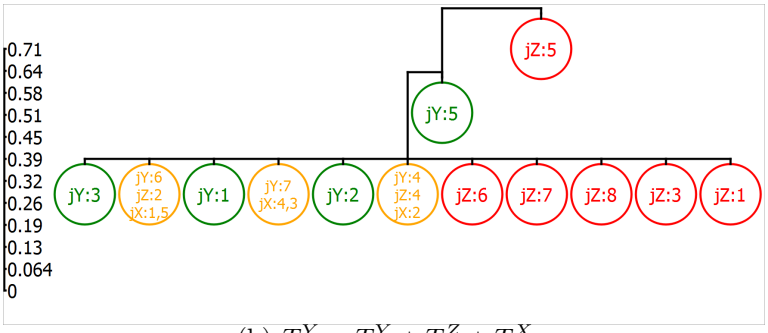
Figure 4.14. Merged place memories of jX.

wise not similar, they are viewed as one place nevertheless and the associated place knowledge is updated accordingly. In summary, two of the places are correctly merged while the remaining are not. When the order of learning is changed, the robot expands its place from 5 to 14 places. Again, while jY:7 is expected to merge with jX:4, this is not the case. Thus the place memory contains one additional place. The knowledge associated with 3 places are correctly updated as seen in Table 4.3(b). For example, jX:y is correctly merged with jX:4 - similar to the previous case. On the other hand, there are also some wrong merging as also seen in the same table. For example, jX:1 and jZ:6 are again wrongly merged. Closer inspection reveals that wrong mergings tend to occur across places that while appearance-wise different nevertheless contain similar entities such as sky, building and walkway.

In case of robot jY, its merged memory $T^Y + T^X + T^Z$ contains 13 places that are arranged 3 levels. This is as expected. The merging of place knowledge is given in



(a) $T^Y = T^Y + T^X + T^Z$.

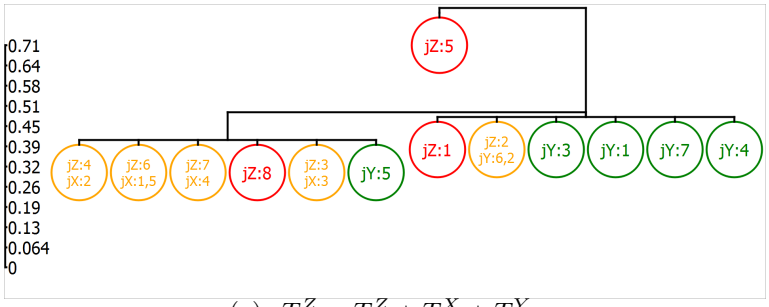


(b) $T^Y = T^Y + T^Z + T^X$.

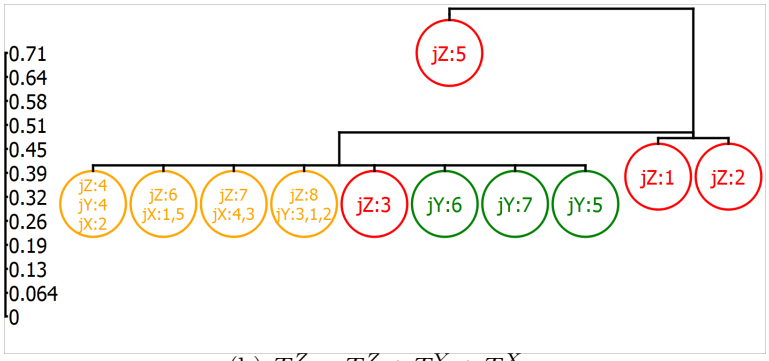
Figure 4.15. Merged place memories of jY.

Table 4.3(c). In this case, the robot tends to confuse most of the places and wrongly merge them. For example, jZ:2 and jY:6 are viewed as being same places. When the learning order is reversed to $T^Y + T^Z + T^X$, again the robot increases its knowledge of places from 7 to 13. The resulting tree structure is in 3 levels. Interestingly, merging performance is considerably much better. The robot correctly merged 5 of the seven places. For example, the knowledge of place jY:4 is updated with that of jZ:4 and jX:2. These places are all similar since they both view the front of library building from a distance.

Finally for robot jZ, its merged place memory $T^Z + T^X + T^Y$ contains 13 places - again arranged in 3 levels. As such, its knowledge of places expands from 8 to 13. It finds the knowledge associated with 7 places as overlapping with its own. Two of these are correct while the remaining are incorrectly merged. For example, while places jZ:4 and jX:2 are correctly found to be overlapping, this is not the case for places jZ:6 and jX:1 or jX:5. With the reversed merging order $T^Z + T^Y + T^X$, the robots expands its



(a) $T^Z = T^Z + T^X + T^Y$.



(b) $T^Z = T^Z + T^Y + T^X$.

Figure 4.16. Merged place memories of jZ.

knowledge of places from 8 to 11. Since we expect this number to be 13, some of the places are wrongly merged. In this case, there are 3 places that are correctly merged while 6 places are incorrectly merged. Again, it is observed wrong mergings occur due to similarity of scenes.

The merged map memories are given in Figures 4.17 4.18 and 4.19 . As they use the merged place memories, their structures reflect the merging of knowledge associated with places that are found to be overlapping. The number of nodes in each merged map is equal to the number of places in the place memory. The spatial relations among different places are shown by the edges. Again, places that are only learned by one robot are indicated by corresponding color (blue - robot jX, green -robot jY and red -robot jZ) while places shown by orange nodes indicate merging of knowledge regarding places that are determined to be overlapping. As some mergings are correct while some are not, some edges are geometrically correct while others are not. For example, with robot jX, 20 of the 35 spatial relations are geometrically correct in the merged map

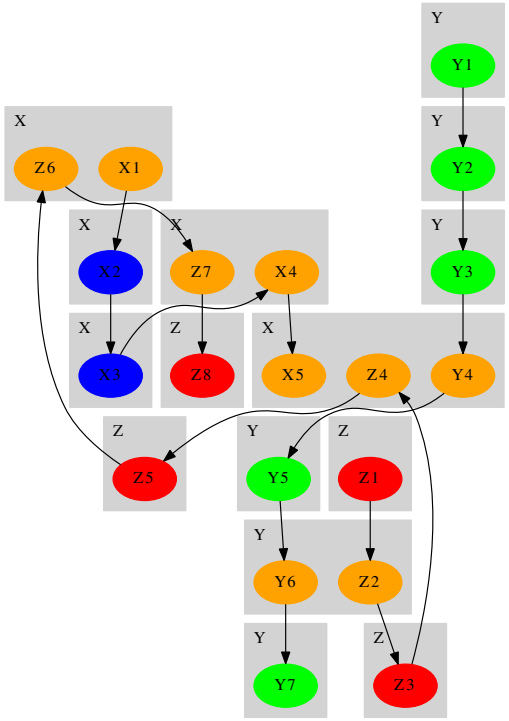
Table 4.4. Overall descriptive statistics for the merged place memories.

Place Memories	Th NormSq	Ac NormSq	Th ρ	Ac ρ
$T^X + T^Y + T^Z$	245.84	245.84	0.623	0.622
$T^X + T^Z + T^Y$	245.84	245.84	0.623	0.623
$T^Y + T^X + T^Z$	245.84	245.84	0.623	0.622
$T^Y + T^Z + T^X$	245.84	245.84	0.623	0.623
$T^Z + T^X + T^Y$	245.84	245.84	0.623	0.622
$T^Z + T^Y + T^X$	245.84	245.84	0.623	0.623

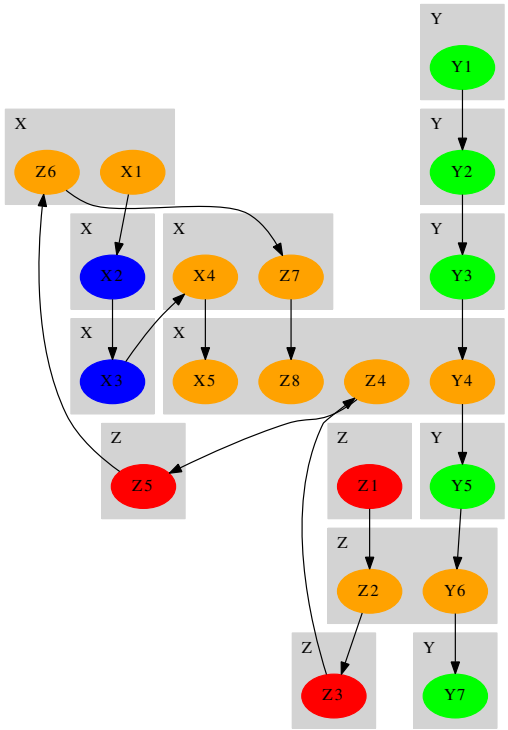
memory $\bar{g}^X + \bar{g}^Y + \bar{g}^Z$. It is observed that the robot can successfully navigate from place $jX : 4$ directly to place $jZ : 8$. On the other hand, as places $jX : 1$ and $jZ : 6$ are incorrectly viewed as overlapping, the robot trying to navigate directly from $jX : 1$ to $jZ : 5$ will end unsuccessfully. On the other hand, with $\bar{g}^X + \bar{g}^Z + \bar{g}^Y$, there are 14 places with the knowledge associated with three places being revised based on merged knowledge. As such, 39 spatial relations are inferred. 20 of these are geometrically correct while 19 are not - due to incorrectly merged places. For robot jY, the merged map contains $\bar{g}^Y + \bar{g}^X + \bar{g}^Z$ contains 13 places with 38 spatial relations inferred. In this case, 16 of these are geometrically correct while the rest are not. In the merged map memory $\bar{g}^Z + \bar{g}^Y + \bar{g}^X$ of robot jZ, there are 11 places with 31 spatial relations inferred. 17 of these are geometrically correct while the remaining 14 are not. In summary, the resulting merged maps are correct with around 50-60% reliability. This is quite good considering only appearances are used in the merging process.

4.8.4. Recognition Performance After Merging

After merging spatial memories, we conduct a series of experiments that aim to assess whether the merged knowledge can enable the robots to recognize the respective places. In these experiments, each of the robots is made to follow two alternative paths as shown in Figure 4.20. The places along these paths have been learned by either the robot itself or through merging of knowledge. Furthermore, there is some

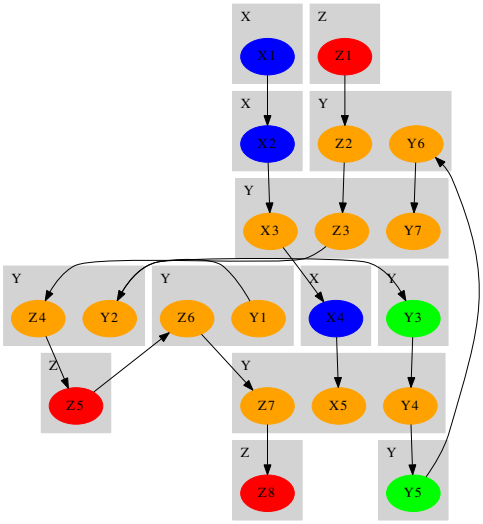


(a) $\bar{g}^X = \bar{g}^X + \bar{g}^Y + \bar{g}^Z$.

