

PLACE RECOGNITION WITH AN ATTENTIVE ROBOT

by

Özgür Erkent

B.S., Mechanical Engineering, Middle East Technical University, 2001

M.S., Cognitive Science, Middle East Technical University, 2004

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

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2013

PLACE RECOGNITION WITH AN ATTENTIVE ROBOT

APPROVED BY:

Prof. H. Işıl Bozma
(Thesis Supervisor)

Prof. Lale Akarun

Prof. Yağmur Denizhan

Prof. Yorgo Istefanopulos

Prof. Hakan Temeltaş

DATE OF APPROVAL: 26.08.2013

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Işıl Bozma for the continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my Ph.D study.

I would like to thank Prof. Lale Akarun and Prof. Yağmur Denizhan for serving as a member of my thesis committee and their academic guidance. I am grateful to Prof. Yorgo Istefanopulos and Prof. Hakan Temeltaş for serving on my jury.

I also would like to thank Prof. Kadri Özçaldıran, Prof. Ayşın Ertüzün and Prof. Günhan Dünder for their motivation and academic guidance.

I would also like to extend my sincerest appreciation to my friends in Intelligent Systems Lab. I would like to thank Haluk Bayram for countless reasons. His intelligence and comradeship helped me to survive in this period of my life. I would also like to thank Berkay Deniz İlhan and Semih Çaycı for their help and contributions to my research with their brilliant ideas and for their friendships with their brilliant personalities. I would also like to thank Ahmet Unutulmaz and Oğuz Atasoy for their support in designing the Motor Control Boards for the robot and their sense of humor which helped us to overcome hardest problems with a smile on our faces. I would also like to thank Hakan Karaoğuz for his help on my research in various areas. A short break with him both forged life long friendships and served as a welcome time of rest.

I would like to sincerely thank Çağatay Soyer for providing answers to my questions regarding his past work at Intelligent Systems Lab.

I am grateful to Prof. Çetin Yılmaz and Prof. Emre Aksan for their help in mechanical design and manufacture of the robot APES. I would also like to mention

and thank Erdem Eren, Emre Korkmaz, and Murat Can Ülgüdür for their efforts in the mechanical design and manufacture process of APES.

Finally, there is no word to describe my gratitude toward my family for their endless support and love during my life.

ABSTRACT

PLACE RECOGNITION WITH AN ATTENTIVE ROBOT

This thesis is concerned with robotic place recognition. From the robot’s perspective, the goal of place recognition is to determine its whereabouts - given its knowledge of places it has visited previously. This problem is broken into four stages: First, the problem of how the robot should look around in the current place is considered and a model of attentive visual behaviours based on a family of artificial potential functions is developed so that the robot is able to consider the interesting points in the scene. Next, the problem of place representation is considered. For this, a topological model referred to as bubble space is developed. Sensory features associated with all the interesting points are encoded as bubble surfaces - in a manner that is implicitly dependent on robot pose. For compact representation, bubble surfaces are transformed into bubble descriptors that are rotationally invariant with respect to heading changes while being computable in an incremental manner as each new set of visual observations is made. Bubble descriptors are used for supervised place learning and recognition with support vector machines. Following, the problem of integration of sensory and semantic cues is considered and a representation referred to as “cue descriptor semantic forest” is proposed. The integration of different feature and semantic cues becomes very simple regardless of their type or their number while allowing relative and possibly partial semantic expressiveness via cue weights. Finally, the problem of unsupervised place learning is considered. A novel approach for the incremental and autonomous construction of bubble descriptor semantic trees is proposed. Comparative experiments on benchmark datasets as well as real-time robotic experiments indicate high recognition rates, dynamic with low computation times.

