FROM PLACE DETECTION TO LONG-TERM PLACE MEMORY

by

Mahmut Demir

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2012

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

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2016

ACKNOWLEDGEMENTS

I have been a member of ISL laboratory since 2008 and it was always a pleasure to work with all the wonderful people in ISL lab. First of all, I would like to express my sincere gratitude to my supervisor Prof. Işıl Bozma. She has been a great advisor to me with her guidance, patience and enthusiasm throughout my undergraduate and master's education. Her guidance and tremendous support had a major influence on this thesis.

I would like to thank Prof. Yağmur 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 also like to express my sincerest thanks and appreciation to all ISL members including Esen Yel, Çağatay Odabaşı, Deniz Seviş Şenel, Gökçe Erdem, Berkan Höke but especially Halil Samed Çıldır and Kadir Türksoy for their countless support during software development and testing.

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

This work has been supported in part by TUBITAK EEEAG-115E380.

ABSTRACT

FROM PLACE DETECTION TO LONG-TERM PLACE MEMORY

This thesis is concerned with automated place detection and its coupling with long-term place memory. Long-term place memory is critical if a robot is to be place-aware as it stores the relevant knowledge for future referral. Place detection is closely coupled to place memory as it determines the appearances belonging to each place and thus plays a key role in regards to which knowledge gets retained in the long-term place memory. In this perspective, the contributions of the thesis are as follows: First, it introduces a novel approach to place detection based on coherent visual segments. Second, a new approach to place representation referred to as 'segments summary graph' is presented. Finally, it is shown that this representation can be utilized for improving the reliability of memory association.

ÖZET

YER SEZİMLEMESİNDEN UZUN DÖNEMLİ HAFIZALARA GEÇİŞ

Bu tezde, otonom bir şekilde yer sezimleme yapılması ve bunun yer hafızası ile ilişkilendirilmesi konusu ele alınmıştır. Robotun bulunduğu ortama ait farkındalık sağlamasında, yer hafızasının ortama ait tüm bilgiyi ileri vadede kullanılmak üzere depolaması nedeniyle önemli bir yeri vardır. Bu noktada, yer sezimleme ise o yere ait görüntü kümelerini yer ile ilişkilendirdiğinden ve hangi bilginin depolanması gerektiğini tayin ettiğinden yer hafızasının oluşturulması aşamasında kritik bir noktada bulunmaktadır. Bu noktaları göz önünde bulundurduğumuzda, tezin katkılarını şöyle sıralayabiliriz: İlk olarak, yer sezimleme problemine çözüm olarak görsel sahne bölütlerinin takip edilmesi ve görsel verinin uyumluluğu tabanlı bir yaklaşım geliştirilmiştir. Böylece her bir yere ait görüntüler kümesi yer hafızasında uygun bir şekilde konumlandırılmış ve depolanmış olacaktır. İkinci olarak, yaklaşımımız yerlerin tanımlanmasında bölüt tabanlı bir gösterim şekli önermektedir. Son olarak, elde ettiğimiz özet bölüt gösteriminin hafıza ilişkilendirilmesinin daha tutarlı bir hale getirilmesinde kullanılması üzerine bir metod geliştirilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
1.1. Contributions . . . 1
1.2. Outline . . . 2
2. PLACE DETECTION . . . 3
2.1. Introduction . . . 3
2.2. Related Literature . . . 3
2.3. General Approach . . . 5
2.4. Segmentation . . . 5
2.5. Region Adjacency Graphs . . . 7
2.6. RAG Matching and Node Existence Matrix . . . 7
2.7. Place Detection . . . 10
2.8. Place Representation and Segments Summary Graphs . . . 12
2.9. Experimental Results . . . 13
2.9.1. Video Summarization Results . . . 14
2.9.2. Place Detection Results . . . 15
2.9.3. Experiments with Jaguar Robot . . . 24
2.10. Conclusion . . . 26
3. INTEGRATION WITH PLACE MEMORY . . . 29
3.1. Introduction . . . 29
3.2. Place Memory . . . 29
3.3. Experimental Results . . . 31

3.3.1. Place Memory 32
3.3.2. Place Association 35
3.3.3. On-Robot Experiments 42
3.4. Conclusion 46
4. CONCLUSION 47
REFERENCES 48
APPENDIX A: BUBBLE SPACE 52
APPENDIX B: SSG SOFTWARE MANUAL 56

LIST OF FIGURES

Figure 2.1. SSG based place detection. 5

Figure 2.2. Segmentation results for different sets of parameters 6

Figure 2.3. Matching two RAGs. 8

Figure 2.4. Node existence matrix 9

Figure 2.5. Place detection methodology. 11

Figure 2.6. Comparative place detection results. 16

Figure 2.7. Place detection as the robot navigates along a short path in Fr site. 18

Figure 2.8. Place detection as the robot navigates through NC site 19

Figure 2.9. Comparative evolutions of coherency in Fr site 20

Figure 2.10. Comparative place detection results for Fr site 21

Figure 2.11. Comparative place detection results for Lj site 21

Figure 2.12. Comparative place detection results for Sa site 22

Figure 2.13. Jaguar robot 26

Figure 2.14. Experiments with Jaguar robot in North Campus site 27

Figure 2.15. Performance analysis: Processing time per frame. 28

Figure 3.1. Place memory and association 30

Figure 3.2. Place memory association. 32

Figure 3.3. Planar projections of BD and SSG based place descriptors 34

Figure 3.4. Place memories 36

Figure 3.5. Place detection in Fr site 37

Figure 3.6. Place memories after second time visits 39

Figure 3.7. Place detection in Sa site 40

Figure 3.8. Place detection in Lj site 41

Figure 3.9. Place memory after revisiting all places 43

Figure 3.10. Precision-Recall curves after revisiting sites 44

Figure 3.11. Place memory after revisiting the North Campus site 46

Figure A.1. Sample bases and visual data. 52

Figure B.1. Main screen of Segments Summary Graphs software 57

Figure B.2. Memory association module screen 58

Figure B.3. Settings and parameters screen 58

LIST OF TABLES

Table 2.1. Comparative place detection performances. 15

Table 3.1. Correspondence of detected places in the second tour with those of first tour in Fr site. 37

Table 3.2. Correspondence of detected places in the second tour with those of first tour in Sa site. 40

Table 3.3. Correspondence of detected places in the second tour with those of first tour in Lj site. 41

Table 3.4. Sssociation rates and maximum number of candidates. 44

Table 3.5. Correspondence of detected places in the first and second tours in North Campus site with Jaguar robot. 45

Table B.1. List of important classes and their explanation. 60

LIST OF SYMBOLS

a_i^k	Binary valued appearing state of i^{th} node at k^{th} base point
$\mathcal{A}^k$	Attribute set
b_i^k	Binary valued disappearing state of i^{th} node at k^{th} base point
c	Graph matching penalty term weight
c_k	Robot's position at k^{th} base point
C^{kl}	Pairwise distance matrix between two graphs
$\mathcal{C}(N)$	Subset of descriptors
$\mathcal{D}_m$	m^{th} detected place
E^k	Edge set
G^k	Region adjacency graph at k^{th} base point
$\mathcal{G}_m$	Segments summary graph of m^{th} detected place
$h(N)$	Tree height of node N
$\bar{I}_m$	Bubble space descriptor of m^{th} detected place
k	Segment merging threshold
$\mathcal{K}$	Index set of base points
M_{ki}	Value of i^{th} node at k^{th} base point in the existence matrix
$\hat{n}^{kl}$	Cardinality of $\hat{\mathcal{N}}^k$
N	A node of place memory
$\hat{\mathcal{N}}^k(l)$	A set of node correspondances between two graphs
N_r	A node to be recognized
$N^{\uparrow}$	Highest ancestor of node N
N_i^k	i^{th} node in k^{th} region adjacency graph
$\mathcal{N}^k$	Set of nodes in k^{th} region adjacency graph
$\mathcal{P}$	Index set of learned places
$s(N)$	Node signature of region adjacency graph node N
$s_1(N)$	The first attribute of node signature - centroid
$s_2(N)$	The second attribute of node signature - color

$s_3(N)$	The third attribute of node signature - radius
S_i^k	i^{th} segment in the k^{th} base point
$w = \left[\begin{array}{cccc} w_a & w_p & w_c & w_e \end{array} \right]$	The weight parameter vector
w_a	Area weight
w_p	Position weight
w_c	Color weight
w_e	Edge weight
x_k	k^{th} base point
$\mathcal{X}$	Base space
α_k	Robot's heading at k^{th} base point
$\beta(D_m)$	Place index of D_m
$\gamma(G_k, G_l)$	Distance between two graphs
$\gamma^B(N, N')$	Bubble space descriptor dissimilarity of two place nodes
$\gamma^S(N, N')$	SSG dissimilarity of two place nodes
δ	(Min.) Minimum segment size
φ^k	The coherency score
π^{kl}	Permutation matrix between two graphs
ρ_i^k	Coherency weight of i^{th} node in the k^{th} base point
σ	Segment smoothing factor
$\sigma(s_1(N))$	Positional stability of node N
τ_c	(Min.) Coherency threshold
τ_f	(Min.) Association threshold
τ_m	(Max.) Segment matching threshold
τ_n	(Min.) Place detection threshold
τ_p	(Min.) SSG existence threshold
τ_s	(Min.) Hybrid method minimum matches threshold
τ_w	Sliding window extension
Ω	A set of candidate nodes

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BD	Bubble Space Descriptor
BOW	Bag of Words
BS	Bubble Space
GUI	Graphical User Interface
MDS	Multi Dimensional Scaling Method
RAG	Region Adjacency Graph
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSG	Segments Summary Graphs
SURF	Speeded Up Robust Features
TSC	Topological Spatial Cognition

1. INTRODUCTION

This thesis is concerned with automated place detection as it pertains to place memory. A 'place' refers to a specific spatial unit or area such as 'X's office' or 'Y park' [1]. Place memory is critical if a robot is to be place-aware. This is where all related knowledge is retained for future referral. Using the knowledge stored therein, the robot can associate the incoming appearance data with past experiences or learn them as necessary. Appearance plays a key role in what this knowledge is about - particular if odometric data is not available or reliable. Each 'place' is defined as a collection of appearances or locations sharing common perceptual signatures or physical boundaries [2]. As such, place detection - namely determining the appearances belonging to each place - becomes critical.

1.1. Contributions

The contributions of this thesis can be summarized as follows:

- Place detection: A novel approach for detecting places is proposed. In this approach, scene content is represented by the segmented regions with their spatial relations encoded as a graph. Place boundaries are then detected using a coherency score which is calculated via tracking the segmented regions. Place detection is improved since prevailing segments tend to be more stable across different viewpoints and dynamic scene changes in contrast to local or global descriptors that have been previously used. This is attributed to the fact that even if they may encode a smaller port of the visual data, nevertheless this part is more prevalent in the respective appearances.
- Segments Summary Graphs (SSG): Detected places are represented by the prevalent segments and with their spatial relations on a graph structure called as 'Segments Summary Graphs ' . SSG encodes the spatial and temporal of properties of the segments found in the borders of the place. As such, it differs from previous

approaches in which such a representation can be obtained only after additional processing. Furthermore, resulting SSGs can be used as additional cues while associating with the place memory and thus make the decisions more reliable.

- Place Memory: Place memory and association are considered - using the places thus detected. This is done within the framework of the topological spatial cognition model that has been previously developed. Furthermore, the resulting segments summary graph representation is used to guide memory association and make it more reliable.

1.2. Outline

This thesis is organized as follows: In Chapter 2, the proposed place detection approach is presented and evaluated experimentally including a comparative study with a previously introduced approach to place detection based on bubble space representation. The coupling of place detection with place memory is explained in Chapter 3. This study includes an extensive experimental evaluation using benchmark data sets. The thesis concludes with a brief summary and comments regarding future directions. For completeness, the mathematical formulation of bubble space representation is presented in Appendix A. The user manual of the SSG software and place memory is given in Appendix B.

2. PLACE DETECTION

2.1. Introduction

Place detection enables the robot to decompose space into spatial units [3]. It is known that space decomposition can even be done in outdoors without any clear physical boundaries. The resulting spatial knowledge is thought to be more consistent with human's place concept [4]. As such, detecting places[1] is an integral capability - if robots are to become spatially aware. Appearance plays a key role in place detection - as geometric or odometric data may not be always available. The key motivation is that visual data from a single location will not encode all the place related knowledge. Therefore, appearance data that is collected through a place detection is used to describe the place knowledge.

Formally, the problem can be defined as follows: Consider a robot that has navigated through a sequence of base points $x_k \in \mathcal{X}$ with $k \in \mathcal{K}$ denoting the index. Each base point $x_k = \begin{bmatrix} c_k^T & \alpha_k \end{bmatrix}^T$ is defined such that $c_k \in R^2$ denotes its planar position and $\alpha_k \in S^1$ is its heading. 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_k will not be known explicitly - as is assumed here. Transition from one place to another may occur in several forms - including passing through a door or a gate, traveling straight through a corridor or street (when the visual content changes in sideways) and rotating around something.

2.2. Related Literature

While, place detection was done through manual annotation, more recently, automated approaches are being increasingly used. In most, the problem is commonly

[1]Note that with some appearance-based SLAM or some topological approaches, each appearance and thus each location is defined as a distinct place [5, 6]. As such, place detection is not required in these systems.

formulated as detecting scene discontinuities - assuming the appearance data to have temporal nature. The consistency of the appearances is tracked with discontinuities signaling transitions among places. Approaches vary in the type of descriptors used in comparing appearances. Global descriptors encode the image as a whole - without extracting any local information such as intensity differences [7], histogram [8], optical flow [9] or GIST [10]. While these descriptors are simpler to process, they tend to be sensitive to appearance variations due to viewpoint changes or dynamic entities - as scene contents are not considered individually [11]. Alternatively, local descriptors such as SIFT or SURF features [12–14] describe relevant local features or landmarks. However, they provide low-level scene information and matching large number of local features in a stable manner can be inefficient. There are also hybrid approaches that encode local features using global schemes such as bag-of-words [15] and bubble space [16]. While place detection performance is comparably improved, it can still run into problems - as scene contents are considered only at the low-level. Alternatively, there are also approaches in which places are detected via specifically detecting boundaries such as passages or doors [4]. As such, the methods rely on the training and performance of such detectors. While a number of recent studies suggest the role of higher-level scene contents in defining places [17], none of previous work considers them in detecting places.

In this work, we consider appearance-based place detection from this perspective and introduce a novel approach based on the prevailing segments. As is well known, segments encode perceptually similar pixels and thus constitute an intermediate level of representation in which an image is decomposed into meaningful regions on the basis of some similarity measure for ensuing higher-level scene analysis. Hence, they tend to be more stable across different viewpoints and dynamic scene changes. As such, they are believed to play a fundamental role in many vision related tasks - including the generation of content-based video summaries [18–20] and object recognition [21]. This has been a motivation for the proposed approach as well.

2.3. General Approach

Place detection is based on partitioning of the set $\mathcal{K}$. Let the partition be denoted by $\{D_1, ..., D_{m^*}\}$ with subsets indexed by $\mathcal{D} = \{1, \ldots, m^*\}$. Each subset $D_i \subset \mathcal{K}$ corresponds to one distinct 'detected place'. As the robot navigates to different base points, the index set $\mathcal{D}$ expands. The proposed approach consists of four stages as shown in Figure 2.1:

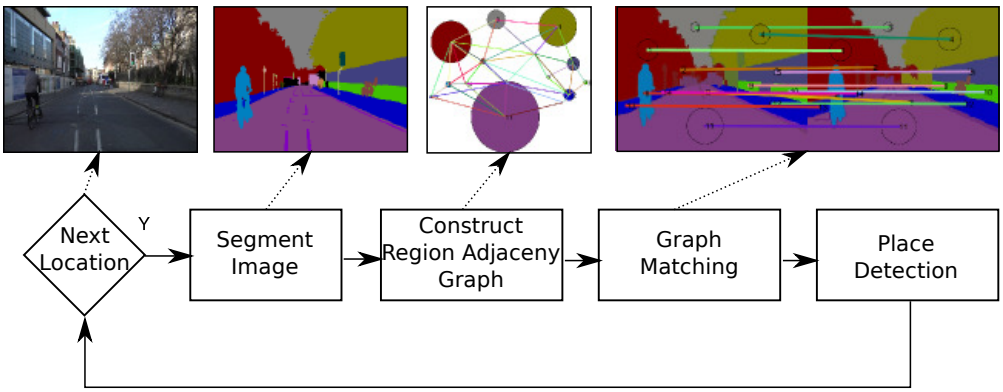


Figure 2.1. SSG based place detection.

(i) First, the incoming visual data (image) from each base point x_k is segmented.