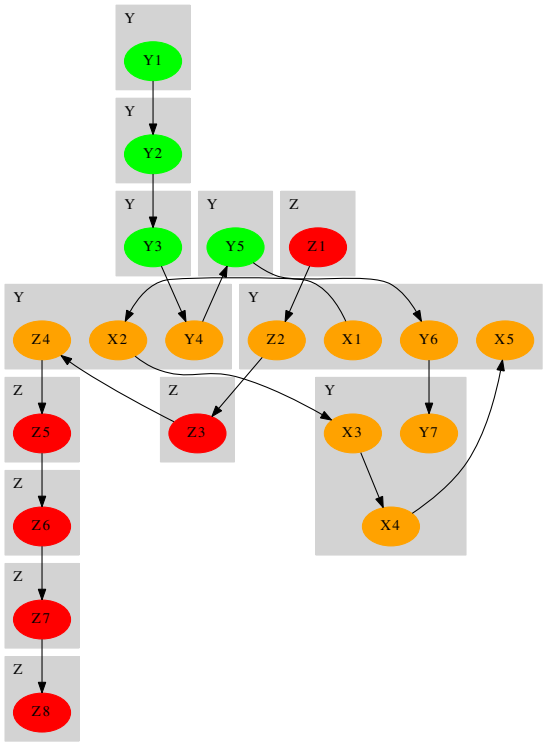


(b) $\bar{g}^X = \bar{g}^X + \bar{g}^Z + \bar{g}^Y$.

Figure 4.17. Merged map memories of jX.

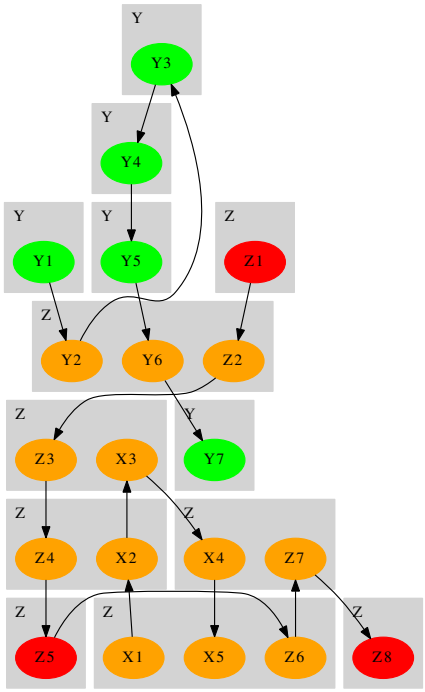


(a) $\bar{g}^Y = \bar{g}^Y + \bar{g}^X + \bar{g}^Z$.

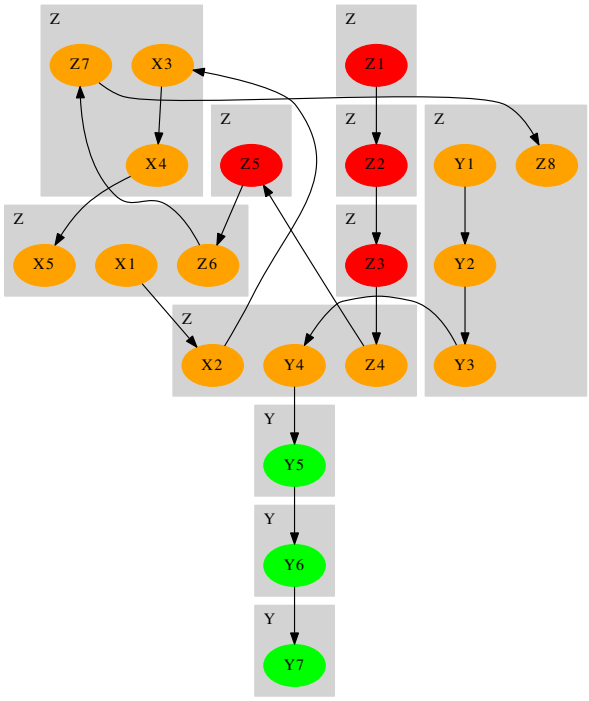


(b) $\bar{g}^Y = \bar{g}^Y + \bar{g}^Z + \bar{g}^X$.

Figure 4.18. Merged map memories of jY.



(a) $\bar{g}^Z = \bar{g}^Z + \bar{g}^X + \bar{g}^Y$.



(b) $\bar{g}^Z = \bar{g}^Z + \bar{g}^Y + \bar{g}^X$.

Figure 4.19. Merged map memories of jZ.

changes in appearance as they are done about 2 months after so that there are some changes in the outdoors appearances due to seasonal changes. As such, we are able to test the recognition performance of the merged place memories - even under changing environmental conditions.

The first path shown in blue is about 100 meters and goes through two places that have been learned by robot jX in previous visit. Robots jY and jZ have learned these places via knowledge sharing. As the scenes associated with each of the places are mostly composed of buildings, appearances are not much affected by seasonal changes. The precision-recall results with $\tau_r = 1.5$ for each robot and merged memory are given in Table 4.5. It is observed that all the robots can perfectly recognize these places except robot jY with the merged memory $T^Y + T^Z + T^X$. In this case, it recognizes one of the places as a new place. As robot jY has learned this place through merging, apparently its knowledge is not sufficient in information content.

The second path shown in red goes through seven places (containing car park, garden and a concrete trail) that have been learned by robot jZ. Robots jX and jY have learned these places through knowledge merging. It is observed that this path is rather challenging compared to the first path because of the significant seasonal changes in the appearance of the garden area. Robot jZ has the highest recognition performance with the merged place memory $T^Z + T^Y + T^X$ memory. It has 80% recall at 25% precision. Interestingly, with merging order changed to $T^Z + T^X + T^Y$, performance changes to 66% recognition at 33% precision. Interestingly, for robot jX which as learned these places based on the knowledge of robot jZ, performance is at the same level with again 66% recognition at 33% precision. The merged memories of Robot jY enable 40% recall at 50% precision. Despite the fact that there are seasonal changes in the appearances of these places, the robots can still recognize them using their merged place memory with acceptable accuracy - even if they have been learned through merging of their knowledge with those of other robots.

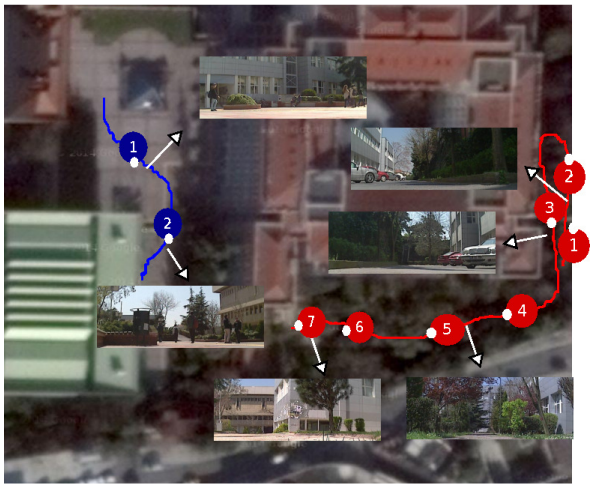


Figure 4.20. Navigating in places that have been learned through merging of spatial memories after 2 months.

4.8.5. Comparative Performance

Finally, we compare performance of knowledge acquired through merging via knowledge that is learned one-by-one as presented in [14]. In these experiments, we consider robots jX and jZ separately and continue using recognition parameter $\tau_r = 1.5$. After one-by-one learning, the place memory T^X of robot jX contains 8 distinct places while that of robot jZ contains 11 places. Further analysis as presented in Table 4.6 reveals that both robots confuses some of the new places with those already learned. For example, robot jX wrongly recognize places jY:1 and jY:2 as place jX:3. Similarly, robot jZ recognizes many of the places from robot jY with jZ:2. Furthermore the places that are confused are different from those of knowledge merging. For example, while robot jX recognizes places jZ:4 and jY:4 as place jX:5 or jZ:6, as place jX:1 or place jZ:7 as place jX:4 after merging, places jX:3, jY:1 and jY:2 are confused after one-by-one learning. The update of jX:3 with jY:1, jY:2 and jY:7 is incorrect in the sense that jY:7 and jX:3 are geometrically nearby although their viewing angles are different while jY:1 and jY:2 are geometrically away. This confusion is attributed to the fact that all contain common entities such as sky, people, walkway. Similarly jX:2 is updated with jZ:4, jZ:5, jZ:6 and 7. Here although there is a translational difference between jX:2 and other merged places, they all see the front courtyard of the library building. As

Table 4.5. Recognition performance of merged place memories with $\tau = 1.5$.

Place Memory	Blue Path		Red Path	
	R (%)	P (%)	R (%)	P (%)
$T^X + T^Y + T^Z$	100	100	60	33
$T^X + T^Z + T^Y$	100	100	60	33
$T^Y + T^X + T^Z$	100	100	40	50
$T^Y + T^Z + T^X$	50	100	40	50
$T^Z + T^X + T^Y$	100	100	60	33
$T^Z + T^Y + T^X$	100	100	80	25

Table 4.6. Case 3 - Obtained place updates, one-by-one learning.

(a) T^X after one-by-one learning.

Places	jX:3	jX:2	jY:2
Updated with	jY:1,3,7	jZ:4,5,6,7	jY:4, jZ:1,2,3,8

(b) T^Z after one-by-one learning.

Places	jZ:2	jY:6
Updated With	jY:2,3,4,7 jX:1,2,4,5	jX:3

a result appearance-wise these places can be easily confused. Finally jY:2 is updated with jY:4 and jZ:1, jZ:2,jZ:3 and jZ:8. There is a very small translational difference between jY:2 and jY:4. This merging is acceptable. On the other hand places jZ:1, jZ:2, jZ:3 and jZ:8 are geometrically away from jY:2 and jY:4. Here again these places share some common entities such as library building, sky and concrete surface. As a result these places are confused and incorrectly merged.

With robot jZ, after one-by-one learning, there were 9 place updates which is the same as $T^Z + T^Y + T^X$. For $T^Z + T^Y + T^X$, places jZ:8, jZ:4 jZ:6 and jZ:7 were updated while for jZ one-by-one jZ:2, jY:6 were updated. Interestingly, jZ:2 is updated with places jY:2, jY:3, jY:4 and jY:7 and places jX:1, jX:2,jX:4 and jX:5. Here jZ:2 share common entities with jY:2 although they are geometrically apart. Thus, they are merged incorrectly. Places jY:3 and jY:4 are geometrically very close to place jY:2. Thus they are also merged into jZ:2. Here places jY:2, jY:3 and jY:4 and jY:7 are

Table 4.7. Comparative performance.