ÖZET

DİKKATE DAYALI BİR ROBOT İLE YER TANIMA

Bu tez robotik yer tanıma ile ilgilidir. Robotun bakış açısından, yer tanımanın amacı robotun nerede olduğunu belirlemektir - daha önceden ziyaret ettiği yerlerin bilgisinin verilmesi koşuluyla. Bu sorun dört aşamaya ayrılmıştır: İlk önce, robotun mevcut konumunda nereye bakması gerektiği ele alınmış ve yapay gizilgüç işlevlerine dayandırılan ilgiye bağlı görsel hareketler modeli geliştirilmiştir, böylece robot ortamdaki ilginç noktaları saptayabilmektedir. Daha sonra, yer gösterimi sorunu ele alınmıştır. Bunun için, baloncuk uzayı olarak adlandırılan bir topolojik harita geliştirilmiştir. Tüm ilginç noktalar ile ilgili duyuşsal özellikler robot konumuna dolaylı bir biçimde bağlı olarak baloncuk yüzeyleri üzerinde kodlanmıştır. Tıkız gösterim için, her yeni gözlemle birlikte baloncuk yüzeyleri robot yönelimine karşı dönel değişmez olan ve artırımlı olarak hesaplanabilen baloncuk tanımlayıcılara dönüştürölmektedirler. Baloncuk tanımlayıcıları destek vektör makineleri ile denetimli yer öğrenme ve tanıma için kullanılırlar. Ardından, duyuşsal ve anlamsal ipuçlarını birleştirme problemi ele alınmış ve “ipucu tanımlayıcı anlamsal ormanı” olarak adlandırılan bir gösterim önerilmiştir. Farklı özellik ve anlamsal ipuçlarının birleştirilmesi, bunların çeşitlerinden veya sayılarından bağımsız olarak oldukça kolay bir hale gelirken ipucu ağırlıkları sayesinde anlamsal açıklayıcılığına kısmen veya bağıl olarak olanak tanımıştır. Son olarak, denetimsiz yer öğrenme sorunu ele alınmıştır. Baloncuk tanımlayıcı anlamsal ağaçlarının artırımlı ve otonom inşası için yeni bir yaklaşım önerilmektedir. Denektaş veri kümelerinde ve gerçek-zamanlı robot üzerinde yapılan kıyaslamalı deneyler yüksek tanıma oranlarına ve düşük hesaplama sürelerine işaret etmektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	v
ÖZET	vi
LIST OF FIGURES	xi
LIST OF TABLES	xv
LIST OF SYMBOLS	xvii
LIST OF ACRONYMS/ABBREVIATIONS	xix
1. INTRODUCTION	1
1.1. Problem Statement	1
1.2. General Approach	2
1.2.1. Place Recognition & Attentive Vision	2
1.2.2. Place Representation in Topological Maps	4
1.2.3. Cue Integration in Topological Place Recognition	5
1.2.4. Autonomous Learning & Topological Maps	5
1.3. Contribution	5
1.4. Organization of the Thesis	6
2. ARTIFICIAL POTENTIAL FUNCTIONS BASED CAMERA MOVEMENTS IN ATTENTIVE ROBOTS	8
2.1. Related Literature	8
2.1.1. Computational Attentive Vision	9
2.1.2. Robotic Vision Systems	10
2.2. Camera Movement Dynamics	11
2.2.1. Camera Motion Dynamics	12
2.3. Artificial Potential Functions	13
2.3.1. Saliency Features	14
2.3.1.1. Bottom-up Features	15
2.3.1.2. Top-down Features	17
2.3.2. Inhibition	18
2.4. Scanpath Algorithm	19