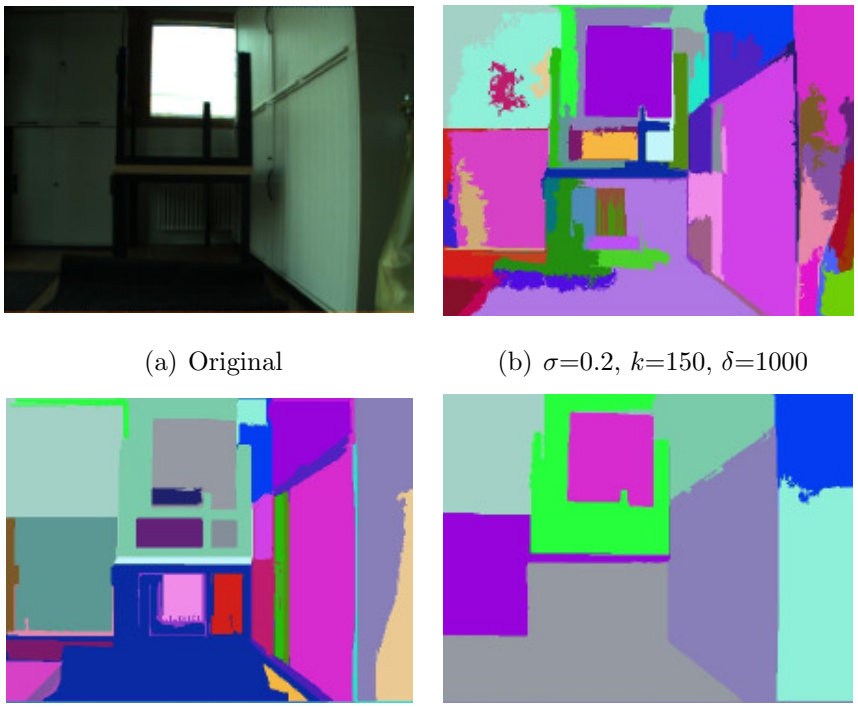
(ii) Next, a region adjacency graph (RAG) is constructed using the segmented image. A RAG expresses the image segments and their spatial relations as nodes and edges respectively [22].

(iii) The next stage is to match the newly formed RAG with those associated with the preceding base points as to identify nodes (segments) that continue to exist.

(iv) In the fourth stage, places are detected via the partitioning of the set $\mathcal{K}$ based on the spatio-temporal coherency of associated RAGs. Each resulting cluster is viewed as being associated with one distinct place. The prevailing segments are used to construct a segments summary graph (SSG) of the detected place.

2.4. Segmentation

The first stage is segmenting the incoming visual data (image) from base point x_k into n^k homogeneous color regions $\mathcal{S}^k = \{S_i^k\}_{i=1}^{n^k}$. There are two requirements.



(a) Original (b) σ=0.2, k=150, δ=1000

(c) σ=0.7, k=150, δ=1000 (d) σ=0.7, k=150, δ=5000

Figure 2.2. Segmentation results for different sets of parameters

First, the segments should be consistent across consecutive frames. As such, it will be possible to track them temporally. Second, the computational complexity of the algorithm should be minimal as possible. Thus, it will be possible to apply it on a robot. Fortunately, segmentation is a well-studied area in the image processing and computer vision communities with the developed algorithms being used extensively [21]. As such, any off-the-shelf segmentation method that satisfies the two requirements can be utilized. In this work, we use the graph-based segmentation algorithm [23]. It is known to be one of the best performing algorithms [24]. The segmentation is based on a predicate for measuring the evidence for a boundary between two regions and is computationally efficient ($O(n \log n)$ time for n image pixels) and thus can be run at video rates in practice. In this algorithm, the number and size of the segments generated depend on three parameters: smoothing factor σ, merging threshold k and minimum segment size δ. The parameter σ is associated with the Gaussian filter that is used to smooth the image as to compensate for digitization artifacts. In practice, $\sigma \cong 0.7 - 0.8$ has been observed to remove such artifacts without producing any visible change in the image. The parameter k sets a scale of observation, in that a larger

k causes a preference for larger segments. Note, however, that k is not a minimum segment size. Smaller segments are allowed when there is a sufficiently large intensity difference between neighboring segments. Finally, segments having size smaller than δ are pruned out. These parameters are carefully tuned as to have segments that are as large and few as possible while retaining their consistency. They are set as $\sigma = 0.7$, $k = 150$ and $\delta = 1000$.

2.5. Region Adjacency Graphs

The next stage is forming the RAG of the incoming visual data (image) at each base point x_k. The segmented regions and their relationships are represented by the nodes and the edges of a region adjacency graph G^k. Each RAG is an attributed graph that consists of $G^k = (\mathcal{N}^k, E^k, \mathcal{A}^k)$ where $\mathcal{N}^k$ is the set of nodes, E^k is the edge set and $\mathcal{A}^k$ is the attribute set that contains attributes related to vertices N_i^k and E_{ij}^k. Each segment $S_i^k \in \mathcal{S}^k$ is associated with a node N_i^k. As such, the cardinality fo $\mathcal{N}^k$ is n^k - namely the number of segments as defined in the previous section. If two segments S_i^k and S_j^k have common borders, then an edge relation E_{ij}^k between the respective nodes N_i^k and N_j^k exists. Each node N_i^k is associated with a node signature $s(N_i^k)$. The node signature $s(N_i^k)$ is a vector of varying dimension that encodes node and edge attributes. Node attributes are the centroid ($s_1(N_i^k) \in \mathbb{R}$), color ($s_2(N_i^k) \in \mathbb{R}$) and radius ($s_3(N_i^k) \in \mathbb{R}$) of each node. They are determined based on the respective segment. Edge attributes ($s_4(N_i^k, N_j^k) \in \mathbb{R}$) are set as inversely proportional to mean color difference between two segments S_i^k and S_j^k. The top three images in Figure 2.1 illustrate RAG construction. For visualization purposes, the position, color and radius of nodes represent the center of mass, mean color and total area of segments respectively.

2.6. RAG Matching and Node Existence Matrix

The second stage is to match a newly formed RAG G^k with preceding RAGs G^l, $l < k$ that are associated with the previous base points within a window of size

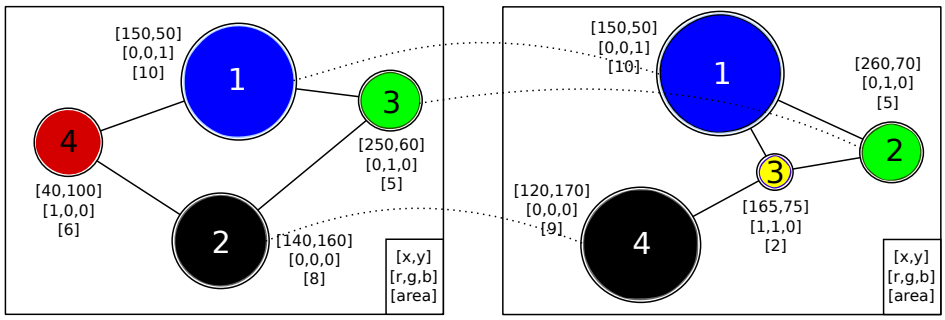


Figure 2.3. Matching two RAGs.

τ_w. This enables the robot to identify nodes (segments) that have appeared previously as well as determining newly appearing segments. Their labels can then be assigned accordingly. The graph matching algorithm is a modified version of matching based on node signatures [25]. As such, the distances between every pair of nodes of G^k and G^l can be computed. Let C^{kl} be the corresponding distance matrix with elements C^{kl}_{ij} defined based on weighted Manhattan distance between the node signatures as:

$$C^{kl}_{ij} = \left\| s(N^k_i) - s(N^l_j) \right\|_w \tag{2.1}$$

The weight parameter vector $w = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix}^T$ affects how the different node attributes weigh in. Their values are set manually and remain unchanged throughout the robot's operation. For instance, increasing the position weight w_1 may increase the accuracy for steady scenes however it degrades matching score in dynamic scenes. The color weight w_2 and area weight w_3 are also affected by the nature of segmentation. In coarse segmentation, segments are likely to enclose distinct objects. In addition, illumination and shading are also quite influential as expected. Due to these reasons color weight parameter is chosen relatively small. On the other hand, the area weight w_3 is relatively larger since prominent objects are likely to be larger in size.

The distance matrix C^{kl} is used as the basis for RAG matching. We use the simple Hungarian algorithm [26]. The resulting permutation π^{kl} defines one-to-one optimal matching between the nodes of two RAGs. However, in practice, some of these assignments may lead to wrong matches with high costs. In order to minimize

such cases, only assignments $\pi^{kl}(i)$ with matching cost $C^{kl}_{i\pi^{kl}(i)}$ less than the segment matching threshold τ_m are considered to be valid matches:

$$\hat{\mathcal{N}}^k(l) = \left\{ i \in \mathcal{N}^k \mid C^{kl}_{i\pi^{kl}(i)} < \tau_m \right\} \tag{2.2}$$

Let the cardinality of $\hat{\mathcal{N}}^k$ be denoted by $\hat{n}^{kl}$. The parameter τ_m is set manually depending on predefined correct and false matches. The example in Figure 2.3 shows the matching of two RAGs. While both consist of four nodes, the nodes differ in their attributes. It is observed that only three of the nodes are matched.

Next, we define the distance between two RAGs G^k and G^l as follows:

$$\gamma(G^k, G^l) = \frac{1}{\hat{n}^{kl}} \sum_{i \in \hat{\mathcal{N}}^k} C^{kl}_{i\pi^{kl}(i)} + c\left|n^k - n^l\right| \tag{2.3}$$

The first term is simply the average of the cost of matching while the second term penalizes if the number of nodes differs as weighted by the parameter $c \in R^{>0}$. The parameter is defined to be cost value per node - namely $c = \frac{1}{n^k} \sum_{i \in \hat{\mathcal{N}}^k} C^{kl}_{i\pi^{kl}(i)}$. The result

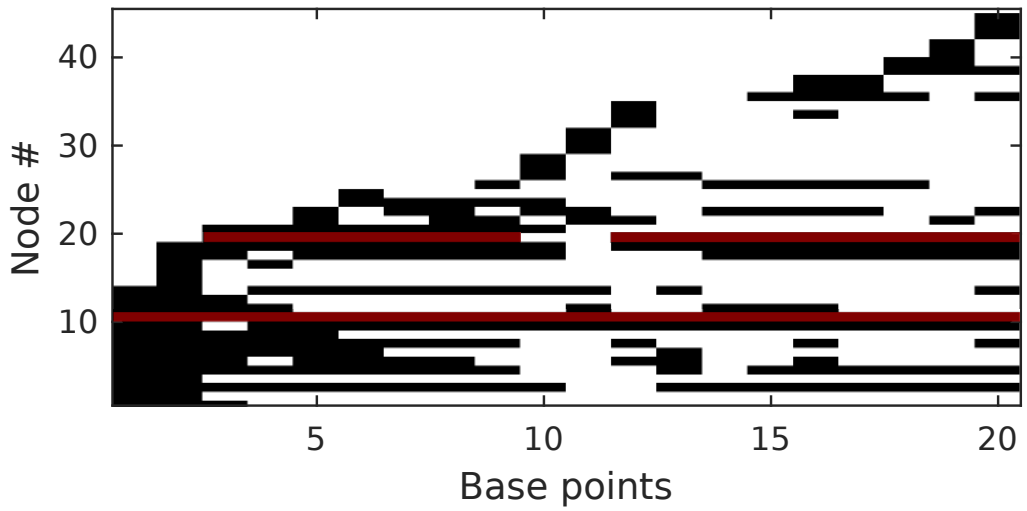


Figure 2.4. Node existence matrix. ($10th$ and $20th$ nodes are shown in red)

of the matching for each RAG across the base points $\mathcal{K}$ is maintained as a dynamic matrix $\mathbf{M}$ with integer valued segment index entries - referred to as node existence

matrix. The matrix entry M_{ki} will be equal to j if the node N_j^k is matched to the segment N_i^{k-1} that has appeared in the previous RAG. In case of no match with the previous entry, the search can be extended further to look for possible matches in the last τ_w base points. Otherwise, the node is added as a new y-axis entry. As such, nodes and thus segments can be tracked even after a short disappearance. The node existence matrix evolves as the robot navigates. Hence, it enables the tracking of the nodes throughout previous RAGs up to current. In the example of Figure 2.4, non-zero entries are shown in black and correspond to the matched segments. It is observed that node#10 has appeared throughout whole sequence. This is in contrast to node#20 that first appeared in the 3^{rd} base point, disappeared in the 10^{th} and then appeared again.

2.7. Place Detection

The partitioning is an iterative process as summarized in Figure 2.5. It is guided by two assumptions: The first is that contents of consecutive base points taken from a particular place will be coherent which implies that the associated RAGs will be similar. The second is that different places will be divided by transition regions which are characterized by high incoherency. As such, each detected place is defined by a maximal set of base points that have a coherent RAG structure.

The coherency φ^k of each RAG G^k measures the number of emerging and disappearing nodes within a sliding window of extension τ_w in the node existence matrix. It considers the segments appearing in the last τ_w RAGs:

$$\varphi^k = \sum_{l=k-\tau_w}^{k} \sum_{i=1}^{|n^l|} \rho_i^l (a_i^l + b_i^l) \tag{2.4}$$

where

$$a_i^l = \begin{cases} 1 & \text{if } M_{li} > 0,\ M_{l-1,i} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

$$b_i^l = \begin{cases} 1 & \text{if } M_{li} = 0, M_{l-1,i} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

$$\rho_i^l \propto \sigma(s_1(N_i^l))^{-1} + s_3(N_i^i) + \sum_{l=k-\tau_w}^{k} a_i^l \tag{2.7}$$

Each node is weighted by ρ_i^l depending on its positional stability $\sigma(s_1(N_i))$, area $s_3(N_i)$ and continuance across consecutive base points as measured by the last term in Equation 2.7. The weights are updated accordingly at each base point. The values of sliding window extension τ_w and place detection threshold τ_n and are set manually depending on the frame rate, video resolution and segmentation parameters. If the frame rate is high, these parameters are tend to be high in order to encode enough spatial information from the environment. For example, if a data set that is acquired indoors with 50cm between consecutive base points, the parameter τ_w is set in the range 20-30 and τ_n is set in the range 5-10 respectively. For the real robot experiments with frame rate of 15 frames per meter, τ_w and τ_n are set as 50 and 10 respectively. In future, we plan to develop an approach that will consider their automatic adaptation based on the incoming visual data.

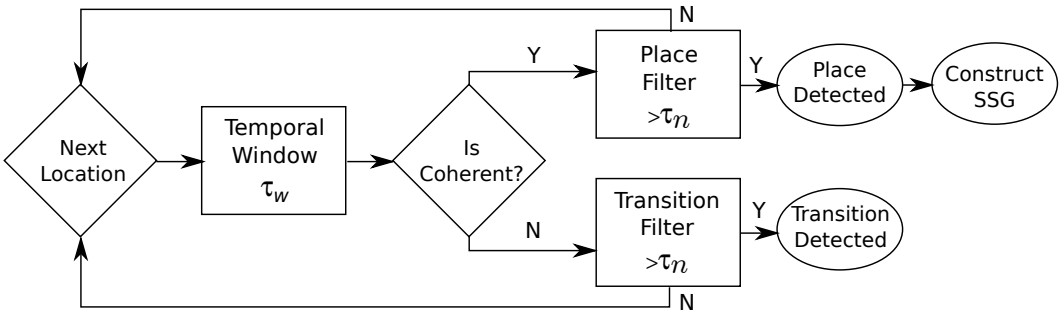


Figure 2.5. Place detection methodology.

The coherency score φ^k is used in deciding whether to start a new place detection, to continue with it or to end the current detection. A new place detection is initiated if coherency score is greater than coherency threshold τ_c consecutively τ_n times while the current place detection ends if it is below the threshold τ_n times. Otherwise the current detection continues. Base points in which there is no detection under progress are referred to as transition regions. Because the coherency score varies φ^k between [0,1] the parameter τ_c is set as 0.5.

2.8. Place Representation and Segments Summary Graphs

While a place is being detected, the incoming appearances are encoded by a set of descriptors. We use two types of descriptors: Segments Summary Graphs (SSG) and Bubble Descriptors (BD). Place descriptors evolve in time and are calculated incrementally as the robot navigates to new base points.

A SSG is an intermediate level descriptor that encodes the coherent nodes and edges observed in the place. The content of the SSG descriptor is constructed based on the spatio-temporal properties of the nodes and the edges as inferred from the associated node existence matrix. As explained in the previous section, the node existence matrix stores all the nodes with their spatial properties such as a position, area and edge relations as well as with their temporal properties such as when they are appeared and disappeared. In order to determine which nodes and edges are to be included, each coherent region in the node existence matrix is considered and the nodes and edges which appear longer than τ_p percent of the associated based points of that place are selected as the candidate summarizing segments. Furthermore, segments with the area smaller than τ_a are filtered out. The τ_a value is determined based on acceptable segment size and is usually set as 5% of the total image size. Finally, the selected nodes and edges associated with the detected place $D_m \in \mathcal{D}$ is represented on graph referred to as Segments Summary Graphs. The similarity of two detected places m and n is

based on their SSG similarity:

$$\gamma^S(m,n) = \gamma(\mathcal{G}^m, \mathcal{G}^n) \tag{2.8}$$

Bubble Space Descriptors[2] (BD) are hybrid descriptors. Previous work has shown its comparative advantages to other representations such as preserving the relative S^2 geometry of visual features, being rotationally invariant and incorporating any number of observations. However, the proposed model is in no way dependent of this particular choice and thus can be used with any other kinds of descriptors. Similar to SSG descriptors, each detected place D_m is represented by the corresponding set of BD descriptors $I(x_j), j \in D_m$. The mean descriptor $\bar{I}_m$ is defined as:

$$\bar{I}_m = \frac{1}{|D_m|} \sum_{j \in D_m} I(x_j) \tag{2.9}$$

The similarity of two places based on their similarity is measured by $\gamma(N, N')$:

$$\gamma^B(m,n) = |\bar{I}_m - \bar{I}_n| \tag{2.10}$$

2.9. Experimental Results

In this section we report our experimental results. First, the proposed approach is evaluated in the context of video summarization problems - including a comparative study on the Open Video Project dataset [27]. Next, we consider place detection using benchmark datasets. Finally, we consider real-time application with a mobile robot.