Robot	Place Memory	Processing Time (msecs)	Blue Path		Red Path	
			R (%)	P (%)	R (%)	P (%)
jX	$T^X + T^Y + T^Z$	80	100	100	66	33
	One-by-one	1145	100	100	80	25
jZ	$T^Z + T^Y + T^X$	85	100	100	80	25
	One-by-one	450	0	0	66	33

geometrically away but again all share common entities such as buildings and concrete surface so they are incorrectly merged. When places of jX are introduced, they have very similar views with jY:7. As a result they are incorrectly merged into the same node with them. Next, jY:6 is updated with jX:3. Here jY:6 sees library building, concrete walkway and some trees while place jZ:3 also contains a view of the library building, trees and a walkway but from a different geometrical location and view point. As a result these places are incorrectly merged because of appearance-wise similarity in between.

The comparison is done with respect to processing time and recognition performance as given in Table 4.7. Interestingly, learning through knowledge merging takes considerably shorter time as compared to one-by-one learning. It takes about 85 msec in contrast to 450-1145 msec for one-by-one learning. This is attributed to the processing of knowledge as a whole or in portions. The recognition performance of robot jX does not seem to be affected by how knowledge is learned. On the other hand, this does not seem to be the case for robot jZ. Its performance with one-by-one learning is considerably poorer. This is probably due to it not recognizing a lot of the places. The results also show that, for small and appearance-wise similar experimental environments, knowledge merging has a clear advantage over one-by-one learning such that false alarms are decreased. This is attributed to the handling the knowledge as a whole or in portions which implies that knowledge structure is preserved to some extent - in contrast to one-by-one learning where this does not hold.

4.8.6. Summary

In summary, we propose an approach that can be used by the robots to merge their spatial knowledge - regardless of their overlap or geographic scale. The resulting merged spatial memories are very compact in the sense that they represent data from thousands of base points with just a few number of nodes. As such, the approach can be easily extendible for large-scale and long-term operation. Furthermore, the robots can use the merged knowledge to recognize places that are learned only through this merging. As other robot's knowledge is processed as a whole or in portions, the resulting merged memories enable better performance as compared to one-by-one learning.

4.9. Conclusion

In this chapter, the problem of merging appearance-based spatial knowledge of multiple robots operating independently. The goal is to have robots that can expand their spatial knowledge with that of other robots [7]. As such, the robots will be able to reason about places that are learned through knowledge merging. Here, we present a novel approach for merging of appearance-based spatial knowledge that is comprised of two separate, but related parts - namely learned places and their spatial relations. The former is retained in the robot's place memory [14] while the latter is encoded in the (topological) map memory. It is assumed that each robot has its individual spatial knowledge. In the merging process, each robot communicates with each of the remaining robots one-by-one and receives the spatial knowledge of the other robot. Once this is over, it first merges its place memory with that of the other robot. It then incorporates other robot's map memory into its own via adding the new spatial relations as appropriately.

The advantages of such an approach are three-fold: First, it scales easily with respect to the amount and overlap of the appearance data. This is because it does not require matching pairs of appearances from two different maps. Furthermore, the expanded place knowledge continues to be organized as a semantic hierarchy that is amenable to human-like interpretations and higher-level symbolic reasoning. Finally,

it can be applied in a decentralized manner by all the robots individually.

This work can be extended in many ways to generate hierarchical navigational paths via incorporating the merged knowledge. Furthermore, the robots can reason about their recognition results and compare their own recognition performance with others to improve the overall recognition.

5. CONCLUSION

This thesis is concerned with autonomous multi-robot topological spatial cognition problem. The first step of the proposed approach is based on the problem of place representation and recognition using multi-sensory data which is explained in Chapter 2. It is shown that bubble space representation can easily be used to combine RGB and depth data while affording acceptable recognition performance even with limited sensing capabilities and simple features. The advantage of RGB-D sensing in bubble space is due to the fact that the associated feature vectors encode both sensory observations and their relative S^2 geometry. Hence, the rich 3D spatial information contained within the 3D depth data is not lost while allowing easy integration of data from different sensing modalities. For place learning and recognition, we have used a standard supervised learning approach - support vector machines (SVM) in conjunction with bubble descriptors.

Next, an autonomous real-time approach is developed for solving the single-robot spatial cognition problem in Chapter 3. Spatial cognition is concerned with the acquisition, organization, utilization and revision of knowledge about spatial environments [3–5]. The proposed model which is explained in Chapter 3, incorporates a long-term spatial memory where two separate, but related types of knowledge are stored. The place memory organizes the learned places in a hierarchy based on their appearance-related similarities while the topological map simply encodes their spatial relations. It also provides the framework to which the robot is able to link new knowledge by association. The processing modules operate together so that the robot builds its spatial memory or utilizes it in an organized, incremental and unsupervised manner.

Finally, the problem of merging appearance-based spatial knowledge of multiple robots operating independently is considered in Chapter 4. The problem is considered as merging of appearance-based spatial knowledge that is comprised of two separate, but related parts - namely learned places and their spatial relations. There are two aspects that differ from previous related work on map merging as such a ‘place’ is

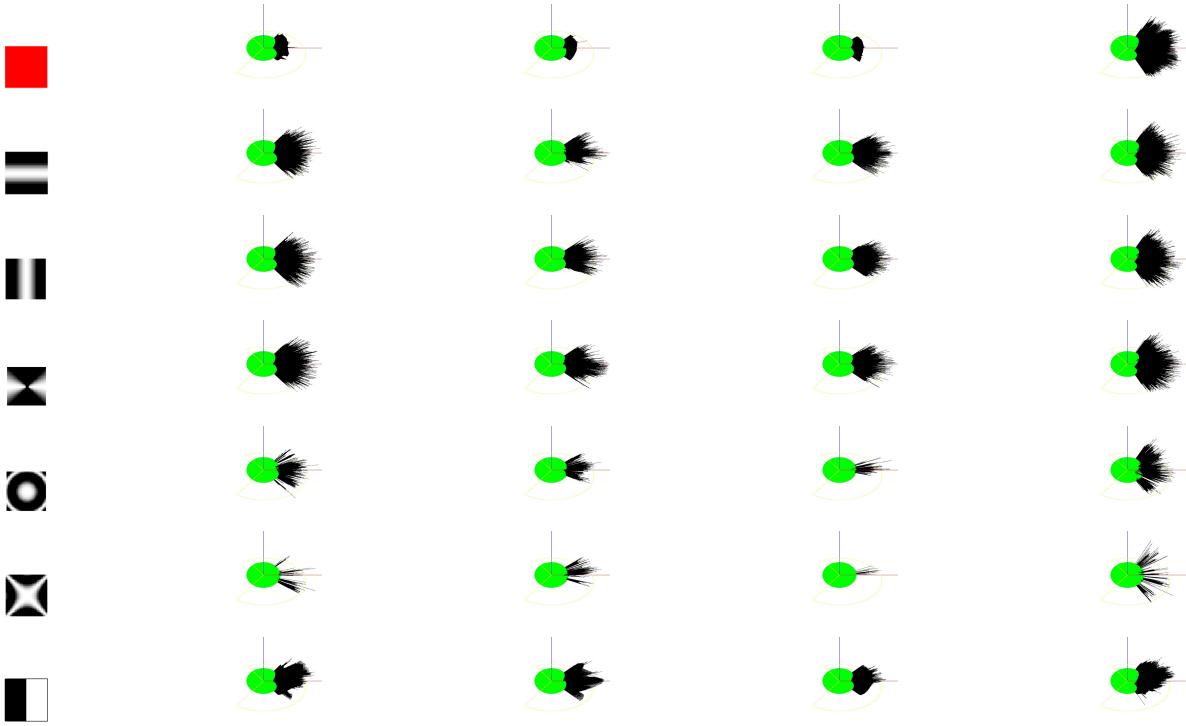
defined to be a collection of appearances sharing common perceptual signatures or physical boundaries. This is in contrast to viewing each appearance as a single place. Furthermore, each robot's place memory is processed as a whole or in portions. This is in contrast to processing each appearance individually.

All the proposed methods are implemented for real-time operation using ROS and extensively tested on datasets as well as outdoor Jaguar robots. In that sense this thesis is a step forward to perform autonomous tasks by reasoning about their environment using their learned knowledge about places.

The proposed approaches in this thesis is a first step in enabling a mobile robot to become spatially aware of its surroundings based on its experiences and obtained knowledge from other robots. A main issue of the appearance based methods is to avoid perceptual aliasing. This requires identifying distinct elements that is possible using pre-training. As we are strive for complete autonomy, this requires deductive learning. Fortunately, as the proposed TSC model that considers not only recognition, but also learning, future work can possibly consider learning based on contents of places and their distinctness. In fact, some of ongoing work has been actually focusing on this very topic. Moreover, assigning quantitative labels such as "kitchen in X flat" are also extremely desirable, but the robot needs to have scene analysis and possibly natural language capabilities. The higher level semantic information could be embedded to the place memory using scene segmentation techniques while topological maps can serve as a basis for creating hierarchical navigational paths for multi-robot systems.

APPENDIX A: BUBBLE SPACE

(a) Visual data from sample bases in the Fr, Lj , Sa and NC sites.



(b) Corresponding bubble surfaces for each of (color, Cartesian, non-Cartesian and intensity) features.

Figure A.1. Representation of visual data from sample bases in Fr, Sa, Lj and NC sites.

This section presents a brief summary of bubble space representation for completeness. The interested reader is referred to [6] for further details. The bubble space $\mathcal{B} = \mathcal{X} \times \mathcal{F}$ is an abstract representation of the robot's base along with its viewing directions (pan and tilt) $\mathcal{F} \subset S^2$ with $b \in \mathcal{B}$ defined as $b = [x\ f]^T$ where $x \in \mathcal{X}$ and $f \in \mathcal{F}$. Bubble surfaces $B_i(x,t) : Im(h(x)) \times R^{\geq 0} \rightarrow R^{\geq 0}$ are hypothetical spherical

surfaces surrounding the robot defined as:

$$B_i(x,t) = \left\{ \begin{bmatrix} f \\ \rho_i(b,t) \end{bmatrix} \mid \forall f \in \mathcal{F} \text{ and } b = [x\, f]^T \right\} \tag{A.1}$$

where the image of a section h – namely $Im(h(x))$ – is the set of viewing directions from a given base x with the section $h : \mathcal{X} \rightarrow \mathcal{B}$ defined as a continuous map such that $\forall x \in \mathcal{X}$, $\pi(h(x)) = x$ and $\pi : \mathcal{B} \rightarrow \mathcal{X}$ defined as the projection of b onto $\mathcal{X}$ as $\pi(b) = x$. Finally, the function $\rho_i : \mathcal{B} \times R^{\geq 0} \rightarrow R^{\geq 0}$ is a Riemannian metric that encodes the observed values of v_i^{th} sensory feature. For simplification of notation, the second argument is omitted whenever time dependency is clear. Each bubble surface is initialized to be a S^2 sphere with radius $\rho_0 \in R^{\geq 0}$ – namely $\rho_i(b,0) = \rho_0$. As the robot looks around, for each viewing direction $f \in \mathcal{F}$, it computes each feature value $q_i(b,t) \geq 0$. Next, each bubble surface $B_i(x,t)$ is deformed at the viewing direction f by an amount that depends on the associated sensory feature value $q_i(b,t)$ as:

$$\rho_i(b,t^+) = q_i(b,t) \tag{A.2}$$

where the superscript t^+ denotes time just after t. As this is done for each feature $v_i \in \mathcal{V}$ where $|\mathcal{V}| = N_v$, a set of N_v bubble surfaces is generated. In the experiments, the robot computes seven bubble surfaces corresponding to seven visual features (hue, Cartesian, non-Cartesian and intensity). For the sample scenes as shown in FigureA.1a, the bubble surfaces are as shown FigureA.1b. The intensity bubble surface is used for checking reliability of sensory data in place detection.