2.5. Experimental Results	22
2.5.1. Free Viewing Experiments	24
2.5.2. Detecting Visual Onsets	29
2.5.3. Search Experiments	31
2.5.4. Summary	33
2.6. Conclusion	34
3. BUBBLE SPACE AND PLACE REPRESENTATION IN TOPOLOGICAL MAPS	36
3.1. Related Work	37
3.2. Bubble Space	40
3.2.1. Bubble Surfaces	41
3.2.2. Bubble Descriptors	47
3.3. Learning & Recognizing Places	48
3.4. Experimental Results	51
3.4.1. Indoors Experiments: COLD Data Set	52
3.4.2. Outdoors Experiments: New College Data Set	60
3.4.3. Both Indoors and Outdoors: Comprehensive Data Set	66
3.4.4. Summary	67
3.5. Conclusion	68
4. CUE DESCRIPTOR SEMANTIC FORESTS FOR TOPOLOGICAL PLACE RECOGNITION	70
4.1. Related Literature	72
4.1.1. Sensory Feature Cues	72
4.1.2. Semantic Cues	73
4.2. Sensory Feature Cue Descriptors	74
4.2.1. Cue Descriptors	75
4.3. Cue Descriptor Semantic Forest	76
4.3.1. Semantic Cues	76
4.3.2. Cue Descriptor Semantic Trees	77
4.4. Place Learning	78
4.4.1. CDSF Construction	79

4.4.2. Learning Cue Weights	79
4.5. Cue Integration for Recognition	82
4.5.1. Sensory Feature Cue Accumulation	83
4.5.2. Semantic Cue Propagation	83
4.6. Experiments	84
4.6.1. Visual Cues - Indoors & Outdoors	87
4.6.2. Adding 2D Laser Cue	89
4.6.3. Missing Sensory Cues	91
4.6.4. Effect of Occlusion	92
4.6.5. Summary	94
4.7. Conclusion	94
5. UNSUPERVISED AND INCREMENTAL BUBBLE DESCRIPTOR TREE CON- STRUCTION	96
5.1. Related Literature	97
5.1.1. Place Learning	97
5.1.2. Data Organization	98
5.2. Bubble Descriptor Semantic Trees	100
5.2.1. BDST Structure	100
5.2.2. Recognition	101
5.3. Incremental BDST Update	102
5.3.1. Adding Nodes	102
5.3.2. Merging of Nodes	104
5.3.3. Update of SVMs	105
5.4. Experiments	105
5.4.1. Incremental Tree Construction	106
5.4.2. Incremental & Unsupervised Tree Construction	110
5.4.3. Comparison of Supervised vs Unsupervised Tree Construction .	113
5.4.4. Summary	115
5.5. Conclusion	116
6. CONCLUSION	117
APPENDIX A: APES G3 - ATTENTIVELY PERCEIVING SYSTEM	120

A.1. Hardware	120
A.1.1. Mechanical Details	120
A.1.2. Electrical Details	122
A.1.3. Electronic Details	124
A.2. Software	124
A.2.1. Required Software	124
A.2.2. How To Use Software	124
A.3. Camera Head Dynamics	126
APPENDIX B: APES G3 - USER GUIDE	128
B.1. Power Connection	128
B.2. Main Board Power Up	128
B.3. Motor Control Board Power Up	132
B.4. Connection to APES	133
B.4.1. Direct Connection	133
B.4.2. Remote Connection	133
APPENDIX C: VISUAL FEATURES	134
APPENDIX D: COMPUTATION OF BUBBLE SURFACE COEFFICIENTS	137
APPENDIX E: BUBBLE DESCRIPTORS & ROTATIONAL INVARIANCE .	140
APPENDIX F: PUBLICATIONS RELATED TO THESIS	143
REFERENCES	145

LIST OF FIGURES

Figure 1.1.	The map of the site.	3
Figure 1.2.	The general approach used in the thesis.	4
Figure 2.1.	One-dimensional view of the configuration space.	11
Figure 2.2.	Closed-loop APF based camera movements.	14
Figure 2.3.	The construction of a sample $\varphi_{k,p}$ evaluated on $P_{f(t_k)}$	19
Figure 2.4.	Time evolution of visual fields from k th fixation to the next. . . .	20
Figure 2.5.	Sampled visual field map values and the resulting trajectory. . . .	22
Figure 2.6.	APES Robot. Complete robot and the camera head.	22
Figure 2.7.	Free viewing.	25
Figure 2.8.	Scenes shown to the robot in the free viewing experiments.	26
Figure 2.9.	HR and MR in free viewing.	28
Figure 2.10.	Comparative fixations under free viewing.	29
Figure 2.11.	Detecting a red visual onset on a monitor.	30
Figure 2.12.	Detecting a red visual onset in the lab.	32