[2]For the interested reader, they are explained briefly in Appendix 'Bubble space'.

2.9.1. Video Summarization Results

In this section, we compare the place detection performance of proposed approach SSG with previous approaches that have been primarily proposed for video summarization. In particular, we consider OVP [27], STIMO [28], VSUMM [29] and OnMSR [19]. The comparison is done using the Open Video Project dataset [27] using results as presented in [19]. The following points need to be noted: First, the evaluation is based on manual annotation of places by a human user (US). Second, differing from the proposed approach, each detected place is not encoded by a SSG. Rather, it is shown by a selected key frame. In order to be comparable, we select the visual data associated with the most coherent RAG in each detected place. For example, for the fifth video in this dataset, the obtained keyframes are as shown in Figure 2.6. It is observed the number of places detected by the first two methods are rather short in comparison to user detected places. As such, different places are merged and seen as one distinct place. On the other hand, the places detected in our approach together with VSUMM and OnMSR approaches are closest to those manually obtained. We then evaluate place detection performance based on three metrics including precision, recall and F-score as defined in [19]:

$$\text{Precision} = \frac{n_{mAS}}{n_{AS}} \; \text{Recall} = \frac{n_{mAS}}{n_{US}} \tag{2.11}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.12}$$

Here, n_{mAS} is the number of keyframes associated with each detected place that are visually similar with those of manual detection, n_{AS} is the total number of places detected and n_{US} is the total number of manually detected places. Two images are visually similar iff the visual content is similar as determined by manual inspection and their image index difference is at most 60 frames. Precision measures how well the

detected places are in agreement with those that are manually annotated. On the other hand, recall measures how much of manually detected places are covered. F-score is an effective metric that balances the precision and the recall scores. Places are detected reliably if all distinct places (as specified by the user) are all detected as compactly as possible. The results are presented in Table 2.1. As expected, the first two methods perform the worst. The other methods vary in the performance. For example, while OnMSR has the best precision performance, VSUMM is better in regards to recall. Finally, our proposed approach SSG has both highest precision and near-highest recall rates. As such, its F-score is the highest.

Table 2.1. Comparative place detection performances.

Algorithms	Precision (%)	Recall (%)	F-score (%)
OVP	43	64	51.4
DT	47	50	48.5
STIMO	39	65	48.8
VSUMM	42	**77**	54.4
OnMSR	50	66	56.9
SSG	**56**	75.9	**64.4**

2.9.2. Place Detection Results

Experiments are done using benchmark data from the indoor Freiburg (Fr), Saarbrucken (Sa) and Ljubljana(Lj) sites under cloudy illumination and outdoor New College (NC) site. In RAG matching, the weights vector is set as $w = \begin{bmatrix} 0.8 & 0.5 & 0.3 & 0.1 \end{bmatrix}$ and $\tau_m = 0.05$. As such, the position and area similarity of segments are relatively weighted more in comparison to color and edge attributes. Finally, for the experiments on data sets, the parameter τ_w is set in the range 20-30 and τ_n is set in the range 5-10 respectively. For the real robot experiments, frame rate is 15 frames per meter and respective values of τ_w and τ_n are selected as 50 and 10.

The Fr site is associated with perspective camera data from 1911 base points acquired from a path of 40m. The robot detects 7 places as as seen in Figure 2.7(a). It

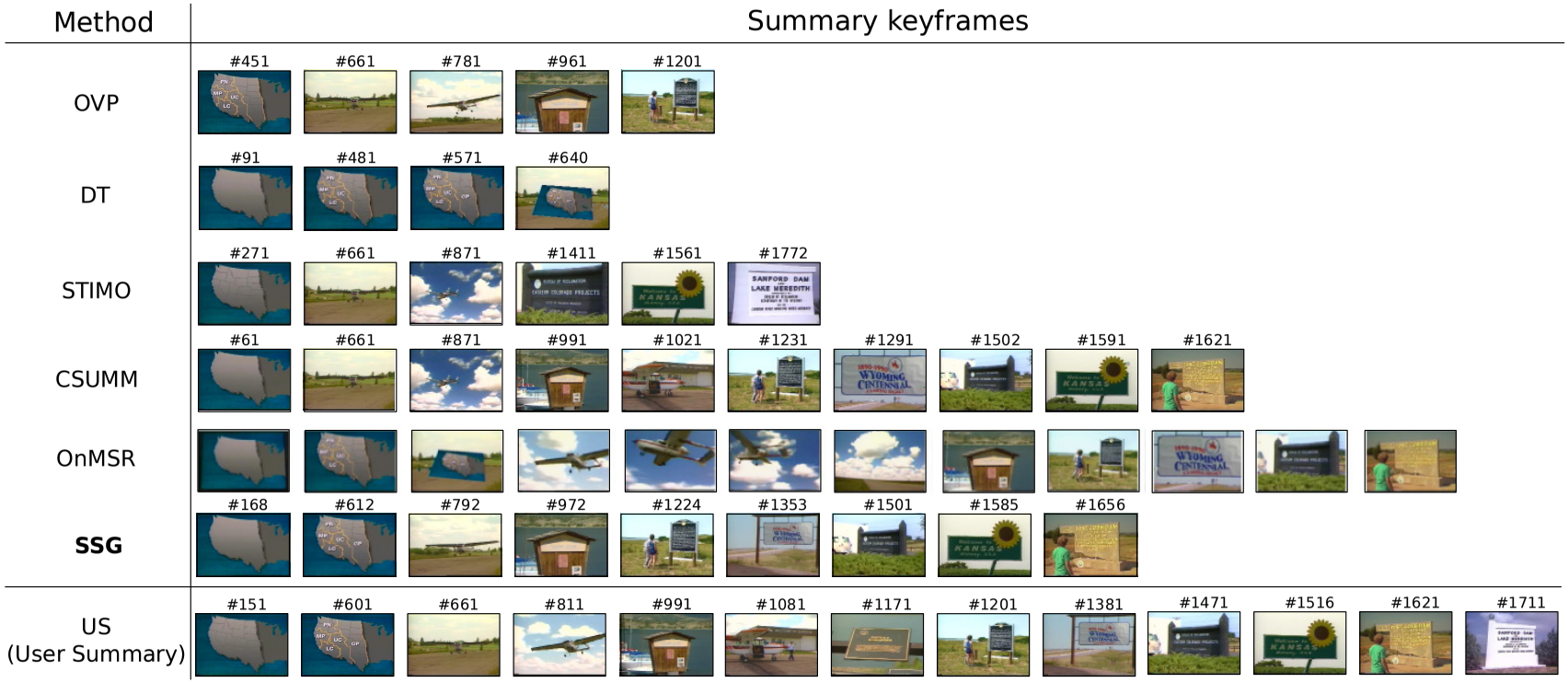
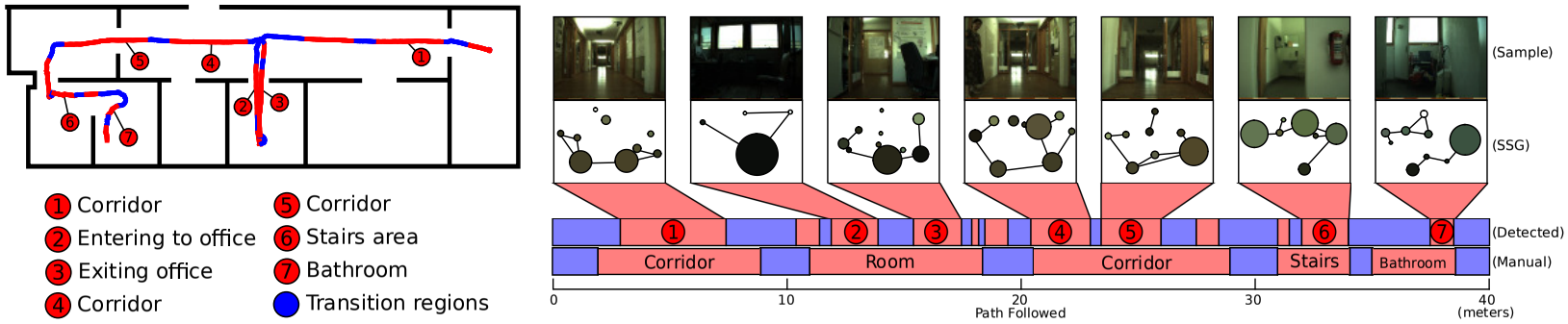


Figure 2.6. Comparative place detection results.

is observed that the robot is able to detect physically separated places (i.e. by door) as distinct places. Sample appearances from each detected place and the comparison of these places with those obtained via manual annotation are also shown as seen in Figure 2.7(b). Interestingly, some of the transition regions are wrongly found to be a part of detected place. Closer inspection reveals that this occurs because the separating walls or doors are transparent so that the robot sees the other place. Furthermore, we also observe that an indoor spatial unit may be detected as multiple places - depending on robot's trajectory and motion. For example, two distinct places are detected as the robot enters a room and turns in the room to exit it. A similar case occurs if the robot's camera moves abruptly in a room so that coherency of the visual contents is lost. The SSG of each detected place is as shown in the same figure. It is observed that the number of prevailing segments in each detected place is at most 10. These segments correspond to the continously observed regions such as floor, ceiling and walls as well as corresponds to contexual objects such as windows, chairs and doors which can only be observed in particular places.

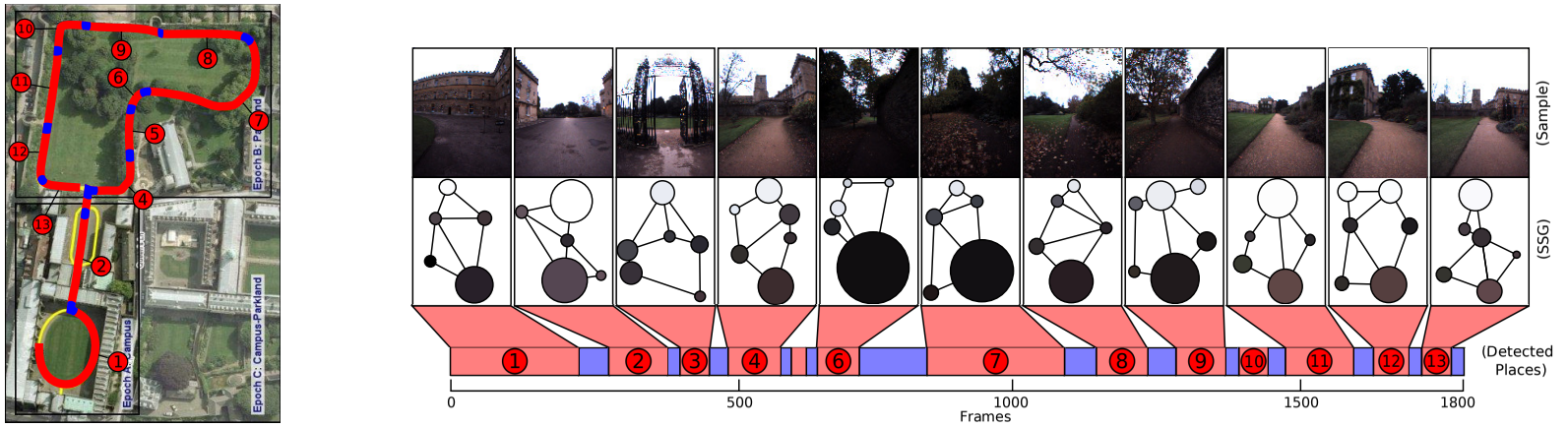
In the NC site, the robot travels along a path of 550m and collects visual data from a perspective camera from 1800 base points. Let it be noted that places in outdoors settings may not obvious even to the human users - as scene content may change gradually. In other words, places are not always separated physically by transition regions. The robot is able to detect 13 places as shown in Figure 2.8(a). Sample appearances from each place and the comparison of these places with those obtained via manual annotation are also shown as seen in Figure 2.8(b). It is observed that passage areas such as gates are detected correctly. For example, a gate separates 1^{st} and 2^{nd} places. A similar situation holds for 3^{rd} and 4^{th} places. Furthermore, street corners are detected as transition region due to the rotation of the robot's camera. For example, this is the case for places 8 and 9. Similarly, the robot detects places 11 and 12 distinctly - even if the appearance change is slow. The resulting SSGs are also shown in the same figure. Close inspection reveals that the encoded content does indeed capture the appearances from that place. For example, SSGs associated with 6^{th} and 7^{th} places are observed to have as one big dark and a small white segment.



(a) Detected places (red) overlaid on the Fr site with transition regions (blue) also indicated.

(b) The evolution of place detection wrt the incoming appearance data along 40m. For each detected place, a sample scene and the respective SSG are as shown. Places as detected manually are also shown.

Figure 2.7. Place detection as the robot navigates along a short path in Fr site.

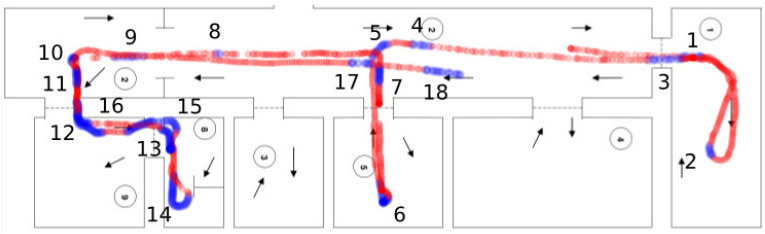


(a) Detected places (red regions) overlaid on the NC site map with transitions (blue) also indicated.

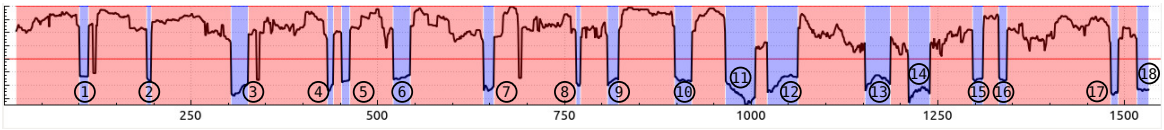
(b) The evolution of place detection wrt the incoming appearance data along 550m. For each detected place, a sample scene and the respective SSG are as shown. Places as detected manually are also shown.

Figure 2.8. Place detection as the robot navigates through NC site

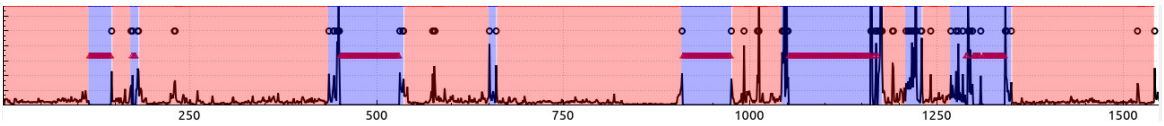
This is quite expected as the respective appearances are nearly dark with sky showing. On the hand, SSGs of 11^{th} and 13^{th} places contain a couple of segments that encode the scene entities such as the sky, vegetation or road.



(a) SSG based detected places in Fr site as numbered on the map



(b) SSG based detected places

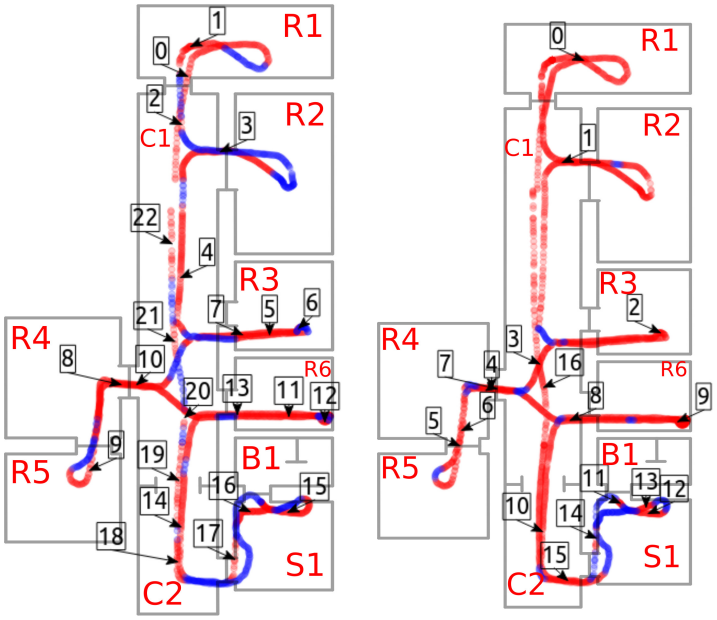


(c) BD based detected places

Figure 2.9. Comparative evolutions of coherency in Fr site. Detection is based on coherency score in SSG method. In BD based approach, red triangles indicates uninformative and black circles shows frames with dissimilarity score greater than the threshold.