Bubble descriptors are holistic (vector) representations of bubble surfaces. They are constructed using the double Fourier series representation of bubble surfaces as:

$$\rho_i(b,t) = \sum_{h_1=0}^{H_1} \sum_{h_2=0}^{H_2} \lambda_{h_1 h_2} z^T_{xi,h_1 h_2}(t) e_{h_1 h_2}(f)$$

If $f \in \mathcal{F}$ is defined as $f = [f_1\ f_2]^T$, for each (h_1, h_2), the vector $e_{h_1h_2}(f) \in R^4$ consists of an orthonormal set of trigonometric basis functions as:

$$e_{h_1h_2}(f) = \begin{bmatrix} cos(h_1f_1)cos(h_2f_2) \\ sin(h_1f_1)cos(h_2f_2) \\ cos(h_1f_1)sin(h_2f_2) \\ sin(h_1f_1)sin(h_2f_2) \end{bmatrix} \tag{A.3}$$

The corresponding vector $z_{xi,h_1h_2}(t) \in R^4$ is defined as:

$$z_{xi,h_1h_2}(t) = \frac{1}{\pi^2} \begin{bmatrix} \int_0^{2\pi}\int_0^{\pi} \rho_i(b,t)cos(h_1f_1)cos(h_2f_2)df_1df_2 \\ \int_0^{2\pi}\int_0^{\pi} \rho_i(b,t)sin(h_1f_1)cos(h_2f_2))df_1df_2 \\ \int_0^{2\pi}\int_0^{\pi} \rho_i(b,t)cos(h_1f_1)sin(h_2f_2))df_1df_2 \\ \int_0^{2\pi}\int_0^{\pi} \rho_i(b,t)sin(h_1f_1)sin(h_2f_2)df_1df_2 \end{bmatrix} \tag{A.4}$$

The parameters $\lambda_{h_1h_2}$ are defined as:

$$\lambda_{h_1h_2} = \begin{cases} \frac{1}{4} & \text{if } h_1 = 0, h_2 = 0 \\ \frac{1}{2} & \text{if } h_1 > 0, h_2 = 0 \text{ or } h_1 = 0, h_2 > 0 \\ 1 & \text{if } h_1 > 0, h_2 > 0 \end{cases} \tag{A.5}$$

A bubble descriptor $I(x,t) \in R^{N_I}$ is a N_I-dimensional vector with $N_I = N_v(H_1 + 1)(H_2 + 1)$ defined as:

$$I(x,t) = [I_{1,00}(x,t), \ldots, I_{N_v,H_1H_2}(x,t)]^T \tag{A.6}$$

where

$$I_{i,h_1h_2}(x,t) = z^T_{xi,h_1h_2}(t)z_{xi,h_1h_2}(t) \tag{A.7}$$

Bubble descriptors have been shown to be rotationally invariant with respect to heading changes while being computable in an incremental manner- as new observations

are made. Furthermore, they are flexible integrating visual features since their dimensionality are independent of the number of observations. Furthermore, no data association [16] is required for finding correspondences among observations taken at different times.

APPENDIX B: IMAGE CLEF PERFORMANCE

In this section, for completeness, experimental results are evaluated based on the scoring used in the ImageClef 2012 challenge [65]. In particular, the score is computed as follows. First, it is initialized to zero. For each correctly classified frame, the score is incremented by one while each misclassified frame decrements the score by one. Unclassified frames have no effect on the score. Learning is varied by considering only daylight data only, night data only and both while testing is done using night data in all the tests. The experiments are repeated 6 times with $\tau \in [0.4, 0.9]$ with increments of 0.1. As discussed previously in Section 2.5, the best overall results and scores are achieved using night data in learning. The confidence parameter is optimal for values $0.5 < \tau < 0.7$. The integrated approach has the highest success rates for all of the combinations with visual data as the next best except with only daylight learning. Using only depth features gives the worst results as expected. With a maximum score of 874 for $\mathcal{L}_4$ and 1133 for $\mathcal{L}_5$, RGB-D based bubble space representations rank as 7^{th} and 5^{th} respectively.

B.1. Variations in Learning and Testing

As explained previously, all the experiments are done using the officially given test sequence in order to be able to compare our results with those of the ImageClef 2012 challenge. In this section, we consider varying the learning and test sets in regards to to robot's path or illumination conditions.

First, we consider learning (training set 1) and test (training set 2) data that differ in the robot's path while having same illumination conditions - namely daylight. The results are as given in Figure B.1. As expected, the results with using only depth features $\mathcal{L}_3$ are the worst due to dependency of this data on local geometry. Using only visual features $\mathcal{L}_1$ and $\mathcal{L}_2$ leads to significantly better performance. Integrated vision-depth sensing gives the best results but the results are very close to vision-only sensing.

As expected, the additional features in $\mathcal{L}_2$ and $\mathcal{L}_5$ improves the results compared to $\mathcal{L}_1$ and $\mathcal{L}_4$ although the improvement is not that gross. When these results are compared with those of Section 2.5, it is observed that the performance is close to that with night learning and testing. This is expected as in both cases, the illumination conditions are the same.

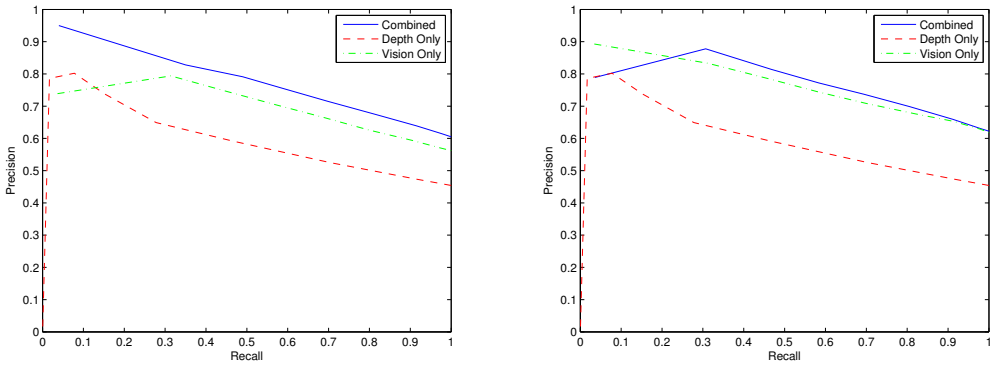


Figure B.1. Precision-recall curves for *training1 vs training2* data with varying feature sets. Left: For $\mathcal{L}_1$ $\mathcal{L}_3$ and $\mathcal{L}_4$; Right: For $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_5$.

Similar experiments are repeated via varying illumination conditions for learning and testing. While learning is done with night data using the training set 3, testing is done with daylight data using training set 1 - which is in contrast to the official test data. In both cases, the robot follows a similar path. In this case, it is observed that while using only visual features, the extended set does not contribute much to performance and is possibly misleading. Another interesting observation is that the depth-only sensing, despite being independent of illumination conditions, cannot have the edge over visual sensing. Limited field of view and depth range of 3D RGB-D cameras possibly increase the sensitivity of depth data to local geometry as compared to the traditional 2D laser range scanners. Using depth sensors having wider field of view with higher resolution would probably increase performance, but for now, visual sensing continues to be the primary sensing modality for place recognition with robots. Compared to results in *Experiments* section, the performance is even worse than daylight learning. We attribute this to the limited visual information of night data that does not allow generalizations. The rich sensory information from visual

data is lost when learning is done in night conditions. As observed earlier, learning plays a critical role in recognition performance.

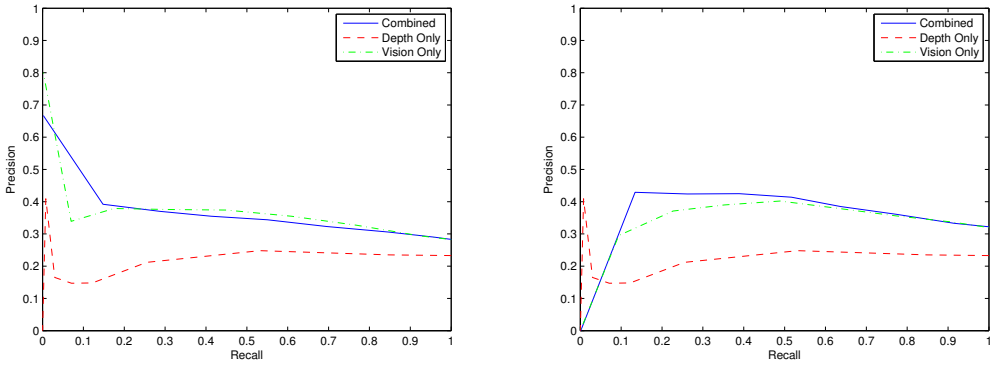


Figure B.2. Precision recall curves for *training3 vs training1* data with varying feature sets. Left: For $\mathcal{L}_1$ $\mathcal{L}_3$ and $\mathcal{L}_4$; Right: For $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_5$.

APPENDIX C: SENSORY DATA RELIABILITY

This section presents a brief discussion of how reliability is measured. Reliability depends on the informativeness, coherency and plenitude of the sensory data. Since sensory data are internally represented using descriptors, they can be measured via processing the descriptors appropriately. In our case, this processing uses the bubble descriptors. The interested reader is kindly referred to [12] for further details.

Informativeness measures whether an incoming sensory data is semantically rich or not. For example, in case of low illumination conditions, everything in the image will look dark. Similarly, if the robot field of view is comprised of an extended object such as a door, again there won't be much variation in the visual or depth images. An indication of both cases may be detected by computing the average deformation $\mu_i(x_k)$ or variance $\sigma_i(x_k)$ of the associated (intensity or depth) bubble surfaces $B_i(x_k, t_k)$:

$$\begin{aligned} \mu_i(x_k) &= \frac{1}{\pi^2}\int_0^{\pi}\int_0^{\pi} \rho_i(b,t_k)df_1df_2 \\ \sigma_i(x_k) &= \int_0^{2\pi}\int_0^{\pi} (\rho_i(b,t_k)-\mu_i(x_k))^2 df_1df_2 \end{aligned}$$

Low values indicate minimal surface deformation which implies that the data is not informative. Hence, the informativeness decision is based on a binary valued function $\varsigma : k \rightarrow \{0,1\}$:

$$\varsigma(x_k) = \begin{cases} 1 & \mu_i(x_k) \leq \tau_\eta \\ 1 & \sigma_i(x_k) \leq \tau_\sigma \\ 0 & \text{otherwise} \end{cases}$$

where τ_η and τ_σ are a priori selected threshold parameters. Sensory data from a particular base point x_k is used if and only if $\varsigma(x_k) = 0$. In this work, the bubble surface associated with intensity feature ($i = 7$) is used.

The coherency of data from two consecutive base points x_k and x_{k-1} is measured by comparing the similarity of their respective bubble descriptors $I(x_k)$ and $I(x_{k-1})$ using a χ^2-distance. For example, in case of jagged robot head or body motion, sensory data from consecutive bases will be quite unrelated. Thus, the incoherency decision is based on a binary valued function $\kappa : k \to \{0, 1\}$:

$$\kappa(x_k) = \begin{cases} 0 & \|I(x_k), I(x_{k-1})\|_{\chi^2} \leq \tau_\kappa \\ 1 & \text{otherwise} \end{cases}$$

A low similarity value as compared with the incoherency threshold τ_κ is an indicator of incoherency.

Finally, the pool of data associated with each place should be of sufficient amount. For example, sensory data from just a few base points - even if informative and coherent - will not in general be indicative of a particular place. The plenitude decision is based on the extent of the detected places D_m - namely those with extent less than a plenitude threshold τ_p are considered to have insufficient amount of data. The values of the informativeness thresholds τ_η and τ_σ, incoherency threshold τ_κ and the plenitude threshold τ_p affect the place detection performance. In this work, they are adjusted manually based on the camera type and nature of incoming sensory data.

APPENDIX D: JAGUAR ROBOT

D.1. Robot System, Hardware and Operation

Jaguar robot is designed for outdoor applications as shown in Figure D.1 (Left). It can operate in extreme terrains and is capable of climbing up stairs (up to 200mm step-length). The technical specifications of the Jaguar robot are given in Table D.1 and Table D.2. The complete robot platform that is composed of robot sensors, on-board computer and an observer laptop is shown in Figure D.1(Right).

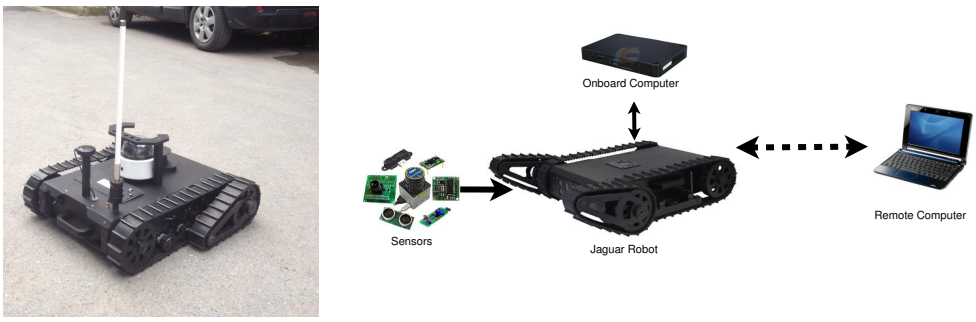


Figure D.1. Left: Jaguar robot; Right: Complete Robot platform.

D.1.1. Operating the Robot

In this section the details of operating the robot is explained.

(i) **Opening the Jaguar Robot:** Install the battery as shown in Figure D.2(left). Make sure that the red connector is connected with red and black with black. The robot is made operational by turning the switch on as shown in Figure D.2(right). A simple check to see if the robot is activated is to observe whether the red light of the laser scanner is on or not.

(ii) **Using the interface:**

- Open the computer running Ubuntu and connect to the Jaguar Robot with

Table D.1. Jaguar robot components.

No	Component	Unit	Properties
1	Length	mm	820
2	Height	mm	176
3	Width	mm	700
4	Portable Weight	kg	25
5	Battery	1	22.2V (Li-Po)
6	Motors	3	1 arm unit, 2 track-wheel unit
7	Encoders		JAGUAR-ME (1227.4 per revolution)
8	Camera 1		30fps, 640x480 resolution
9	Camera 2		30fps 640x480 resolution
10	2D Laser		Scanning Angle:270° (Resolution:0.25°)
11	GPS		OGPS501
12	IMU		IMU9000
13	Wireless		WRT802G

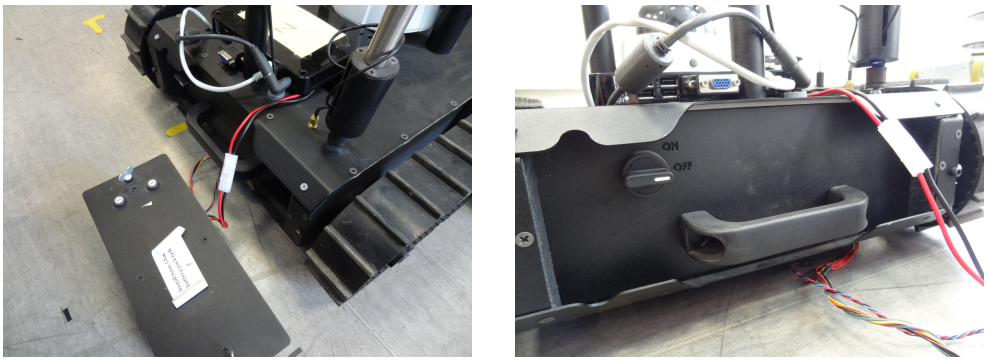


Figure D.2. Robot electrical connections for startup. Left: Battery connection; Right: Power Switch.

Table D.2. Jaguar robot - technical data.

Property	Unit	Theory	Experimental
Axis No	3	3	
Maximum Linear Speed	m/s	1.52	1
Maximum Rotational Speed	rad/sec	0.57	0.4
Maximum Arm Speed	rad/sec	1.52	1
Drive Method		tracked	
Maximum Slope to Climb Up	degree	higher than 45°	higher than 45°
External Interface		Wireless Connection	
Working Temperature	Celcius	Up to approx. 24°	

wireless as shown in Figure D.3. Default password for the wireless network is "drrobotdrrobot" and one need to specify a valid static ip address, ie. "192.168.0.157", to connect to the robot.

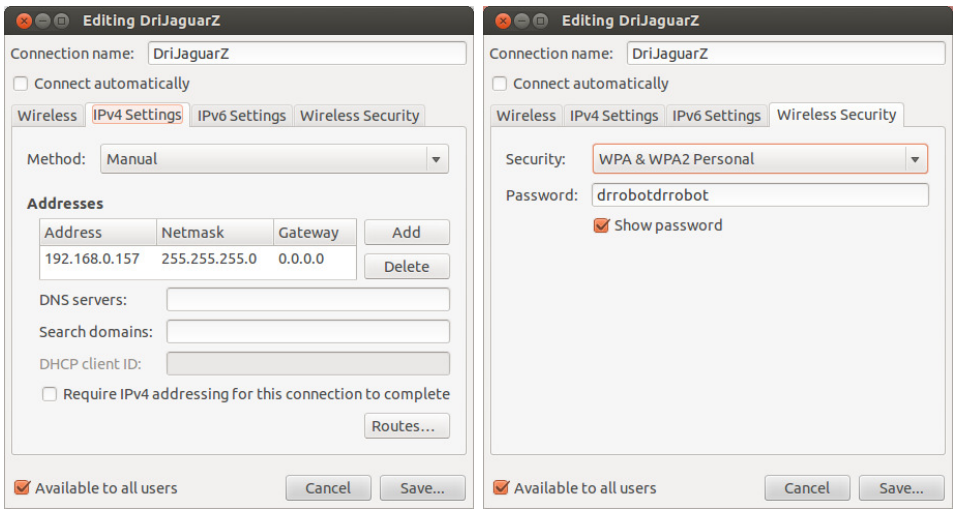


Figure D.3. Wireless settings for connecting to the robot.

- Open a terminal and run "`roscore`".
- Start all the nodes necessary ROS nodes using the following commands each on a different terminal tab.

```
roslaunch drrobot_jaguarV2_player drrobot_play1.launch
roslaunch axis_cameraHK axis1.launch
```

```
roslaunch axis_cameraHK axis_PTZ.launch
rosrun hokuyo_node_tcp hokuyo_node
```

- In case of "package not found" error, use the following command before each roslaunch and rosrun operation

```
source /home/jaguar/fuerte_workspace/setup.bash
```

- Now, the robot is ready to run. The interface can be invoked via going to the bin directory "jaguarControlISL/qtviewer/bin".

```
./imu_gps
```

- If it does not work, ensure imu_gps file is to be allowed to be run as an executable.

D.2. Robot Software User Guide

In this section, the graphical user interface tool that is used to control and command the robot as well as the software structure that is used throughout the experiments are explained.

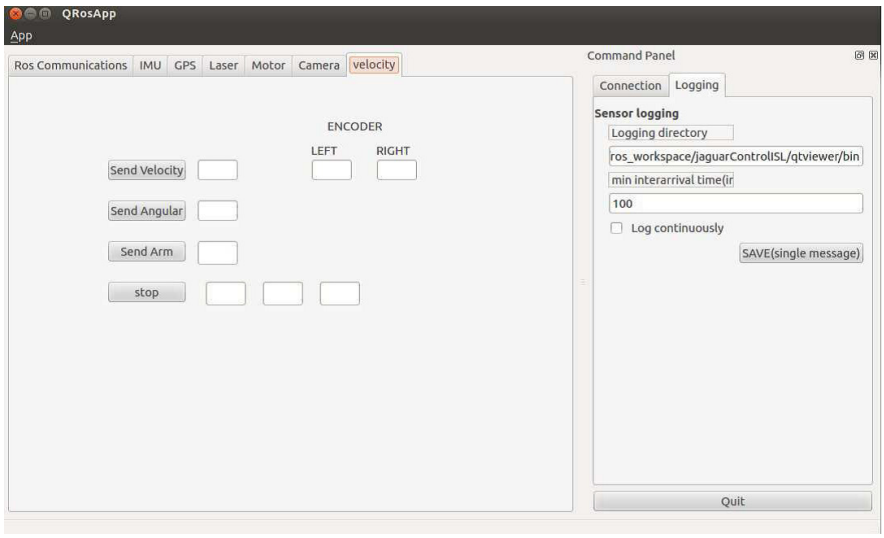


Figure D.4. Velocity tab.

D.2.1. Graphical Command & Control Tool

The command & control software has been designed using QT on the purpose of displaying data from various sensors of the robot and sending velocity commands to the motors in a convenient way. The interface uses specially implemented ROS nodes for the communication between user and robot components. Each node can be controlled using its related tab in the interface.

- Velocity Node: As shown in Figure D.4, the user can send velocity commands to wheels/arms of the robot and obtain encoder information using this tab. Velocity commands are sent in linear and angular parts. Encoder information has 16-bit resolution and distance traveled can be estimated by using it.
- Laser Node: This node handles the data from the laser range sensor. The laser range sensor is able to scan an angle of 270 degrees at 10 Hz frequency with a resolution of 0.25 degrees. It has a distance range of 30 meters. After some pre-processing, the measurement vector obtained from the laser scanner is visualized at the laser tab as shown in Figure D.5.