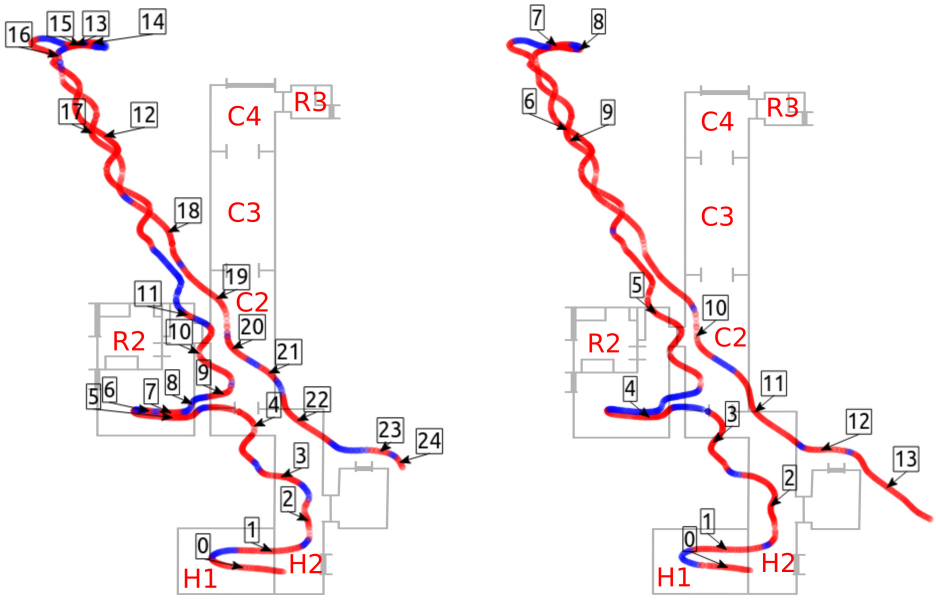
Next, we compare the places as detected by our approach (SSG) with a previously introduced place detection approach (BD) [3]. In this approach, place detection is done in a similar manner using a dissimilarity score. Nevertheless, there are some differences: appearances are compared using the bubble descriptors [30] and sensory data reliability is ensured via checking for informativeness.

First, we present detailed comparative for a longer path in the Fr site in Figure 2.9. It is observed that there are more places detected in the SSG approach. This is attributed to using segments - rather than a hybrid descriptor. As such, sudden changes in coherency score signal transition regions more reliably. It is observed that



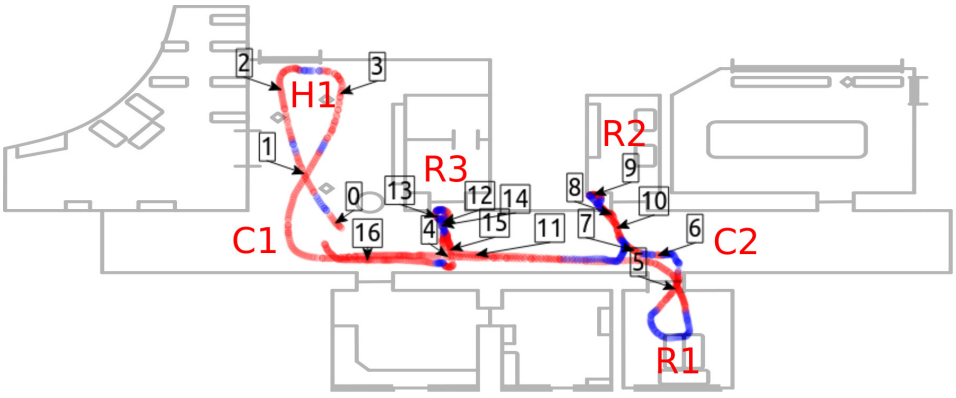
(a) Fr Site - SSG (b) Fr Site - BD

Figure 2.10. Comparative place detection results for Fr site

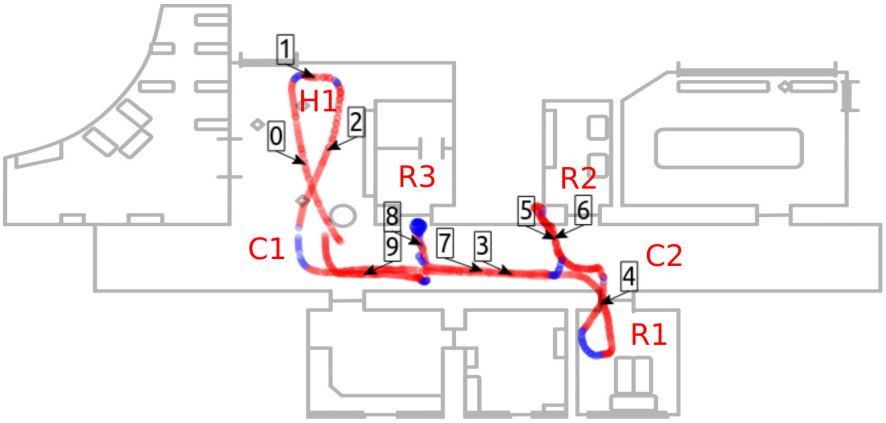


(a) Lj Site - SSG (b) Lj Site - BD

Figure 2.11. Comparative place detection results for Lj site



(a) Sa Site - SSG



(b) Sa Site - BD

Figure 2.12. Comparative place detection results for Sa site

coherency drops sharply and remains low during the transitions. This is in contrast to BD approach where the dissimilarity score is relatively more unstable(i.e. there are short term peaks only during transitions) and the exact value of the threshold is hard to be determined. Moreover, some transition regions couldn't be detected, such as in frames 1000 and 1500, because the the dissimilarity is low and cannot be observed through enough number of frames. Next, we consider extended routes in the Fr, Lj and Sa sites.

We consider extended routes in the Fr, Lj and Sa sites - again under cloudy illumination. It should be noted that the odometric data that is used only for presentation purposes is not reliable along some parts of these paths. The deviation is most evident in the Lj site. The results are presented in Figure 2.10, 2.11 and 2.12. As seen in Figure 2.10(a) and 2.10(b), there are 23 detected places with SSG while this number is 17 for BD. When the robot enters room R1, it does not detect the transition in both cases. However, the transition is detected with SSG as the robot exits the room. The transisition is detected with both of the approaches at the entrance of room R2. However using BD room R2 and a part of the corridor C1 are merged as one place (place 1). In contrast, with SSG the room R2 (place 3), and corridor C1 (place 4) are detected as distinct places. SSG detects room R3 as two places (place 5 and 7) whereas BD detects as one place (place 2). However the content changes relatively fast due to rotation of the robot in this room and it is expected to be detected as two distinct places. Room R4 and R5 are detected as distinct places, (place 8 and 9), respectively in SSG approach. BD approach also detects two places but the transition between R4 and R5 is not detected at the correct location. Similarly, the entrance of room R6 is detected by SSG but not detected by BD. Furthermore, corridor is merged to room R6. Corridor C2 is detected in both of the approaches; however it is separated into two in SSG because of the door appeared on the way. As the appearances coming from stairs S1 and bathroom B1 areas are rather complicated, they are detected as transition regions most of the time.

Detected places in Lj site are illustrated in Figure 2.11(a) and 2.11(b), respectively. In this case, there are 25 detected places with SSG while this number is 14 for BD. We notice that the number of places detected in corridors C2, C3 and C4 differ. This is due to doors that appear due to the zig-zag type of motion. In hallways H1 and H2, we obtain similar detection results. In room R2, transition is detected by both of the approaches but the room is split into three places in SSG whereas BD detects most of the region as transition. We also obtained similar detection in regions C4 and R3. Similarly, SSG approach detects more places in Sa site as shown in Figure 2.12(a) and 2.12(b), respectively. Transitions between corridor C1 and H1 are detected by SSG but BD merges two regions. Corridors are detected reliably in two approaches. In room R3, there are two sub regions and they are detected as distinct places (place 12-15) in SSG however BD detects whole room as one place (place 8). In summary, it is observed places are detected more reliably as places can be differentiated even if their transition is gradual. As such, SSG approach is also likely to generate longer transition regions - if the coherencies of the respective RAGs are low. In addition, the robot is able to simultaneously generate the SSG which describes the detected places based on the prevailing segments and their spatial relations.

2.9.3. Experiments with Jaguar Robot

The last set of experiments are done with our Jaguar robot shown in Figure 2.13. As visual sensing (both the hardware and the acquisition geometry) is different from the first set of experiments, the parameters are adjusted accordingly as follows: $\tau_w = 50$, $\tau_n = 10$ and $\tau_m = 0.01$. In the first tour, the robot is teleoperated to follow a path of approximately 450 meters as shown in Figure 2.14(a) and collects data at 7484 base points. Using the proposed approach, it detects 28 places in total as shown in the map. Detected places are depicted in a linearized format for comparison purposes as shown in Figure 2.14(b). The tour starts from indoor corridor area as depicted in a green dot and the whole area is detected as one place. Then, the robot visits vegetation area (place 3-10) passing through a car parking area. It is observed that, most of the transitions (as illustrated in blue) in the vegetation area occured during passing through corners

and the remaining areas are detected as one place (places 3, 7 and 10). Next, the robot travels through a car parking area and the whole region is detected correctly as one place (place 11). Then, the robot follows a path that contains vegetation and buildings and detects two places in total (place 12 and 13). At the end of the path, the robot takes a tour around the hall entrance and detects two places (place 14 and 15) as expected. In the return path, the robot detects two places in the same path. However, in the car parking area, the robot detects three places (places 20-23) in contary to the first time visit where it was detected as one place. This is probably due to zig-zag type motion of the robot in the return part. Then, the robot enters to the building again. Inside the building, three places are detected where the first two correspond to the entrance and corridor and the last corresponds to the laboratory area. In most of the cases, it is observed that the transition regions are detected accurately. Although most of the detected transitions are the ones which originates from the rotational motion of the robot, gradual transitions are also detected in most of the cases.

In the second tour, the robot is again teleoperated to follow a similar path at another time. The robot collected data at 8077 basepoints and detected 30 places in total as shown in Figure 2.14(c). It is observed that most of the detected places overlap with the first tour and the total number of detected places are very close. However, the robot detected relatively larger number of places in some regions, especially between places 17-24 due to dynamical factors such as walking people and moving cars observed along the path.

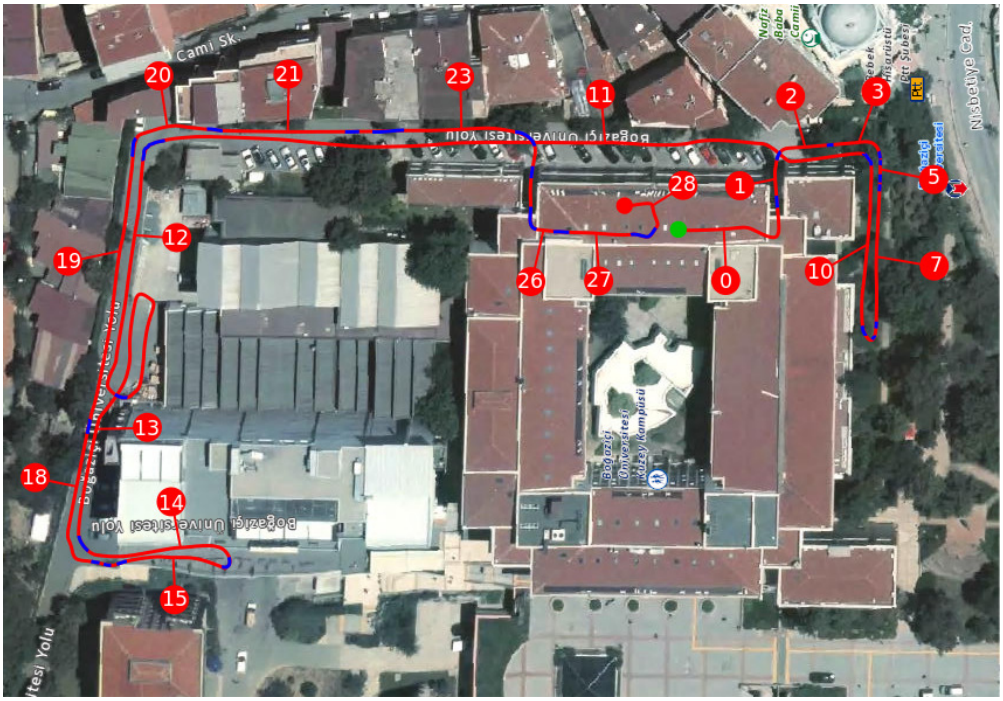
The computational performance of the robot is also analyzed. The per frame processing times vary depending on the spatial cognition activities of the robot and the size of the place memory. In the place discovery mode, the processing time is found to be in the range of 250-400 ms as shown in Figure 2.15. If the extent is infinite, then processing time would increase as seen in this figure. It should be noted that our proposed approach is designed to be scalable for long life operation and per frame processing times is expected to be constant and upper bounded. The slight increase in the processing time is mainly due to two reasons: First, although only the last τ_w base

points are processed at each time, we store all past node information in the memory for debugging purposes and that causes memory read and copy operations to take longer time. Second, graphical user interface is used for debugging purposes only and it is not optimized.

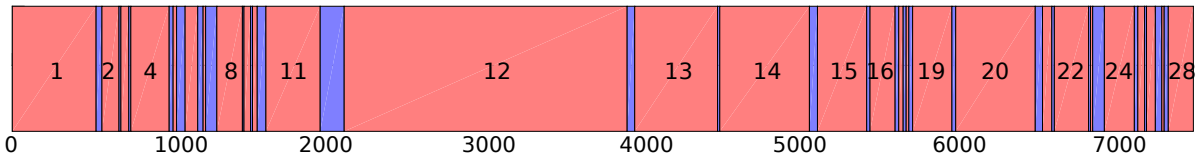
Figure 2.13. Jaguar robot

2.10. Conclusion

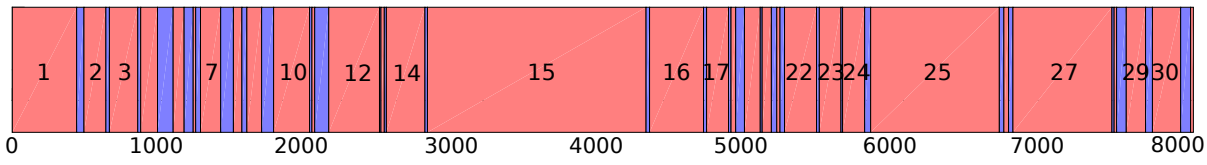
In this chapter, we introduce a novel approach to appearance based place detection based on the prevailing segments. Our motivation is that segments encode the scene contents at an intermediate level of representation while being relatively stable under a wider range of viewpoints and dynamical changes - differing from global, local or hybrid descriptors. Each incoming appearance is first segmented and larger segments along their spatial relations are represented by a regions adjacency graph. Places are detected via tracking the coherency of region adjacency graphs across the incoming appearance data. As such, place detection can be done more reliably while simultaneously generating a segments summary graph for each place that can be used in ensuing the semantic analysis of the place. It is observed places are detected as places can be differentiated even if their transition is gradual. The possible extension



(a) SSG based detected places in North Campus site as numbered on the map



(b) Detected places in the first tour



(c) Detected places in the second tour

Figure 2.14. Places detected by the Jaguar robot in North Campus site.

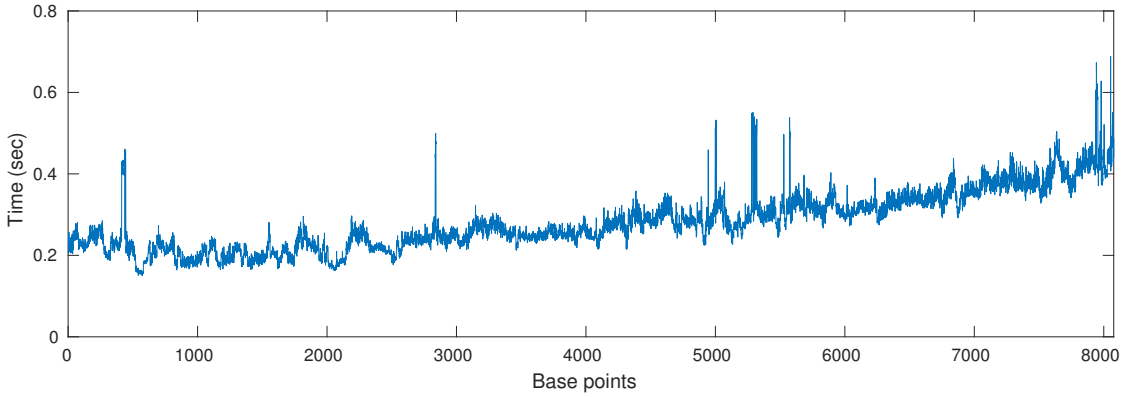


Figure 2.15. Performance analysis: Processing time per frame.

to current approach would be to use SSG representation for looking at the scene in detail and recognizing the objects contained therein.

3. INTEGRATION WITH PLACE MEMORY

3.1. Introduction

Once a place is detected as D_m with $m \in \mathcal{D}$, the robot attempts to associate it to past experience it via relating to its long-term place memory. Place memory retains the knowledge of places. The purpose of the memory association is to decide if the detected place is one of the previously visited places. If the association is narrowed down to a single place, then it is referred to as maximal association. Such an association implies that the robot has maximal familiarity with its surroundings.