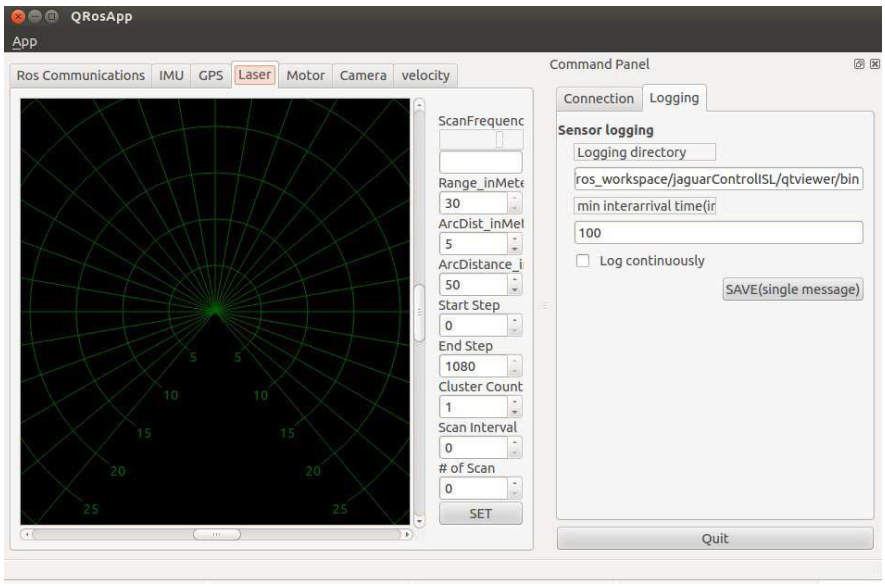


Figure D.5. Laser tab.

- Camera Node: This node provides visual data from PTZ and front cameras of

the robot and sends out PTZ commands. The robot can obtain visual data in every possible direction via these cameras. As shown in Figure D.6, the images are displayed and recorded using camera tab.

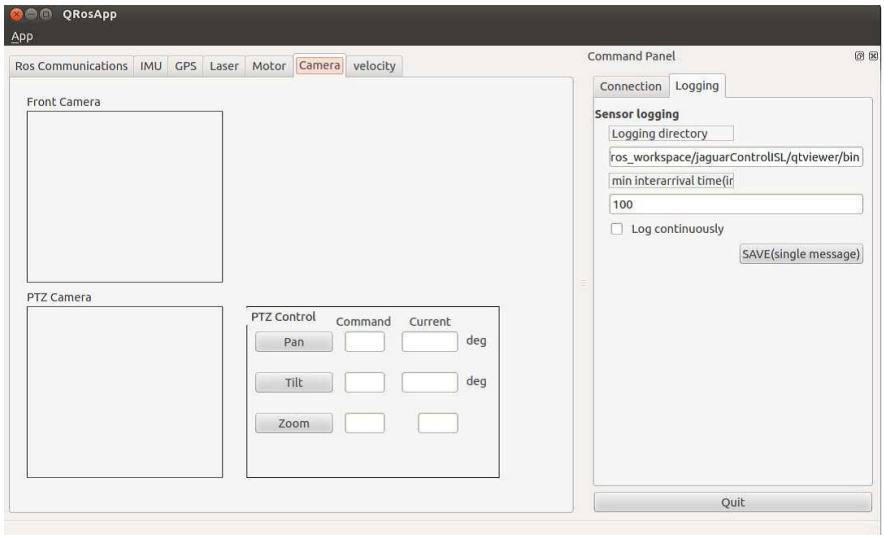


Figure D.6. Camera tab.

D.2.2. On-Robot Implementation of the TSC Model

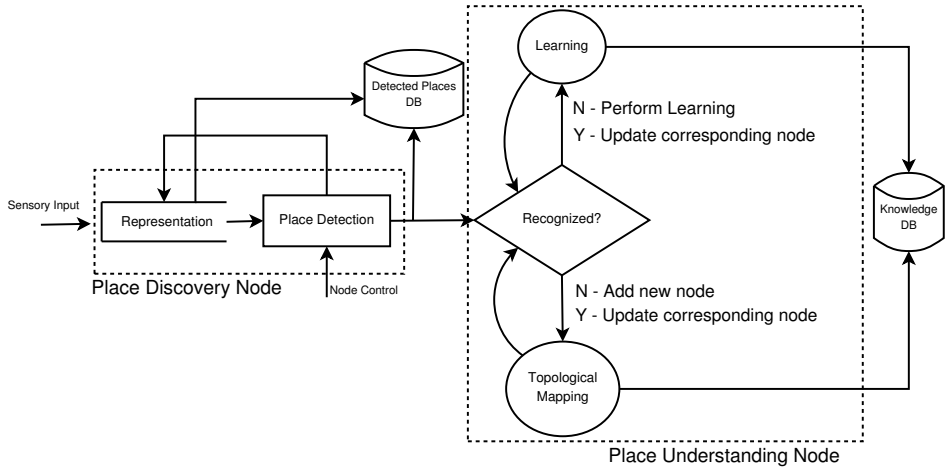


Figure D.7. Design of on-robot implementation of TSC model.

The proposed TSC model is developed in the framework of Robot Operating System (ROS) - an open-source robot meta-operating system. The on-robot implementation of the TSC model is achieved via the design and development of

2 ROS nodes and integrated with the previously developed sense-communicate-act architecture that is running on the robot [127]. The design of these two nodes is given in Figure D.7 where the dashed boxes show the internal structures of these nodes.

The place discovery node is composed of two modules: sensory data representation and place detection. The node subscribes to two messages. The first one is the *Sensory Input* - namely the image frame. The second one is the *Node Control* which controls the node's operation such as start, pause and shutdown. Upon receiving an image frame, it transforms it into an internal representation and then activates the place detection module. In the place detection module, there are two possibilities. Either place detection is in progress or a place is detected. In the former, the robot goes back to waiting for another sensory input. In the latter, a place detection event occurs. In this case, the robot first stores the knowledge associated with the detected place data as well as the transition region that led to this detected place in the robot's knowledge base as follows:

(i) Base points: The base point id, the associated descriptor and status information are stored. Status represents whether the base point is marked as uninformative, incoherent or coherent.

(ii) Detected places: The id, mean descriptor and member base point ids are stored.

(iii) Transition regions: The start and end points of a transitions between places and their ids are stored.

It then advertises the detected place ID.

The place understanding node is responsible for initializing, updating and maintaining long-term spatial memory. It is composed of recognition, learning and mapping modules. It works synchronously with the place discovery node and is activated whenever a place detection event occurs. Upon receiving the new detected place id, it reads the newly detected place from the *detected places* database and performs the necessary operations. If the place memory is empty , it initializes the place memory. If place memory is not empty, it actives the recognition module where either the newly detected place is recognized or not. In the

former, a recognition event occurs while in the latter, both place learning and mapping are invoked. Learning event occurs upon updating the place memory via incorporating the newly detected place into the place memory. Simultaneously, a mapping event occurs when the topological map is revised to include a new node and an edge that connects it to the previous place. The long-term memory is stored in *knowledge* database as follows:

(i) Place memory : The place memory structure including the children of the internal nodes, the connections between nodes and mean descriptors of internal nodes are stored.

(ii) Learned places: Learned places correspond to the terminal nodes of the place memory. The id, mean descriptor, member base point ids and member base point descriptors are stored. Note that a learned place can be composed of many detected places if these places are recognized to be from the same place. Thus, the variable member places stores the ids of detected places that compose the learned place.

(iii) Topological map: The topological map structure is stored as a connected list.

APPENDIX E: PUBLICATIONS

During this PhD study, the following papers have been published/submitted:

Chapter 2

- Karaoğuz H., Ö. Erkent and H.I. Bozma, "RGB-D based place representation in topological maps", *Machine Vision and Applications*, 25(8): 1913 - 1927, 2014.
- Karaoğuz H, Ö. Erkent, H. Bayram ve H.I. Bozma "Tek Robottan Çoklu Robotlara Ortam Haritalama",*EMO Bilimsel Dergi*, 4(2), 2013.
- Karaoğuz H., Ö. Erkent ve H.I. Bozma, "Topolojik Haritalarda 3B Uzaklık Ölçüm Bilgisine Dayalı Yer Gösterimi", *YTÜ BEEM Özel Sayısı*, 5(1), 74–85, 2013.

Chapter 3

- Karaoğuz H. and H.I. Bozma "An Integrated Model of Autonomous Topological Spatial Cognition", *Autonomous Robots*, 2015 (submitted).
- Erkent, Ö., H. Karaoğuz and H.I. Bozma, "Long Term Place Memory and Learning", *The International Journal of Robotics Research*, 2015 (submitted).
- Karaoğuz H. and H. I. Bozma, "Topological Place Recognition Based on Long-Term Memory Retrieval", *Int. Conf. Adv. Robot. (ICAR)*, 2015.
- Karaoguz H. and H.I. Bozma, "Reliable Topological Place Detection in Bubble Space", *In Proc. Int. Conf. on Robot. Autom. (ICRA)*, 697-702, 2014.
- Karaoğuz H. ve H.I. Bozma, "Topolojik Yer Tanıma için Uzun Dönemli Hafıza Tarama", *Türkiye Otonom Robotlar Konferansı*, 2014.

Chapter 4

- Karaoğuz, H., H. Bayram and H.I. Bozma."Communication Integrated Control Architecture in Multirobot Systems". *ICRA 2013 Workshop "Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration*, 2013.

REFERENCES

1. Tversky, B., "Functional Significance of Visuospatial Representations", *Handbook of Higher-Level Visuospatial Thinking*, pp. 1 – 34, Cambridge University Press, 2005.

2. Casati, R., "Topology and Cognition", *Encyclopedia of Cognitive Science*, Vol. 4, pp. 410–417, 2000.

3. Robert, L., *Spatial Cognition: Geographic Environments*, Kluwer Academic Publishers, Dordrecht, 1997.

4. Denis, M. and J. M. Loomis, "Perspectives on Human Spatial Cognition: Memory, Navigation, and Environmental Learning", *Psychological Research*, Vol. 71, No. 3, pp. 235–239, 2007.

5. Dolins, F. and R. Mitchell, "Linking Spatial Cognition and Spatial Perception", *Spatial Cognition, Spatial Perception: Mapping The Self and Space.*, pp. 1–31, 2010.

6. Erkent, O. and H. I. Bozma, "Bubble Space and Place Representation in Topological Maps", *International Journal of Robotics Research*, Vol. 32, No. 6, pp. 671 – 688, 2013.

7. Parker, L., "Distributed Intelligence: Overview of the Field And Its Application In Multi-robot Systems", *Journal of Physical Agents*, pp. 1–6, 2008.

8. Pronobis, A., O. Martinez Mozos and B. Caputo, "SVM-based Discriminative Accumulation Scheme For Place Recognition", *IEEE International Conference on Robotics and Automation*, pp. 522–529, 2008.

9. Erkent, O. and H. I. Bozma, "Place Representation in Topological Maps Based

on Bubble Space", *IEEE International Conference on Robotics and Automation*, pp. 3497 –3502, 2012.

10. Samatova, N. and G. Ostrouchov, "RACHET: An Efficient Cover-based Merging of Clustering Hierarchies from Distributed Datasets", *Distributed and Parallel Databases*, pp. 157–180, 2002.