3.2. Place Memory

Place memory is built using previously proposed model [31]. Suppose that the robot has learned p^* places – namely $\mathcal{P} = \{1, ..., p^*\}$. Initially, the set of learned places $\mathcal{P} = \emptyset$ with $p^* = 0$. Each place $p \in \mathcal{P}$ is defined by appearance data collected from a multitude of base points $x_j(p)$ as determined in place detection.

The respective appearance data are then encoded by a set of descriptors which are then retained in the place memory. We use two types of descriptors as explained in Section 2.8: Segments Summary Graphs (SSG) and Bubble Descriptors (BD). Place descriptors evolve in time and are calculated incrementally as the robot navigates to new base points.

The memory is organized in a tree hierarchy - based on previous work [32]. There are two aspects of the hierarchy. First, its structure is defined by a nested sequence of partitions of $\mathcal{P}$ in the appearance space. Each node N is associated with a subset of bubble descriptors $\mathcal{C}(N)$ and SSGs $\mathcal{G}(N)$ that are associated with a cluster of places $\mathcal{P}(N) \subset \mathcal{P}$. If N is a root note, then let $\mathcal{C} \equiv \mathcal{C}(N)$. If N is a terminal node, then the set $\mathcal{C}(N)$ consists of only the descriptors associated and $\mathcal{P}(N) = \{p\}$. Edges between

two different nodes N and N' are constructed based on the proximity of their centroids as measured by the distance function γ.

$$\gamma(N, N') = \left|\bar{I}(N) - I(N')\right| \tag{3.1}$$

where $I(N)$ is the mean of the bubble descriptors of places associated with node N as:

$$I(N) = \frac{1}{|P(N)|} \sum_{p \in P(N)} \bar{I}_p \tag{3.2}$$

Second, the structure evolves based on the hierarchical single link clustering method SLINK [33], In this method, a place is inserted into hierarchy based on the pairwise similarity score in the appearance space as measured by its distance to the places associated with its nodes.

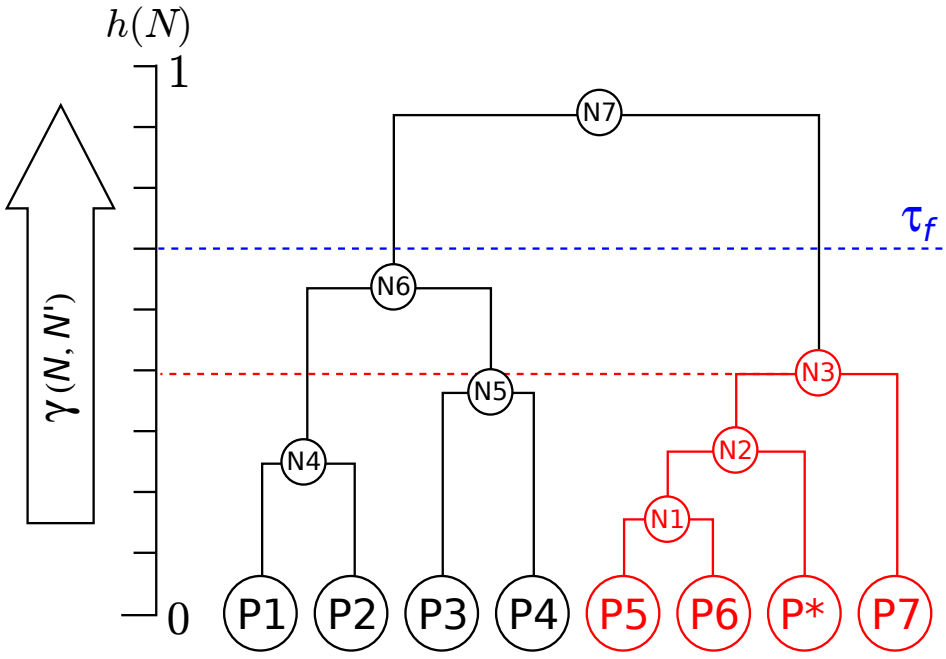


Figure 3.1. Place memory and association

The hierarchical organization enables the robot to efficiently relate to its existing knowledge. As such, it tries to assign each detected place a place $\beta(D_m) \in \mathcal{P}$ from its place memory. This is because the robot can associate an incoming place with its memory via traversing down the hierarchy [16]. Here, we propose novel approach in

which the respective SSGs are used in the decision process as well. The main steps of the reasoning are shown in Figure 3.2. First, the detected place D_m is first inserted into the place memory as a new place P^* temporarily. Let the corresponding node be indicated by $N(P^*)$. Next, the largest subtree of the memory hierarchy that has at most τ_f height while containing P^* as its terminal node is determined. Let Ω^m denote the terminal nodes of this subtree. For example, referring to Figure 3.1, if we assume that P^* is the node that is temporarily inserted, the subtree with distance less than τ_f will be that with $N3$ as its root node as shown. For example, in Figure 3.2, consider a newly detected place (indicated by light green). If it is inserted as shown, the candidate places Ω^m that are familiar with it would be the places as indicated by the dark green. The last step is to determine $\beta(D_m)$ from the set Ω. This is based on a hybrid decision criteria as given in Equation 3.3. It considers the matching of both hybrid descriptors as well as their SSG representations as:

$$\beta(D_m) = \begin{cases} \in \text{argmin}_{N' \in \Omega^m} \gamma(N(P^*), N') & \text{if } \hat{\mathcal{N}}^m(N') > \tau_s \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $\hat{\mathcal{N}}^m(N')$ denotes the number of matching nodes (segments) between the SSG of the detected place D_m and those of terminal node N'. If $\beta(D_m) > 0$, then the detected place is recognized as the place associated with respective terminal node and the associated place descriptor is inserted into $\mathcal{C}(N_r)$ while the remaining place memory remains unchanged. Otherwise, the place is not recognized either because of having an empty candidate set or not satisfying the hybrid decision criteria. The place label of the detected but unrecognized place is set as $\beta(D_m) = p^* + 1$. In this case, the location of the temporarily inserted place node is made permanent and hence the place memory is updated.

3.3. Experimental Results

In this section, we present experimental results as follows: First, we study how the place memory is formed. Second, memory association performance is investigated.

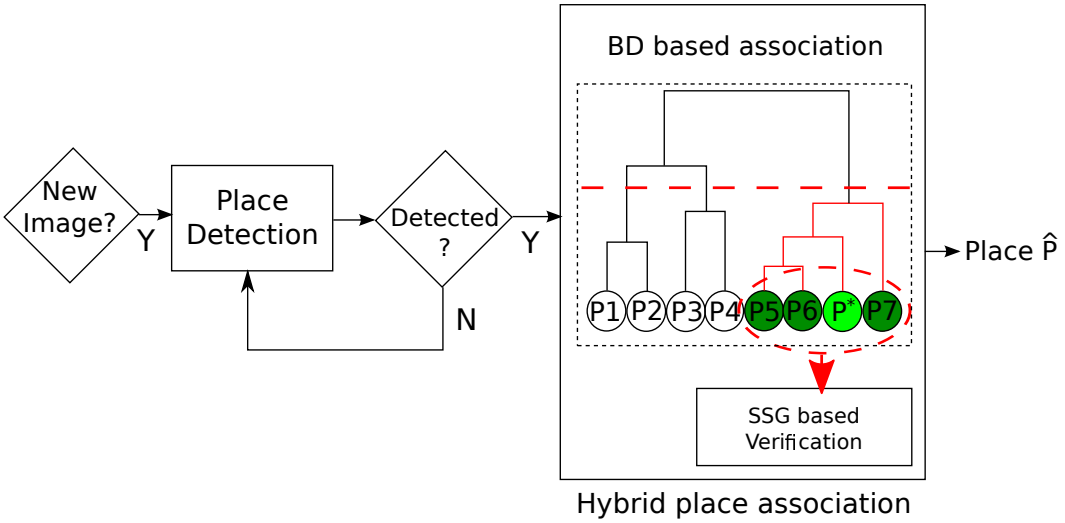


Figure 3.2. Place memory association.

3.3.1. Place Memory

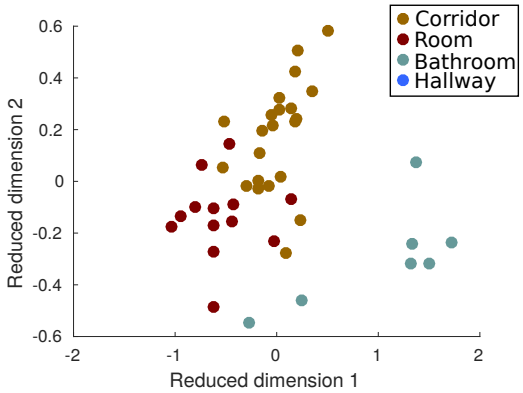
In this section, we evaluate place memories constructed based on the detected places in the COLD dataset under cloudy illumination. These are manually given two labels based on their functionality (office, corridor, hall and bathroom) and site (Fr, Lj and Sa). The place memories differ in the type of descriptors used: SSG and BD. The effectiveness of the resulting memories is evaluated by a human considering knowledge organization and memory association.

First, evaluate the resulting organization with respect to their labels. We expect places having similar labels to be close together in the place memory. For this, SSG and BD descriptors are projected onto 2D plane using a multi dimensional scaling (MDS) method. The results are shown in Figure 3.3 with detected places labelled according to their function labels. For example, bathroom category is well separated from the other categories with the BD while that is not the case with SSG. While the descriptors of places having either room or corridor labels are close with both of the types, BD's are further whereas that is not the case for SSG. A similar situation holds for descriptors of places having hall label. In Lj site, SSG based descriptors of three categories as shown in Figure 3.3(f) are scattered homogeneously without showing any apparent clustering pattern. This is in contrast to BD case in Figure 3.3(e) where the hall category is

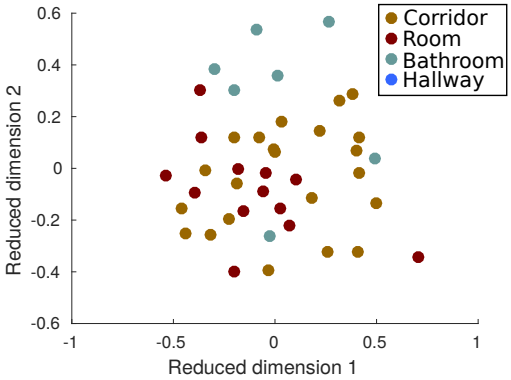
separable from the room category up to a certain degree. These observations lead us to think that a global such as a bubble descriptor is more suitable for learning. Actually, this is expected since a global descriptor encodes the whole appearance and thus is able to distinguish it from appearances from other places much better. As such, the place memory is built using bubble descriptors.

Next, we proceed to verify this claim. For this, we compare place memories whose knowledge includes appearances from the detected places that are encoded internally as either SSG or BD - as given in Figure 3.4. The manually given two labels of the terminal nodes are indicated by colored triangle for the function (Red=Office, Orange=Corridor, Blue=Hall, Turquoise=Bathroom) and colored circle for the site (Blue=Fr, Black=Sa, Orange=Lj). The number of the natural clusters, homogeneity among clusters and the accordance with the ground-truth categories are the criteria used in the evaluation.

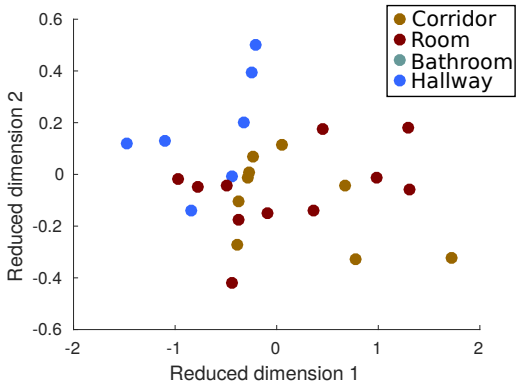
For example, the place memories that store the knowledge of Fr site are shown in Figure 3.4(b) and 3.4(a) respectively. It is observed that the BD based place memory contains three natural clusters which is in accordance with the number of categories in the Fr site. The first cluster is formed at the highest level and contains places from bathroom category only. The second and the third clusters are formed at the second level. The second cluster mostly contains places corresponding to the room category while the third cluster contains places from the corridor only. Place 13(bathroom) and 16(corridor) are wrongly inserted into the second cluster however the familiarity degree of them within cluster is relatively less. Similarly, SSG based place memory contains three natural clusters. However, the first cluster corresponding to the bathroom category has only one member. The second and the third clusters are relatively less homogeneous. Two places related to the bathroom category are inserted into the second cluster however familiarity degree is low within the group which indicates that they are distinct from the rest of the cluster. Furthermore, the second cluster contains places from the room and corridor categories but they are grouped within their category. The third cluster mostly contain places from the corridor. Place 6 and 13 are outliers and wrongly inserted into this cluster. Place memories of Sa site are given in



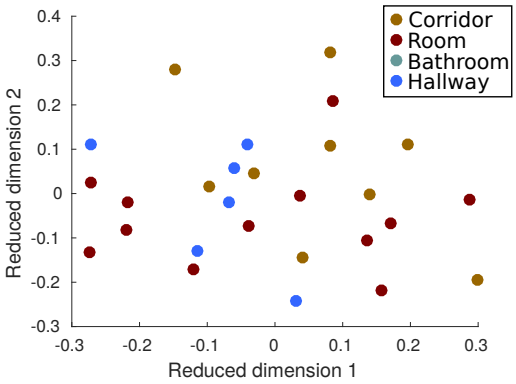
(a) Descriptor: BD, Site: Fr



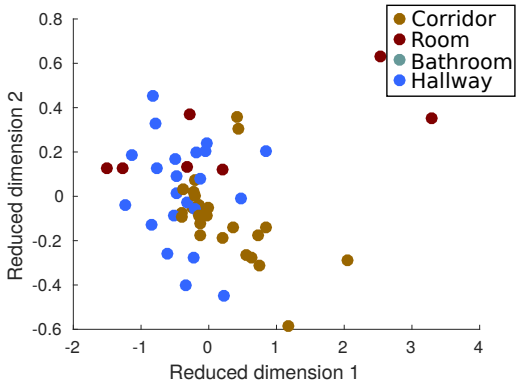
(b) Descriptor: SSG, Site: Fr



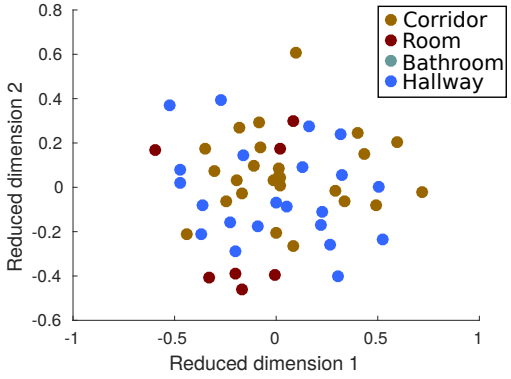
(c) Descriptor: BD, Site: Sa



(d) Descriptor: SSG, Site: Sa



(e) Descriptor: BD, Site: Lj



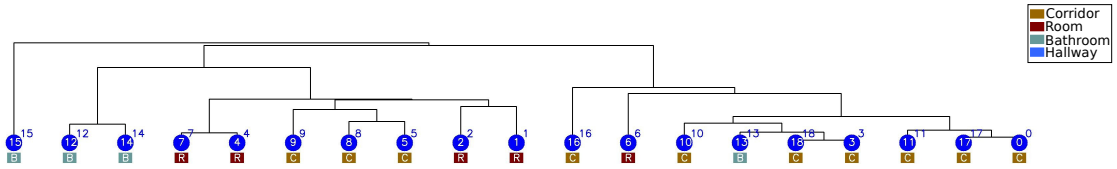
(f) Descriptor: SSG, Site: Lj

Figure 3.3. Planar projections of place descriptors using MDS method. Place categories are indicated by colors: Room (red) , Corridor (orange), hall (blue), bathroom (turquoise).

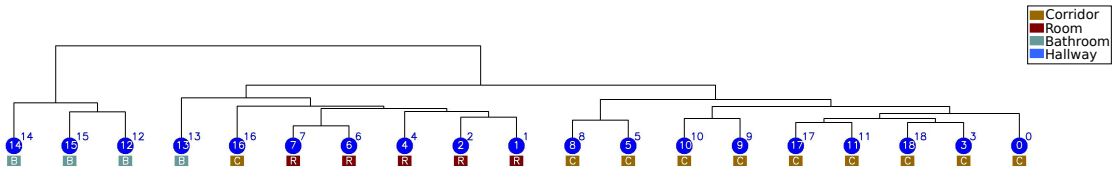
Figure 3.4(d) and 3.4(c). In BD based memory, there are two four natural clusters. The first cluster is formed at the highest level and contains two places from the room and corridor categories, respectively. A closer inspection reveals that these places are both visually and physically close to each other. The second cluster is formed next and it contains places mostly from the room category. Third cluster contains places from the corridor category only. The last cluster contains places from the hall category. The place 8 is wrongly inserted into this cluster and it should be placed into the second cluster. Similarly, there are four natural clusters in SSG based memory however cluster contents do not share any similarity with the BD based memory. Furthermore, resulting hierarchy is not in accordance with the content of the clusters. Lastly, places memories corresponding to Lj site are given in Figure 3.4(f) and 3.4(e). It is observed that place 8 is highly dissimilar to the rest of the memory and placed as a single cluster. The rest of the memory is separated into three natural clusters. The first cluster mostly contains places from the hall category. The second cluster contains places detected from the corridor only. The cluster three is divided into two: the first cluster contains places from the corridor whereas second cluster mostly contains places from the hall. Contents of the places 27 and 14 are not related to the cluster and the familiarities within the group are low as expected. SSG based memory of Lj site does not reveal any apparent natural clustering pattern however some places from the corridor and hall categories are grouped in respective clusters on the right half of the cue tree. As validated by the results of the second set of experiments, using BD descriptors in comparing the similarity of places offers a better clustering performance and hence the place memory reveals the ground-truth hierarchy more accurately.