11. Karaoguz, H., O. Erkent and H. Bozma, "RGB-D Based Place Representation in Topological Maps", *Machine Vision and Applications*, pp. 1–15, 2014.

12. Karaoguz, H. and H. I. Bozma, "Reliable Topological Place Detection in Bubble Space", *IEEE International Conference on Robotics and Automation*, pp. 697–702, 2014.

13. Karaoguz, H. and H. I. Bozma, "Topological Place Recognition Based on Long-Term Memory Retrieval", *International Conference on Advanced Robotics*, 2015.

14. Erkent, O. and H. I. Bozma, "Long-Term Topological Place Learning", *IEEE International Conference on Robotica and Automation*, pp. 5462–5467, 2015.

15. Ulrich, I. and I. Nourbakhsh, "Appearance-based Place Recognition for Topological Localization", *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1023 – 1029, 2000.

16. Williams, B., 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.

17. Henry, P., M. Krainin, E. Herbst, X. Ren and D. Fox, "RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments", *International Journal of Robotics Research*, Vol. 31, No. 5, pp. 647–663, 2012.

18. Shi, S., L. Kodagoda and R. Ranasinghe, "Fast Indoor Scene Classification Using

3D Point Clouds", *Australasian Conference on Robotics and Automation*, 2011.

19. Harris, C. and M. Stephens, "A Combined Corner and Edge Detector", *4th Alvey Vision Conference*, pp. 147–151, 1988.

20. Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

21. Bay, H., T. Tuytelaars and L. Van Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, pp. 346–359, 2008.

22. Lamon, R., I. Nourbakhsh, B. Jensenl and R. Siegwart, "Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization", *IEEE International Conference on Robotics and Automation*, pp. 1609–1610, 2001.

23. Bozma, H. I., G. Çakiroglu and C. Soyer, "Biologically Inspired Cartesian and Non-Cartesian Filters for Attentional Sequences", *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1261–1274, 2003.

24. Mozos, O. and W. Burgard, "Supervised Learning of Topological Maps using Semantic Information Extracted from Range Data", *IEEE International Conference on Robotics and Automation*, pp. 2722 – 2777, 2006.

25. Vasudevan, S., S. Gachter, V. Nguyen and R. Siegwart, "Cognitive Maps for Mobile Robots-An Object Based Approach", *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 359–371, 2007.

26. Fazl-Ersi, E. and J. K. Tsotsos, "Histogram of Oriented Uniform Patterns for Robust Place Recognition and Categorization", *International Journal of Robotics Research*, Vol. 31, No. 4, pp. 468–483, 2012.

27. Davison, A., I. Reid, N. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", *IEEE Transactions on Pattern Analysis and Machine Intelli-*

gence, Vol. 29, No. 6, pp. 1052 –1067, 2007.

28. Oliva, A. and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope", *International Journal of Computer Vision*, Vol. 42, No. 3, pp. 145–175, 2001.

29. Torralba, A., K. P. Murphy, W. T. Freeman and M. A.Rubin, "Context-based Vision System for Place and Object Recognition", *IEEE International Conference on Computer Vision*, Vol. 1, p. 273, 2003.

30. Jogan, M. and A. Leonardis, "Robust Localization Using Panoramic View-based Recognition", *Pattern Recognition, 15th International Conference on*, Vol. 4, pp. 136 –139, 2000.

31. Qiu, G., "Indexing Chromatic and Achromatic Patterns for Content-based Colour Image Retrieval", *Pattern Recognition*, Vol. 35, No. 8, pp. 1675 – 1686, 2002.

32. Pronobis, A. and B. Caputo, "Confidence-based Cue Integration for Visual Place Recognition", *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 2394–2401, 2007.

33. Weiss, C., H. Tamimi, A. Masselli and A. Zell, "A Hybrid Approach For Vision-based Outdoor Robot Localization Using Global And Local Image Features", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1047 –1052, 2007.

34. Xing, L. and A. Pronobis, "Multi-cue Discriminative Place Recognition", *Multilingual Information Access Evaluation II. Multimedia Experiments*, Vol. 6242, pp. 315–323, 2010.

35. Konolige, K., J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit and P. Fua, "View-based Maps", *International Journal of Robotics Research*, Vol. 29, No. 8, pp. 941–957, 2010.

36. Murillo, A., J. Guerrero and C. Sagues, "SURF Features For Efficient Robot Localization With Omnidirectional Images", *IEEE International Conference on Robotics and Automation*, pp. 3901 –3907, 2007.

37. Nowak, E., F. Jurie and B. Triggs, "Sampling Strategies for Bag-of-features Image Classification", *European Conference on Computer Vision*, pp. 490–503, 2006.

38. Bosch, A., A. Zisserman and X. Munoz, "Scene Classification Using a Hybrid Generative-Discriminative Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 4, pp. 712–727, 2008.

39. Lazebnik, S., C. Schmid and J. Ponce, "Beyond Bags Of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.

40. Cummins, M. and P. Newman, "Appearance-Only SLAM at Large Scale with FAB-MAP 2.0", *The International Journal of Robotics Research*, Vol. 30, No. 9, pp. 1100–1123, 2011.

41. Fraundorfer, F., C. Engels and D. Nister, "Topological Mapping, Localization and Navigation Using Image Collections", *IEEE International Conference on Intelligent Robots and Systems*, pp. 3872 – 3387, 2007.

42. Wolf, J., W. Burgard and H. Burkhardt, "Robust Vision-based Localization by Combining an Image-Retrieval System with Monte Carlo Localization", *IEEE Transactions on Robotics*, Vol. 21, No. 2, pp. 208 – 216, 2005.

43. Blumenthal, S., E. Prassler, J. Fischer and W. Nowak, "Towards Identification of Best Practice Algorithms in 3D Perception and Modeling", *IEEE International Conference on Robotics and Automation*, pp. 3554–3561, 2011.

44. Steder, B., G. Grisetti and W. Burgard, "Robust Place Recognition for 3D Range Data Based on Point Features", *IEEE International Conference on Robotics and*

Automation, pp. 1400–1405, 2010.

45. Smith, M., I. Posner and P. Newman, "Adaptive Compression for 3D Laser Data", *International Journal of Robotics Research*, Vol. 30, No. 7, pp. 914–935, 2011.

46. Wang, L., J. Chen and B. Yuan, "Simplified Representation for 3D Point Cloud Data", *IEEE International Conference on Signal Processing*, pp. 1271 – 1274, 2010.

47. Magnusson, M., H. Andreasson, A. Nuchter and A. Lilienthal, "Appearance-based Loop Detection from 3d Laser Data Using The Normal Distributions Transform", *IEEE International Conference on Robotics and Automation*, pp. 23–28, 2009.

48. Bogdan, R., N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration", *IEEE International Conference on Robotics and Automation*, pp. 1848–1853, 2009.

49. Steder, B., M. Ruhnke, S. Grzonka and W. Burgard, "Place Recognition in 3D Scans Using a Combination of Bag Of Words and Point Feature Based Relative Pose Estimation", *IEEE International Conference on Intelligent Robots and Systems*, pp. 1249–1255, 2011.

50. Granström, K., T. Schön, J. Nieto and F. Ramos, "Learning to Close Loops from Range Data", *International Journal of Robotics Research*, Vol. 30, No. 14, pp. 1–27, 2011.

51. Mozos, O. M., A. Rottmann, R. Triebel, P. Jensfelt and W. Burgard, "Semantic Labeling of Places Using Information Extracted From Laser and Vision Sensor Data", *IEEE IROS Workshop: From Sensors to Human Spatial Concepts*, 2006.

52. Mozos, O. M., R. Triebel, P. Jensfelt, A. Rottmann and W. Burgard, "Supervised Semantic Labeling of Places Using Information Extracted From Sensor Data", *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 391–402, 2007.

53. Pronobis, A., O. Martinez Mozos, B. Caputo and P. Jensfelt, "Multi-modal Semantic Place Classification", *The International Journal of Robotics Research*, Vol. 29, No. 2-3, pp. 298–320, 2010.

54. Shi, L., S. Kodagoda and G. Dissanayake, "Laser Range Data Based Semantic Labeling of Places", *IEEE International Conference on Intelligent Robots and Systems*, pp. 5941 –5946, 2010.

55. Sousa, P., R. Araiijo and U. Nunes, "Real-Time Labeling of Places Using Support Vector Machines", *IEEE International Symposium on Industrial Electronics*, pp. 2022–2027, 2007.

56. Martínez-Gómez, J., I. García-Varea and B. Caputo, "Baseline Multimodal Place Classifier for the 2012 Robot Vision Task.", *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

57. Boros, E., A.-L. Gînsca and A. Iftene, "UAIC Participation at Robot Vision @ 2012 - An Updated Vision.", *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

58. Perronnin, F., Y. Liu, J. Sanchez and H. Poirier, "Large-scale Image Retrieval with Compressed Fisher Vectors", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3384 –3391, 2010.

59. Redolfi, J. and J. Sánchez, "Leveraging Robust Signatures for Mobile Robot Semantic Localization", *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

60. Mozos, O. M., H. Mizutani, R. Kurazume and T. Hasegawa, "Categorization of Indoor Places Using the Kinect Sensor", *Sensors*, Vol. 12, No. 5, pp. 6695–6711, 2012.

61. Larson, A. M. and L. C. Loschky, "The Contributions of Central Versus Peripheral Vision to Scene Gist Recognition", *Journal of Vision*, Vol. 9, No. 10, pp. 1–16, 9

2009.

62. Platt, J. C., "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods", *Advances in Large Margin Classifiers*, Vol. 10, No. 3, pp. 61–74, 1999.

63. Wang, M.-L. and H.-Y. Lin, "An Extended-HCT Semantic Description for Visual Place Recognition", *International Journal of Robotics Research*, Vol. 30, No. 11, pp. 1403–1420, 2011.

64. Chang, C.-C. and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, pp. 1–27, 2011.

65. Martinez-Gomez, I., J. Garcia-Varea and B. Caputo, "Overview of the ImageCLEF 2012 Robot Vision Task", *In CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*, 2012.

66. Zivkovic, Z., O. Booij and B. Kröse, "From Images to Rooms", *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 411–418, 2007.

67. Ranganathan, A., "PLISS: Detecting and Labeling Places Using Online Change-Point Detection", *Proc. Robotics: Science and Systems*, 2010.

68. Cummins, M. and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance", *International Journal of Robotics Research*, Vol. 27, pp. 647–665, 2008.

69. Newman, P., G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole and I. Reid, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers", *International Journal of Robotics Research*, Vol. 28, No. 11-12, pp. 1406–1433, 2009.

70. Murphy, L. and G. Sibley, "Incremental Unsupervised Topological Place Discov-

ery", *International Conference on Robotics and Automation*, pp. 1312 – 1318, 2014.

71. Tapus, A. and R. Siegwart, "Incremental Robot Mapping with Fingerprints of Places", *International Conference on Intelligent Robots and Systems*, pp. 2429–2434, 2005.

72. Ho, K. L. and P. Newman, "Detecting Loop Closure with Scene Sequences", *International Journal of Computer Vision*, Vol. 74, No. 3, pp. 261–286, 2007.

73. Vasudevan, S. and R. Siegwart, "Bayesian Space Conceptualization and Place Classification for Semantic Maps in Mobile Robotics", *Robotics and Autonomous Systems*, Vol. 56, No. 6, pp. 522–537, 2008.

74. Tversky, B. and K. Hemenway, "Categories of Scenes", *Cognitive Psychology*, Vol. 15, pp. 121 – 149, 1983.

75. Tversky, B., "Cognitive Maps, Cognitive Collages, and Spatial Mental Models", *Spatial Information Theory A Theoretical Basis for GIS*, pp. 14–24, Springer, 1993.

76. Pronobis, A., K. Sjöö, A. Aydemir, A. N. Bishop and P. Jensfelt, "Representing Spatial Knowledge in Mobile Cognitive Systems", *11th International Conference on Intelligent Autonomous Systems*, 2010.

77. Pronobis, A. and P. Jensfelt, "Large-scale Semantic Mapping and Reasoning with Heterogeneous Modalities", *International Conference on Robotics and Automation*, pp. 3515–3522, 2012.

78. Chella, A., I. Macaluso and L. Riano, "Automatic Place Detection and Localization in Autonomous Robotics", *International Conference on Intelligent Robots and Systems*, pp. 741–746, 2007.

79. Kuipers, B., "The Spatial Semantic Hierarchy", *Artificial Intelligence*, Vol. 119, No. 1-2, pp. 191 – 233, 2000.

80. Beeson, P., J. Modayil and B. Kuipers, "Factoring the Mapping Problem: Mobile Robot Map-building in the Hybrid Spatial Semantic Hierarchy", *International Journal of Robotics Research*, Vol. 29, No. 4, pp. 428–459, 2010.

81. Mozos, O. M., P. Jensfelt, H. Zender, G.-J. M. Kruijff and W. Burgard, "From Labels to Semantics: An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots", *IEEE/RSJ IROS Workshop: From Sensors to Human Spatial Concepts*, 2007.

82. Ranganathan, A., "PLISS: Labeling Places Using Online Changepoint Detection", *Autonomous Robots*, Vol. 32, No. 4, pp. 351–368, 2012.

83. Shi, L., S. Kodagoda and G. Dissanayake, "Application of Semi-supervised Learning with Voronoi Graph for Place Classification", *International Conference on Intelligent Robots and Systems*, pp. 2991–2996, 2012.

84. Ursic, P., M. Kristan, D. Skocaj and A. Leonardis, "Room Classification Using A Hierarchical Representation of Space", *International Conference on Intelligent Robots and Systems*, pp. 1371–1378, 2012.

85. Yeh, T. and T. Darrell, "Dynamic Visual Category Learning", *International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

86. Lim, J., J. M. Frahm and M. Pollefeys, "Online Environment Mapping Using Metric-topological Maps", *International Journal of Robotics Research*, Vol. 31, No. 12, pp. 1394–1408, 2012.

87. Zivkovic, Z., B. Bakker and B. Krose, "Hierarchical Map Building Using Visual Landmarks and Geometric Constraints", *International Conference on Intelligent Robots and Systems*, pp. 2480–2485, 2005.

88. Posner, I., D. Schroeter and P. M. Newman, "Using Scene Similarity for Place Labelling", *Experimental Robotics*, Vol. 39, pp. 85–98, Springer, 2008.

89. Martinez-Gomez, J. and B. Caputo, "Towards Semi-Supervised Learning of Semantic Spatial Concepts", *International Conference on Robotics and Automation*, pp. 1936–1943, 2011.

90. Liu, M. and R. Siegwart, "DP-FACT: Towards Topological Mapping and Scene Recognition with Color for Omnidirectional Camera", *International Conference on Robotics and Automation*, pp. 3503–3508, 2012.

91. Walter, M. R., S. Hemachandra, B. Homberg, S. Tellex and S. Teller, "A Framework for Learning Semantic Maps from Grounded Natural Language Descriptions", *International Journal of Robotics Research*, Vol. 33, No. 9, pp. 1167–1190, 2014.

92. Chang, C. C. and C. J. Lin, "LIBSVM: A Library for Support Vector Machines", *ACM Transactions on Intelligent Sytems and Technology*, Vol. 2, pp. 1–27, 2011.

93. Remolina, E. and B. Kuipers, "Towards a General Theory of Topological Maps", *Artificial Intelligence*, Vol. 152, No. 1, pp. 47 – 104, 2004.

94. Sibson, R., "SLINK: An Optimally Efficient Algorithm For The Single-link Cluster Method", *The Computer Journal*, Vol. 16, No. 1, pp. 30–34, 1973.

95. Pronobis, A. and B. Caputo, "COLD: COsy Localization Database", *International Journal of Robotics Research*, Vol. 28, No. 5, pp. 588–594, 2009.

96. Smith, M., I. Baldwin, W. Churchill, R. Paul and P. Newman, "The New College Vision and Laser Data Set", *International Journal of Robotics Research*, Vol. 28, No. 5, pp. 595–599, 2009.

97. Stipes, J., R. Hawthorne, D. Scheidt and D. Pacifico, "Cooperative Localization

and Mapping", *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pp. 596–601, 2006.

98. Erinc, G. and S. Carpin, "Anytime Merging of Appearance-based Maps", *Autonomous Robots*, Vol. 36, No. 3, pp. 241–256, 2014.

99. Zhou, X. and S. Roumeliotis, "Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1785–1792, 2006.

100. Nieto-Granda, C., J. G. Rogers and H. I. Christensen, "Coordination Strategies for Multi-robot Exploration and Mapping", *International Journal of Robotics Research*, 2014.

101. Lee, H.-C. and B.-H. Lee, "Improved Feature Map Merging Using Virtual Supporting Lines for Multi-Robot Systems", *Advanced Robotics*, Vol. 25, No. 13-14, pp. 1675–1696, 2011.

102. Leung, K., T. Barfoot and H. Liu, "Distributed and Decentralized Cooperative Simultaneous Localization and Mapping For Dynamic And Sparse Robot Networks", *International Conference on Robotics and Automation*, pp. 3841–3847, 2011.

103. Aragues, R., J. Cortes and C. Sagues, "Distributed Consensus on Robot Networks for Dynamically Merging Feature-based Maps", *IEEE Transactions on Robotics*, Vol. 28, No. 4, pp. 840–854, 2012.

104. Thrun, S., W. Burgard and D. Fox, "A Real-time Algorithm for Mobile Robot Mapping with Applications to Multi-robot and 3d Mapping", *IEEE International Conference on Robotics and Automation.*, Vol. 1, 2000.

105. Williams, S. B., G. Dissanayake and H. Durrant-Whyte, "Towards Multi-vehicle Simultaneous Localisation and Mapping", *IEEE International Conference on*

Robotics and Automation, Vol. 3, pp. 2743–2748, 2002.

106. Konolige, K., D. Fox, B. Limketkai, J. Ko and B. Stewart, "Map Merging for Distributed Robot Navigation", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 212–217, 2003.

107. Ko, J., B. Stewart, D. Fox, K. Konolige and B. Limketkai, "A Practical, Decision-theoretic Approach to Multi-robot Mapping and Exploration", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 4, 2003.

108. Howard, A., "Multi-robot Mapping Using Manifold Representations", *IEEE International Conference on Robotics and Automation*, pp. 4198–4203, Ieee, 2004.

109. Ozcukur, E., B. Kurt and L. Akin, "A Collaborative Multi-Robot Localization Method Without Robot Identification", *RoboCup 2008: Robot Soccer World Cup XII*, Vol. 5399, pp. 189–199, 2009.

110. Gil, A., O. Reinoso, M. Ballesta and M. Juliá, "Multi-robot Visual SLAM Using a Rao-Blackwellized Particle Filter", *Robotics and Autonomous Systems*, Vol. 58, No. 1, pp. 68–80, 2010.

111. Thrun, S., "A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots", *International Journal of Robotics Research*, Vol. 20, No. 5, pp. 335–363, 2001.

112. Hajjdiab, H. and R. Laganiere, "Vision-based Multi-robot Simultaneous Localization and Mapping", *First Canadian Conference on Computer and Robot Vision*, pp. 155–162, 2004.

113. Carpin, S., A. Birk and V. Jucikas, "On Map Merging", *Robotics and Autonomous Systems*, Vol. 53, pp. 1–14, 2005.

114. Birk, A. and S. Carpin, "Merging Occupancy Grid Maps from Multiple Robots",

Proceedings of the IEEE: Special Issue on Multi-Robot Systems, Vol. 94, No. 7, pp. 1384–1397, 2006.

115. Amigoni, F., S. Gasparini and M. Gini, "Building Segment-based Maps Without Pose Information", *Proceedings of the IEEE: Special Issue on Multi-Robot Systems*, Vol. 94, No. 7, pp. 1340–1359, 2006.

116. Adluru, N., L. Latecki, M. Sobel and R. Lakaemper, "Merging Maps of Multiple Robots", *19th International Conference on Pattern Recognition*, pp. 8–11, 2008.

117. Ma, X., R. Guo, Y. Li and W. Chen, "Adaptive Genetic Algorithm for Occupancy Grid Maps Merging", *World Congress on Intelligent Control and Automation (WCICA)*, pp. 5704–5709, 2008.

118. Tungadi, F., W. L. D. Lui, L. Kleeman and R. Jarvis, "Robust Online Map Merging System Using Laser Scan Matching and Omnidirectional Vision", *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*, pp. 7–14, 2010.

119. Balakirsky, S., T. Hemker, M. Reggiani and O. von Stryk, *Simulation, Modeling, and Programming for Autonomous Robots*, Springer, NewYork, 2010.

120. Marjovi, A., S. Choobdar and L. Marques, "Robotic Clusters: Multi-robot Systems as Computer Clusters", *Robotics and Autonomous Systems*, Vol. 60, No. 9, pp. 1191–1204, 2012.

121. Tomono, M., "Merging of 3d Visual Maps Based On Part-map Retrieval and Path Consistency", *IEEE International Conference on Intelligent Robots and Systems*, pp. 5172–5179, 2013.

122. Carpin, S., "Fast and Accurate Map Merging for Multi-robot Systems", *Autonomous Robots*, Vol. 25, No. 3, pp. 305–316, 2008.

123. Saeedi, S., L. Paull and M. Trentini, "Map Merging Using Hough Peak Matching",

International Conference on Intelligent Robots and Systems, pp. 4683–4688, 2012.

124. Kuipers, B. and Y.-T. Byun, "A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations", *Journal of Robotics and Autonomous Systems*, Vol. 8, pp. 47–63, 1991.

125. Huang, W. H., "Topological Map Merging", *International Journal of Robotics Research*, Vol. 24, No. 8, pp. 601–613, 2005.

126. Ferreira, F., J. Dias and V. Santos, "Merging Topological Maps for Localisation in Large Environments", *International Climbing and Walking Robots and The Support Technologies for Mobile Machines*, pp. 1–17, 2008.

127. Karaoguz, H., H. Bayram and H. I. Bozma, "Communication Integrated Control Architecture in Multirobot Systems", *ICRA Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration*, 2013.