3.3.2. Place Association

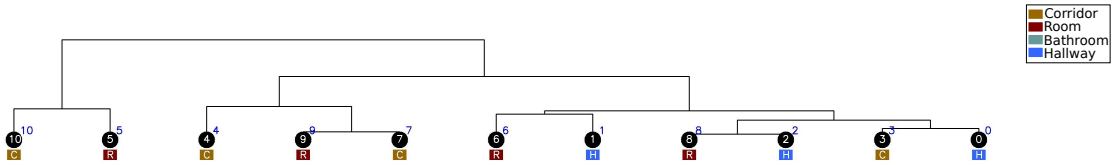
In the place association experiments, the robot revisits some of the visited sites once again and we evaluate the association performance. While robot travels through visited places, the exact path as well as the illumination differ. Furthermore, at some sites, the robot visits some places for the first time. Therefore, the total number of detected places as well as the extent of the places will not be exactly the same with



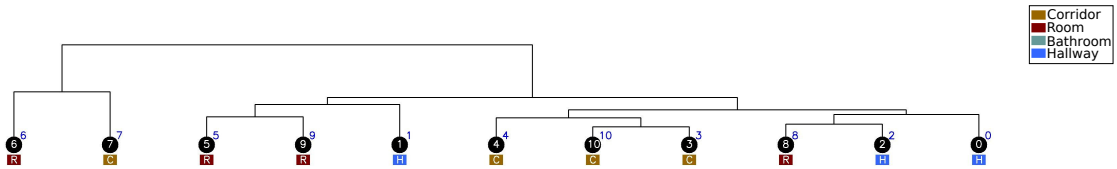
(a) Descriptor: SSG, Site: Fr



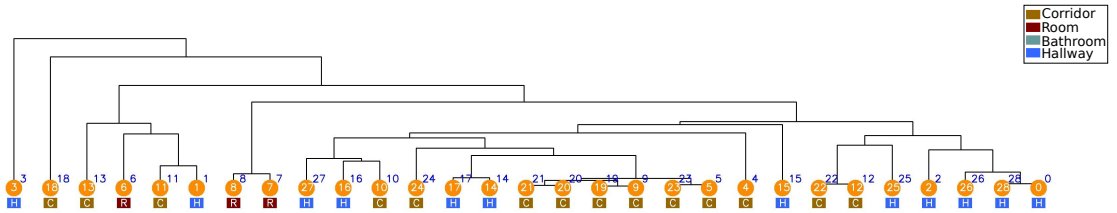
(b) Descriptor: BD, Site: Fr



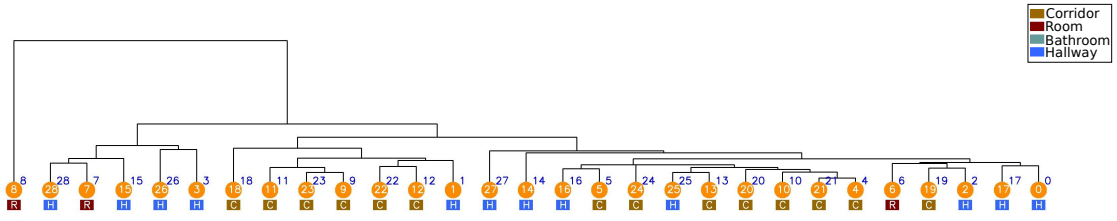
(c) Descriptor: SSG, Site: Sa



(d) Descriptor: BD, Site: Sa



(e) Descriptor: SSG, Site: Lj



(f) Descriptor: BD, Site: Lj

Figure 3.4. Place memories. Numbers in circles indicate detected place indices.

their counterparts in the first tour. The place memory is constructed using BD. The experiments are repeated using three alternative memory association methods: BD, SSG and Hybrid methods with varying association cost threshold $\tau_f \in \{0, 1\}$. We evaluate performance through three means. First, we consider how the place memory evolves. If the detected place is maximally associated, it will be placed just below to the respective node instead of inserting it as a new place node. In this case, its correctness is verified manually based on its visual as well as locational similarity. Second, we compute maximal association recall-precision rates. Finally, we use a 'familiarity' score to retrieve a set of candidate places among which will be the current place. This is because due to dynamic changes in the scene appearance, while maximal association may not be attained, familiarity may be possible. In this case, we find a set of candidate places based on a familiarity score instead of outputting just one exact match.

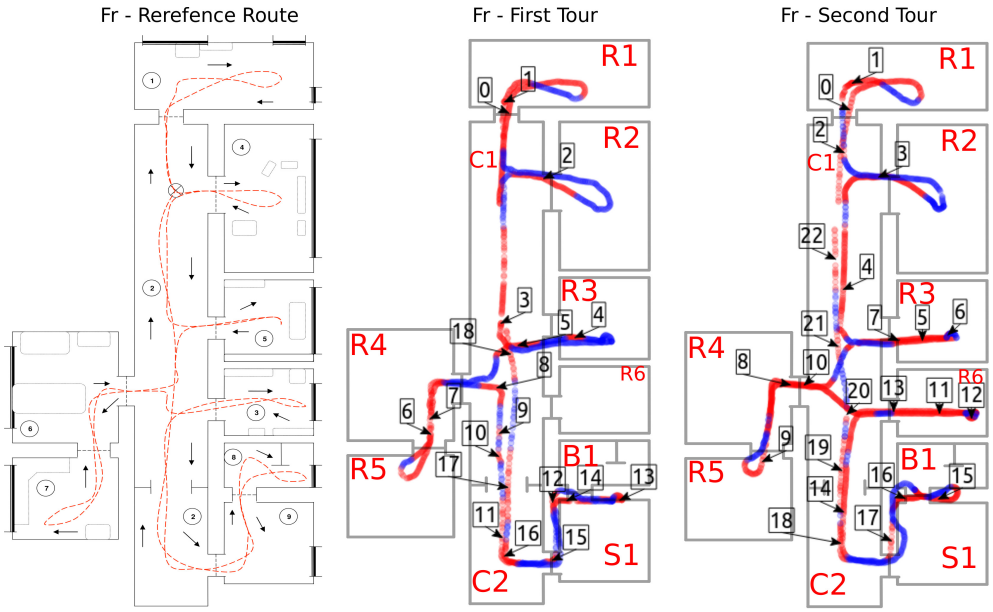
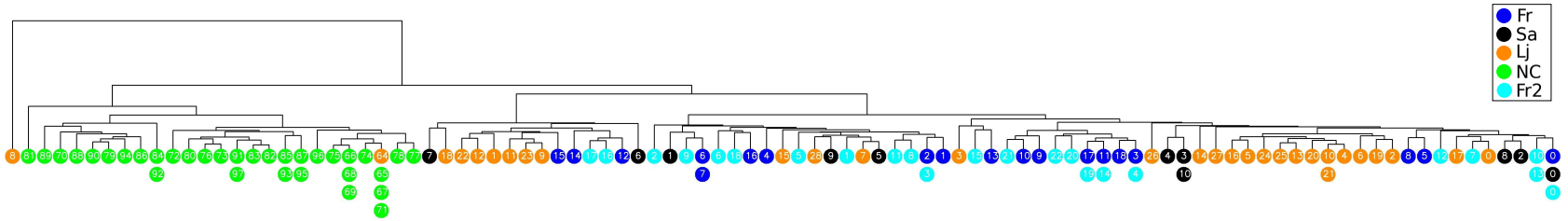


Figure 3.5. Place detection in Fr site.

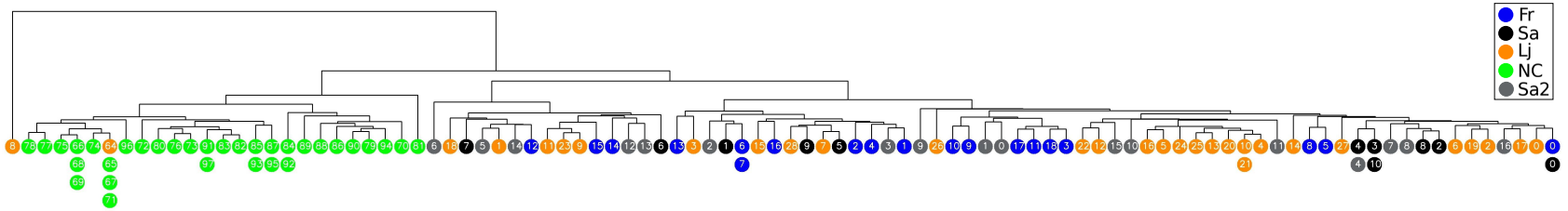
Table 3.1. Correspondence of detected places in the second tour with those of first tour in Fr site.

Tour #	Detected place index														
2	0	1	3	4	7	8	9	10	14	15	16	17	18	19	21
1	0	1	2	3	5	6	7	8	9,10,11	13	14	15	16	17	18

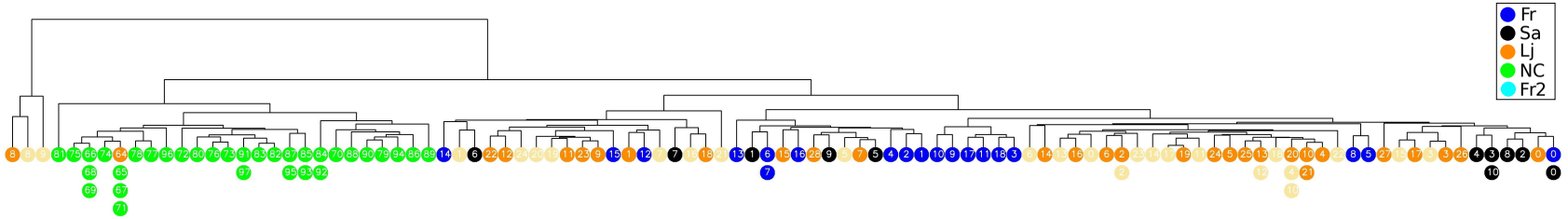
In the first part of the experiments, the robot revisits only one of the previously visited sites. After revisiting the Fr site, the robot detects 23 places as shown in Figure 3.5. It is observed that 15 of these places coincide with those previously revisited as given in Table 3.1 and should be associated as such. The place memory evolves as shown in Figure 3.6(a). It is observed that only few places are maximally associated. For example, detected places 0, 3 and 4 from the second tour (light blue nodes) are maximally associated with the detected places 0, 2 and 3 (dark blue nodes) of first tour. Interestingly, it is observed that spatially and visually similar places are grouped closer in the memory. In the Fr site, detected places in the first and second visit are colored in dark and light blue, respectively and they are clustered together in several locations in the memory. For example, places 7 and 13 from the first visit are inserted next to their counterpart places 9 and 15 from the second visit, respectively. In some other cases, places are inserted into the memory in such a way that they have the same parent at the second level. Places 1, 5, 8 and 14 from the first visit and places 1, 7, 10 and 16 from the second visit are examples of such a case. These results show that even if maximal association does not occur, the detected place will be inserted into very close neighborhood of its counterpart in the memory. The recall precision curves are given in Figure 3.10(a). It is observed that maximal association rates are much better when BD or hybrid method is used. The maximal association performances of BD and hybrid method are comparable. While the hybrid method can achieve higher precision at medium recall rates (i.e. 70%) BD method can have higher recall rates. When the robot operates at full precision, we can obtain recall rate of 20%, at most. That means only 3 of 15 places are maximally associated while the remaining 12 places are not hence they are inserted as a new place into the place memory. Table 3.4 shows the association rates when a familiarity metric is used. In Fr site, the candidates are observed to contain the place to be recognized in 82% of the cases and the number of candidates is at max 4 in any case. These results show that familiarity metric can be utilized as an auxiliary method in memory association. For example, it can provide a prior in terms of a set of candidates to limit the search space for the final decision.



(a) Revisiting Fr site



(b) Revisiting Sa site



(c) Revisiting Lj site

Figure 3.6. Place memories after second time visits

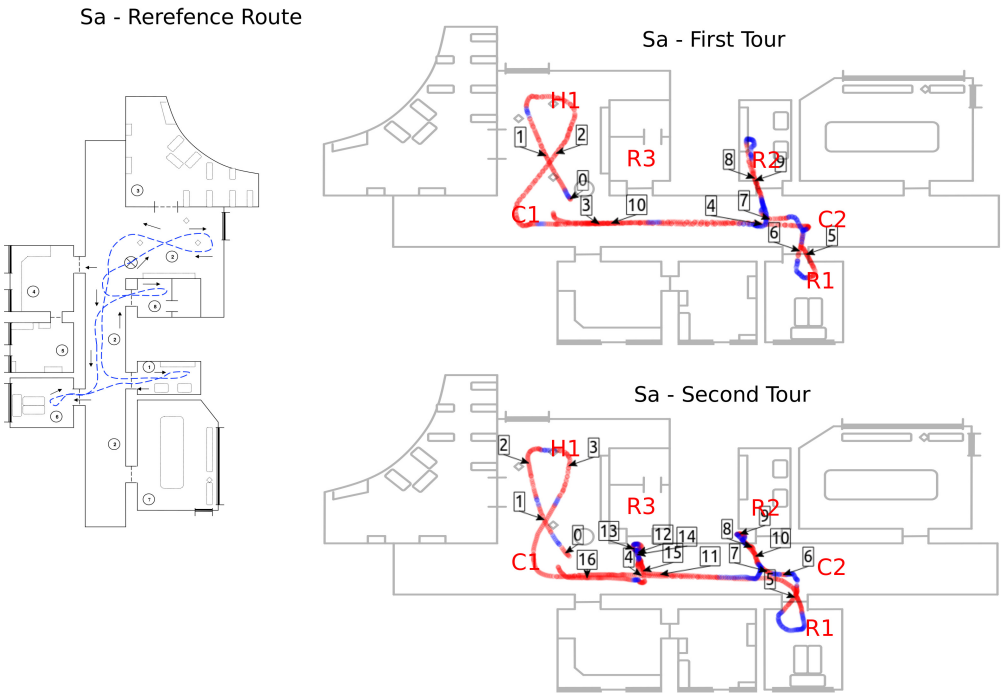


Figure 3.7. Place detection in Sa site.

Table 3.2. Correspondence of detected places in the second tour with those of first tour in Sa site.

Tour #	Detected place index									
2	0	1,2	3	4	5	6	7	8	10	11
1	0	1	2	3,4	5	6	7	8	9	10

When the robot revisits Sa site, it detects 17 places as seen in Figure 3.7. It is noted that 11 of these places have been previously visited as given in Table 3.2. The place memory evolves as shown in Figure 3.6(b). Detected places in the Sa site are observed to be spreaded out in clusters across the memory. This is mainly due to two main reasons: First, place contents are not very characteristic so that already formed memory can handle insertion of detected places without deforming the shape of the memory. Second, detected places from the Sa and Lj site share a lot of common visual content therefore they are grouped together in most of the cases. It is noted that only one detected place is maximally associated. However, many places such as places 1, 3 and 8 from the first visit and places 2, 4 and 8 from the second visit are inserted into the immediate neighborhood of their counterpart places. Association rate for the

Sa site is calculated as 80% which means that places to be recognized are among the selected candidates in 80% of all test cases. With 100% precision we can obtain recall rate up to 30% using a hybrid method as shown in Figure 3.10(b). When the precision becomes around 65%, recall incrases up to 60%. Using BD method, we obtain lower precision at the same recall rates however maximum recall can go up as much as 90% at 20% precision. SSG method performs poor in terms of precision at any recall rate.

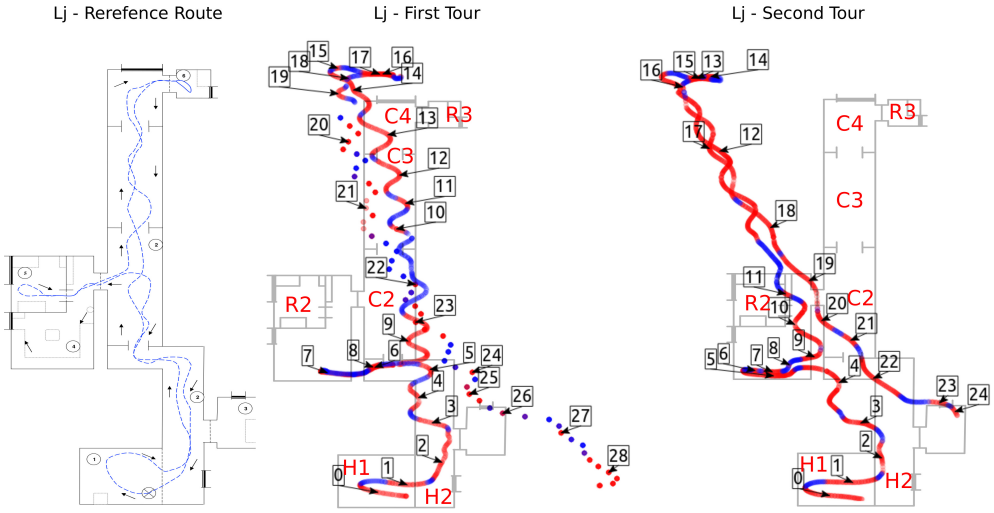


Figure 3.8. Place detection in Lj site.

Table 3.3. Correspondence of detected places in the second tour with those of first tour in Lj site.

Tour #	Detected place index																
2	0	1	2	3	4	5	7,8	9,10	12	13	15	17	19,20	21	22	23	24
1	0	1	2	3	4,5	7	8	9	10,11,12,13,14	16	17	19,20,21	23	24	25	27	28

Finally, the robot revisits Lj site and detects 25 places as seen in Figure 3.8. 20 of these coincide with those from the previously visited places as given in Table 3.3. Detected places from similar locations are grouped together and inserted into very close neighborhood. For example, places 3 and 17 from the first visit are inserted next to their counterpart places 3 and 15 from the second visit. Similar to the previous case, detected places from Sa site and Lj site are located closely to each other as expected. Association rate is calculated as 74% which is slightly lower compared to other sites. However, the number of detected places is almost three times of the Sa site and 1.5

times of Fr site which makes accurate association in Lj site harder to achieve. Neither of BD or hybrid method can achieve 100% precision rate in this site. The maximum achievable precision is about 85% with 40% recall using BD based method. SSG method correctly recognizes one place with 5% recall however precision rate sharply decreases when the recall rate increased. Interestingly, BD based method can achieve higher precision compared to the hybrid method in this case. However, detected places in the revisit tour of Lj site do not overlap mostly with the places from the first visit and therefore evaluation may not reflect the actual performances for this particular site.

In the second part, the robot revisit all the places and forms a complete memory of all visited places as shown in Figure 3.9. Two main criteria can be proposed for checking the success rate of the association of detected places in the memory: First, relative locations of detected places should not be changed much as the memory expands. Second, association rates should not decrease much as the number of places gets larger. In order to check the first criteria, complete memory is compared against previously formed memories where only one site is revisited in each. Relative locations and hierarchical relations of the places associated with each site are observed to be preserved in the complete map, as well. However, hierarchical distances between the memory clusters are increased due to the expansion of memory. The recall precision curve is given in Figure 3.10(d). We see that the increase in the total number of places does not affect its memory association performance much. We can still obtain around 90% precision with 20% recall and 70% precision with 40% recall. These results conclude that proposed memory organisation framework is a good candidate for storing places as well as their hierarchy. Moreover, it is proved that our method is scalable as the association rates are preserved as the size of the memory increases.

3.3.3. On-Robot Experiments

In these experiments, the Jaguar robot travels though North Campus site as discussed in Section 2.9.3. It detects 28 and 30 places in the two separate tours as shown in Figure 2.14(a). The correspondence of detected places from the second tour

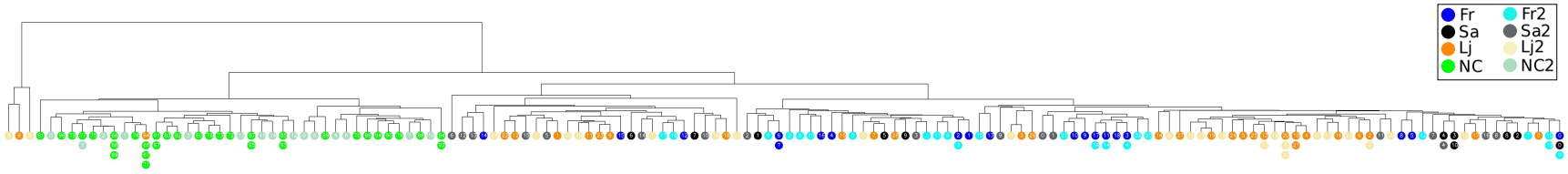
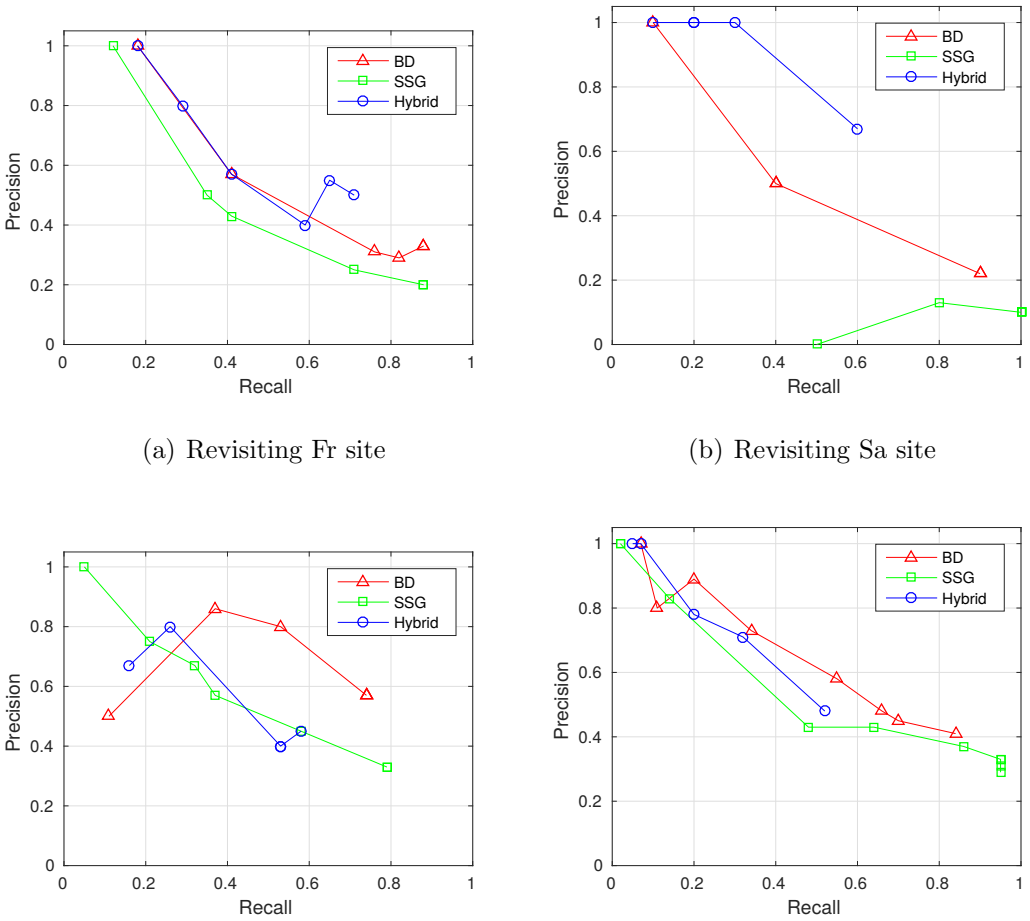


Figure 3.9. Place memory after revisiting all places



(a) Revisiting Fr site

(b) Revisiting Sa site

(c) Revisiting Lj site

(d) Revisiting all sites

Figure 3.10. Precision-Recall curves after revisiting sites

Table 3.4. Association rates and maximum number of candidates: The place estimate will be among the candidates with given association rate.

Site	Association Rate	Max # of candidates
Fr	82%	4
Sa	80%	3
Lj	74%	3

Table 3.5. Correspondence of detected places in the first and second tours in North Campus site with Jaguar robot.

Tour #	Detected place index																
1	0	1	2	3 4 5	7	8 9	10	11	12 13	14	15 16 17	18 19	20	21 22 23	24	26 27	28
2	0	1	2	3 4 5	6	7	9	11 12 13 14	15 16 17 18	21 22	23	24	25	26	27	29	30

with those of the first tour are given in Table 3.5. The place memory after these two tours is as shown in Figure 3.11. Places that are added after first and the second tours are indicated in black and red respectively. It is observed that the place memory is first divided into two main groups which correspond to the indoor and outdoor regions. For example, black colored places 26-28 and red colored places 2 and 29 are observed to be indoor places. The second group is further divided into two groups. The first group contains mostly places from vegatation areas. For example, vegetation area in the first tour are encoded in the places 4-10 and all of these places are located in the this group in the memory. On the other hand, places in the second group mostly correspond to the car parking or buildings area. These results show that places are inserted into the place memory hierarchically according to their visual contents. Next, maximal association performances are evaluated. Although maximal association couldn't be observed in many cases, it can be easily observed from Figure 3.11 that revisited places are inserted into very close neighborhood (5 nodes away) of their counterpart places from the first tour in nearly two-thirds of the cases. Furthermore, several places are recognized to places from the same tour. This is expected as these places are either adjacent places or having similar contents.

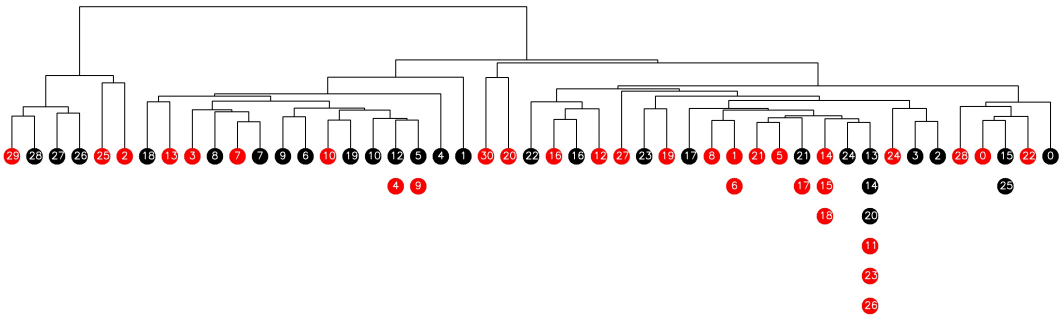


Figure 3.11. Place memory after revisiting the North Campus site. Learned places from first tour are indiated in black while those from the second tour are indicated in red.

3.4. Conclusion

In this chapter, we consider the coupling of place detection with place memory. This manifests itself through two mechanisms. First, the knowledge stored in place memory in regards to each place is determined by the the appearance data belonging to the respective detected place. In particular, they are stored after being internally represented using bubble descriptor representation. This is a hybrid representation that has characteristics of both global and local descriptors. The effects of using SSG and BD based descriptors are evaluated through a set of experiments. Second, association with the memory is done considering the currently detected place as it relates to the place memory with the respective segments summary graph used in the decision making. A hybrid decision criteria which utilizes both the local content and global scene information through SSG and BD descriptors, respectively. As such, the reliability of memory association can be improved.

4. CONCLUSION

This thesis focuses on appearance-based place detection and its coupling with place memory. A novel approach to place detection based on coherent segments is introduced. This is motivated by the fact segments encode scene contents at an intermediate level of representation while being relatively stable under a wider range of viewpoints and dynamical changes - differing from global, local or hybrid descriptors. Places are detected via tracking the coherency of region adjacency graphs across the incoming appearance data. As such, place detection can be done more reliably while simultaneously generating a segments summary graph for each place that can be used in ensuing the semantic analysis of the place. Following, the coupling of place detection with place memory is considered. This manifests itself through two mechanisms. First, the knowledge stored in place memory in regards to each place is determined by the the appearance data belonging to the respective detected place. In particular, they are stored after being internally represented using bubble descriptor representation. This is a hybrid representation that has characteristics of both global and local descriptors. Second, association with the memory is done considering the currently detected place as it relates to the place memory with the respective segments summary graph used in the decision making. As such, the reliability of memory association can be improved.

We are considering two extensions of this work. First, the place detection module thus developed will be integrated within the topological spatial cognition model that has been previously developed. As place memory expands, the converted hierarchies can be utilized in an unsupervised manner to find the natural categories of places. Second, the resulting segments summary graphs will be used for a higher level semantic understanding - in particular the recognition of objects within the place.

REFERENCES

1. Miller, S., *Space and Sense*, Psychology Press, 2008.

2. Kuipers, B., "The spatial semantic hierarchy", *Artificial intelligence*, Vol. 119, No. 1, pp. 191–233, 2000.

3. Karaoguz, H. and H. I. Bozma, "Reliable topological place detection in bubble space", *IEEE International Conference on Robotics Automation*, pp. 697–702, 2014.

4. Vasudevan, S., S. Gächter, 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.

5. Cummins, M. and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance", *The International Journal of Robots Research*, Vol. 27, No. 6, pp. 647–665, 2008.

6. Konolige, K., J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit and P. Fua, "View-based maps", *The International Journal of Robotics Research*, 2010.

7. Matsumoto, Y., M. Inaba and H. Inoue, "Visual navigation using view-sequenced route representation", *IEEE International Conference on Robotics Automation*, pp. 83 – 88, 1996.

8. Ulrich, I. and I. Nourbakhsh, "Appearance-based place recognition for topological localization", *International Conference on Robotics and Automation*, Vol. 2, pp. 1023–1029, IEEE, 2000.

9. Nourani-Vatani, N., P. V. K. Borges, J. M. Roberts and M. V. Srinivasan, "On the use of optical flow for scene change detection and description", *Journal of Intelligent & Robots Systems*, Vol. 74, No. 3-4, pp. 817–846, 2014.

10. Rituerto, A., A. Murillo and J. Guerrero, "Semantic labeling for indoor topological mapping using a wearable catadioptric system", *Robots and Automation Systems*, Vol. 62, No. 5, pp. 685–695, 2014.

11. Nicosevici, T. and R. Garcia, *Efficient 3D Scene Modeling and Mosaicing*, Vol. 87, Springer, 2013.

12. Korrapati, H. and Y. Mezouar, "Vision-based sparse topological mapping", *Robots and Automation Systems*, Vol. 62, No. 9, pp. 1259–1270, 2014.

13. Murphy, L. and G. Sibley, "Incremental unsupervised topological place discovery", *IEEE International Conference on Robotics and Automation*, pp. 1312–1318, IEEE, 2014.

14. Zivkovic, Z., B. Bakker and B. Krose, "Hierarchical map building using visual landmarks and geometric constraints", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2480–2485, IEEE, 2005.

15. Guillaume, H., M. Dubois, F. Emmanuelle and P. Tarroux, "Temporal bag-of-words-a generative model for visual place recognition using temporal integration", *VISAPP-International Conference on Computer Vision Theory and Applications*, 2011.

16. Karaoğuz, H. and H. I. Bozma, "An integrated model of autonomous topological spatial cognition", *Autonomous Robots*, pp. 1–24.

17. Espinace, P., T. Kollar, N. Roy and A. Soto, "Indoor Scene Recognition by a Mobile Robot Through Adaptive Object Detection", *Robotics Autonomus Systems*, Vol. 61, No. 9, pp. 932–947, 2013.

18. Lee, J., J. Oh and S. Hwang, "Scenario based dynamic video abstractions using graph matching", *ACM International Conference on Multimedia*, pp. 810–819, ACM, 2005.

19. Mei, S., G. Guan, Z. Wang, S. Wan, M. He and D. D. Feng, "Video summarization via minimum sparse reconstruction", *Pattern Recognition Letters*, Vol. 48, No. 2, pp. 522–533, 2015.

20. Demir, M. and H. I. Bozma, "Video Summarization via Segments Summary Graphs", *ICCV: Video Summarization for Large-Scale Analytics Workshop*, pp. 19–25, 2015.

21. Andreopoulos, A. and J. K. Tsotsos, "50 Years of object recognition: Directions forward", *Computer Vision and Image Understanding*, Vol. 117, No. 8, pp. 827 – 891, 2013.

22. Ngo, C.-W., Y.-F. Ma and H.-J. Zhang, "Video summarization and scene detection by graph modeling", *IEEE Transactions on Circuits and Systems for Video Tech.*, Vol. 15, No. 2, pp. 296–305, 2005.

23. Felzenszwalb, P. F. and D. P. Huttenlocher, "Efficient graph-based image segmentation", *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167–181, 2004.

24. Everingham, M., S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective", *International Journal of Computer Vision*, Vol. 111, No. 1, pp. 98–136, 2015.

25. Jouili, S., I. Mili and S. Tabbone, "Attributed graph matching using local descriptions", *Advanced Concepts for Intelligent Vision Systems*, pp. 89–99, 2009.

26. Kuhn, H. W., "The Hungarian method for the assignment problem", *Naval research logistics quarterly*, Vol. 2, No. 1-2, pp. 83–97, 1955.

27. Marchionini, G. and G. Geisler, "The open video digital library", *D-Lib Magazine*, Vol. 8, No. 12, pp. 1082–9873, 2002.

28. Furini, M., F. Geraci, M. Montangero and M. Pellegrini, "STIMO: Still and moving video storyboard for the web scenario", *Multimedia Tools and Applications*, Vol. 46, No. 1, pp. 47–69, 2010.

29. de Avila, S. E. F., A. P. B. Lopes, A. da Luz and A. de Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method", *Pattern Recognition Letters*, Vol. 32, No. 1, pp. 56–68, 2011.

30. Erkent, Ö. and H. I. Bozma, "Bubble space and place representation in topological maps", *The International Journal of Robotics Research*, Vol. 32, No. 6, pp. 672–689, 2013.

31. Erkent, Ö. and I. Bozma, "Place representation in topological maps based on bubble space", *IEEE International Conference on Robotics and Automation*, pp. 3497–3502, IEEE, 2012.

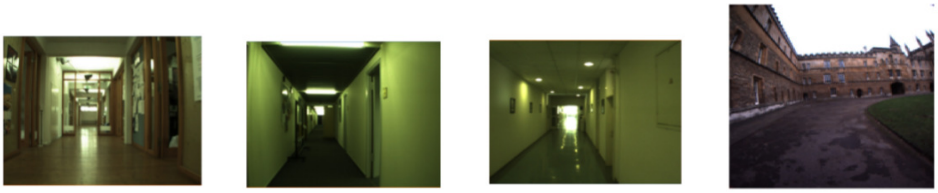
32. Erkent, Ö. and H. I. Bozma, "Long-term topological place learning", *IEEE International Conference on Robotics and Automation*, pp. 5462–5467, IEEE, 2015.

33. Sibson, R., "SLINK: an optimally efficient algorithm for the single-link cluster method", *The Computer Journal*, Vol. 16, No. 1, pp. 30–34, 1973.

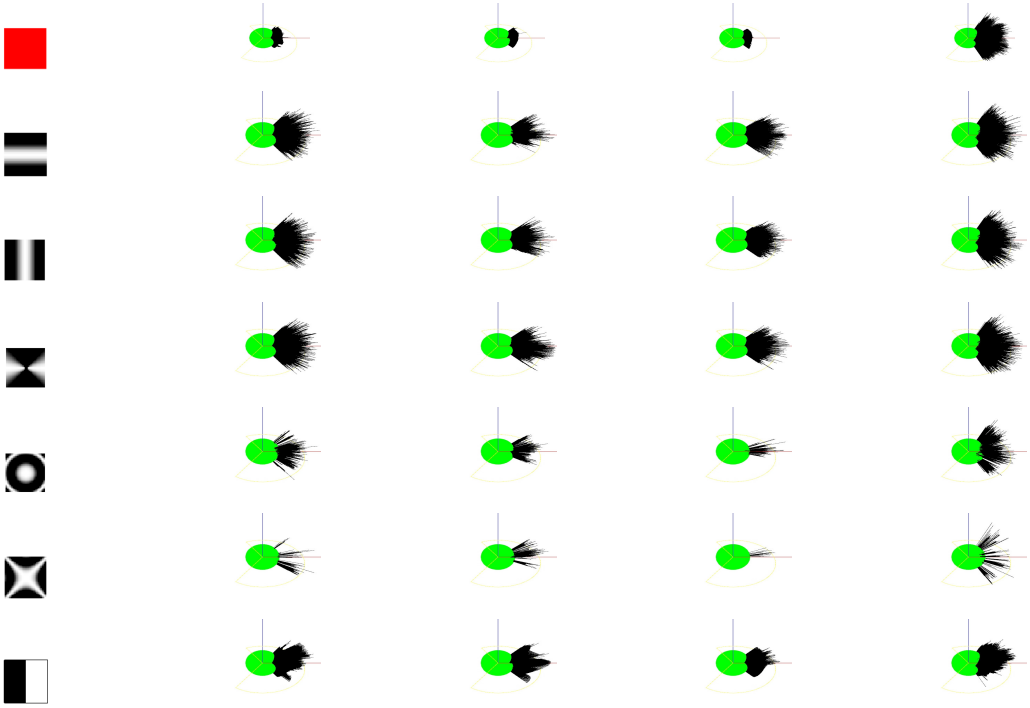
34. Pronobis, A. and B. Caputo, "COLD: COsy Localization Database", *The International Journal of Robotics Research*, Vol. 28, No. 5, pp. 588–594, 2009.

35. Smith, M., I. Baldwin, W. Churchill, R. Paul and P. Newman, "The new college vision and laser data set", *International Journal Robotics Research*, Vol. 28, No. 5, pp. 595–599, 2009.

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, [34] and NC sites [35].

This section presents a brief summary of bubble space representation for completeness. The interested reader is referred to [30] 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 Figure A.1(a), the bubble surfaces are as shown Figure A.1(b). 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:

$$\rho_i(b,t) = \sum_{h_1=0}^{H_1} \sum_{h_2=0}^{H_2} \lambda_{h_1 h_2} z^T_{xi,h_1 h_2}(t) e_{h_1 h_2}(f) \tag{A.3}$$

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

$$e_{h_1 h_2}(f) = \begin{bmatrix} cos(h_1 f_1) cos(h_2 f_2) \\ sin(h_1 f_1) cos(h_2 f_2) \\ cos(h_1 f_1) sin(h_2 f_2) \\ sin(h_1 f_1) sin(h_2 f_2) \end{bmatrix} \tag{A.4}$$

The corresponding vector $z_{xi,h_1 h_2}(t) \in R^4$ is defined as:

$$z_{xi,h_1 h_2}(t) = \frac{1}{\pi^2} \begin{bmatrix} \int_0^{2\pi} \int_0^{\pi} \rho_i(b,t) cos(h_1 f_1) cos(h_2 f_2) df_1 df_2 \\ \int_0^{2\pi} \int_0^{\pi} \rho_i(b,t) sin(h_1 f_1) cos(h_2 f_2)) df_1 df_2 \\ \int_0^{2\pi} \int_0^{\pi} \rho_i(b,t) cos(h_1 f_1) sin(h_2 f_2)) df_1 df_2 \\ \int_0^{2\pi} \int_0^{\pi} \rho_i(b,t) sin(h_1 f_1) sin(h_2 f_2) df_1 df_2 \end{bmatrix} \tag{A.5}$$

The parameters $\lambda_{h_1 h_2}$ are defined as:

$$\lambda_{h_1 h_2} = \begin{cases} \frac{1}{4} & \text{if } h_1 = 0, h_2 = 0 \\ \frac{1}{2} & \text{if } h_1 > 0, h_2 = 0 \text{ } or \text{ } h_1 = 0, h_2 > 0 \\ 1 & \text{if } h_1 > 0, h_2 > 0 \end{cases} \tag{A.6}$$

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:

$$I(x,t) = [I_{1,00}(x,t), \ldots, I_{N_v,H_1H_2}(x,t)]^T \tag{A.7}$$

where

$$I_{i,h_1h_2}(x,t) = z^T_{xi,h_1h_2}(t) z_{xi,h_1h_2}(t) \tag{A.8}$$

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 is required for finding correspondences among observations taken at different times.

APPENDIX B: SSG SOFTWARE MANUAL

The proposed approach is implemented in C++ along with a GUI. In order to compile the source codes, C++ compiler, QT, OpenCV and ROS libraries are required. Up-to-date source codes can be reached via the GitHub account of the ISL. In this appendix, the developed SSG software is explained in detail. The software includes the proposed approach as well as some extensions that is helpful in debugging the approach. In addition, a brief explanation about the classes and functions in the source code is also provided.

The software is composed of three main screens: Main screen, memory association screen and parameter tuning screen. Main screen is used to run the algorithm online and see the immediate results of the place detection. Furthermore, some intermediate steps that are used in the calculation of the coherency score is shown in the main screen for debugging purposes. In the memory association screen, the hierarchical structure that is constructed based on the detected places is depicted. This module enables you to change the parameters online and see the immediate results on the resulting hierarchy. Lastly, parameter tuning screen enables you to tune the parameters and to tweak some additional settings related to GUI. Now, the detailed explanation of each screen will be given.

The main screen is presented in Figure B.1. There are seven frames. The main control buttons enable user to start/stop or run step by step the place detection process. In the first frame (1.a), the current appearance is shown. Next to it, the results of the graph matching of the last two appearances are given. In the next frame (1.c), SSG of the last detected place is shown. Below to it, detected places up to current base point are depicted. Here, black lines indicate the coherency score where the red line is for the coherency threshold. In the following frame, red regions (1.d) stand for detected places whereas blue regions (1.e) stand for transition regions. Below to it, node existence map that enables to track segments temporally across base points. White lines signify

for segments here. The coherency score is calculated based on the appear/disappear behavior of these segments. On the right of the main screen, detected places are projected on the respective map.

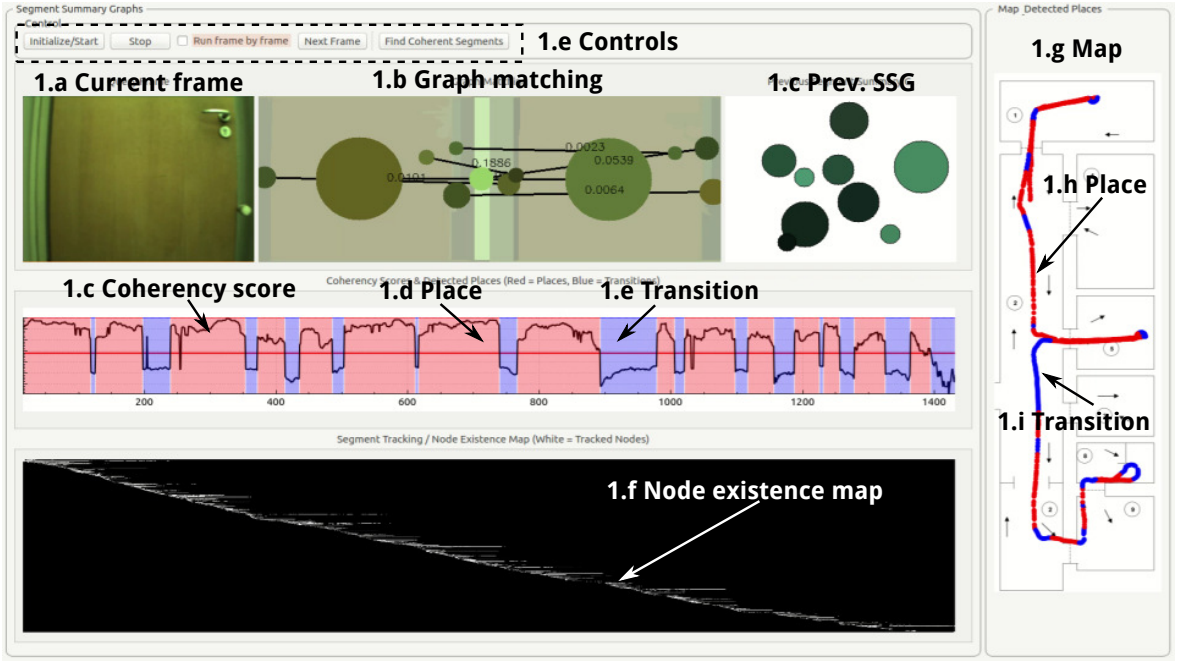


Figure B.1. Main screen of Segments Summary Graphs software

The memory hierarchy module screen is given in Figure B.2. There are two main parts in the screen: the top part is for the control and settings and the bottom parts is for displaying the resulting hierarchy. This module enables user to set the recognition method through a drop-down box (2.b) as well as tune the association parameters through sliders (2.c). The detected places can either be loaded from the database using a button shown in 2.a or via online detection method. In the bottom, the resulting hierarchy is displayed where the colors of detected places are hard-coded in the source code.

The last screen is for tuning the parameters. The parameters and extra setting can be saved into a text file using the button on the top. At startup default parameters are loaded from this file. Graph matching parameters (3.a), segmentation parameters (3.b), place detection parameters (3.c) and coherency parameters (3.d) can be changed at any time including the run-time however it is advised to keep the parameters same

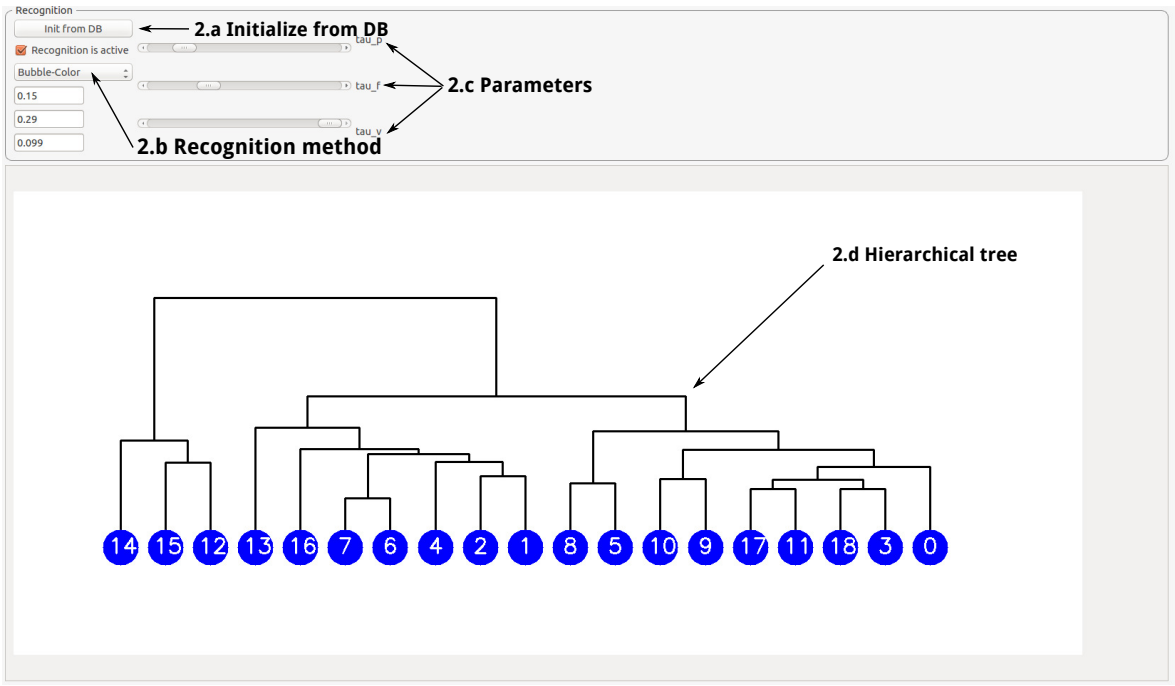


Figure B.2. Memory association module screen

through the experiment because parameter change would not affect already detected places. Parameters given in the extra settings (3.e) box for changing the display of the hierarchical tree.

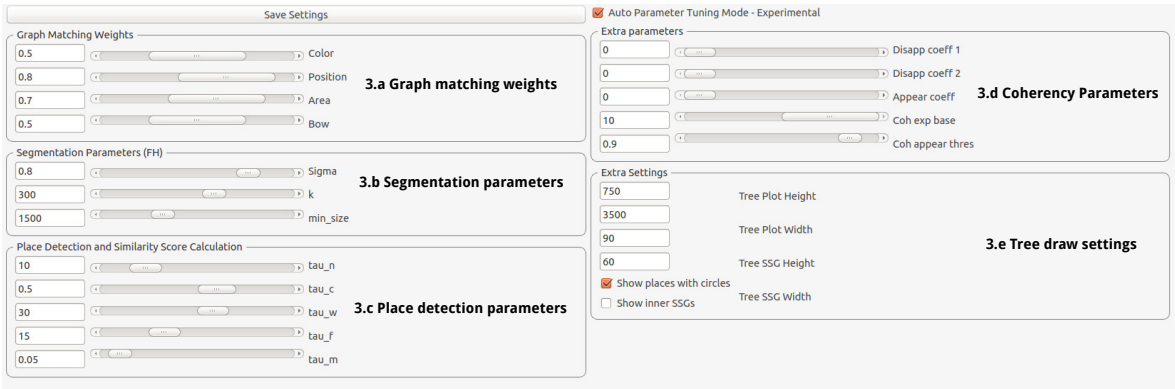


Figure B.3. Settings and parameters screen

All outputs and settings files are stored in the ‘Output’ folder which is located in the root. The folder contains settings files, datasets information file, databases, detection and maximal association results etc. Datasets.txt file contains the information about as available datasets and selection of the active dataset is done via the same file. It is possible to run multiple datasets consecutively. Parameters for each dataset

is stored in a separate settings file as located in the output folder. Databases are also stored in this folder. User can chose either load from already created database, create its own database based on the new place detection results or not create database at all. Last but not least, keep in mind that there are also various settings that can be changed through source code. 'defs.h' file contains some definitions that are used through the code mostly for graphical purposes.

The source code of the software is composed of a number of classes and functions. Functions such as Hungarian matching, clustering and graph-based segmentation are used as of-the-shelf algorithms and placed in separate files. The most important classes and their descriptions are given in Table B.1. For more detailed explanation about the algorithms please refer to the source code which is written as much self documented as possible.

Table B.1. List of important classes and their explanation.

Class	Explanation
`GraphMatch`	Node-to-node distance calculation, Hungarian assignment based graph matching and graph match drawing algorithms are implemented
`Association`	Various place recognition approaches and memory tree drawing functions are implemented
`Segmentation`	A helper function for graph-based segmentation algorithm is implemented. Node signatures are also created here.
`SegmentTrack`	Segment tracking and the construction of node existence matrix is performed in this class.
`SSGProc`	SSG and BD based place descriptors are constructed in this class
`TSCHybrid`	The main algorithm (`processImagesHierarchical()`) is implemented in this function. Plotting functions, place detection method and database read/write operations are also implemented in this class.
`Utils/UtilTypes`	Various utility functions and variable type definitions
`MainWindow`	An interface class between GUI and algorithms. Most of the algorithms are called from this class.