Figure 2.13.	Search experiment scenes $v(f(0), \cdot)$ with the searched feature. . . .	33
Figure 2.14.	Search for yellow.	34
Figure 3.1.	Robot's base point and viewing directions.	40
Figure 3.2.	A robot at a given base point.	42
Figure 3.3.	The deformation of bubble surface from a particular robot base. .	45
Figure 3.4.	Original and reconstructed bubble surface.	46
Figure 3.5.	A sample recognition task Freiburg Lab.	53
Figure 3.6.	Precision recall curves for varying test scenarios.	59
Figure 3.7.	A route in New College data set segmented into 7 places.	61
Figure 3.8.	Sample images from New College data set.	62
Figure 3.9.	Precision-recall curves for varying learning and recognition. . . .	66
Figure 3.10.	Precision recall curves for the comprehensive data set.	67
Figure 4.1.	Representation in bubble space.	75
Figure 4.2.	A CDST structure with $N_L = 2$	78
Figure 4.3.	Corresponding nodes for level $l = 1$ in a 2-CDSF.	80
Figure 4.4.	Hierarchical propagation of semantic cues.	84

Figure 4.5.	Cues that are used in the experiments.	85
Figure 4.6.	Sample images.	86
Figure 4.7.	A CDST with semantic cues.	88
Figure 4.8.	The locations of the 2D laser sensors.	89
Figure 4.9.	Sample occluded scene.	92
Figure 4.10.	Recognition rates with varying occlusion.	93
Figure 5.1.	SLINK Algorithm.	103
Figure 5.2.	The original tree and after inner-node merging.	104
Figure 5.3.	Labels from supervised datasets.	107
Figure 5.4.	Incremental trees constructed with supervised labels.	108
Figure 5.5.	Labels from unsupervised datasets.	111
Figure 5.6.	Incremental tree construction.	112
Figure A.1.	A general view of the APES G3.	122
Figure A.2.	A 3D model of the active vision systems.	122
Figure A.3.	Wiring diagram of APES.	123
Figure A.4.	(a) Image plane orientations; (b) Image transformation.	126

Figure A.5.	Computation of gradient vector.	127
Figure B.1.	The power connection for main board.	129
Figure B.2.	The power connection for motor control boards and the camera. .	130
Figure B.3.	The power button for the main board of APES.	131
Figure B.4.	On/off switch for MCB is pointed out.	132
Figure C.1.	Visual feature filters.	136

LIST OF TABLES

Table 2.1.	APF parameter values for different visual behaviors.	23
Table 2.2.	Free viewing results with $N_S = 15$ fixations.	27
Table 2.3.	Comparative results in free viewing.	29
Table 2.4.	Results in detecting visual onsets.	31
Table 2.5.	Results of search experiments.	35
Table 3.1.	The comparison of four approaches.	54
Table 3.2.	Recognition rates for Freiburg Lab.	56
Table 3.3.	Recognition rates for Saarbrücken Lab.	57
Table 3.4.	Recognition rates for Ljubljana Lab.	58
Table 3.5.	Recall rates for specified precision levels.	60
Table 3.6.	Recognition rates for varying learning and recognition scenarios. .	63
Table 3.7.	Confusion Matrix.	63
Table 3.8.	Confusion Matrix when learning under day illumination.	64
Table 3.9.	Recall rates for specified precision levels as given in Figure 3.9. . .	65

Table 3.10.	Recall rates for specified precision levels with an unknown place.	65
Table 3.11.	Recognition rates for the comprehensive data set.	67
Table 4.1.	Recognition rates - BuS vs CDSF.	88
Table 4.2.	Learning times with newly added laser cue.	90
Table 4.3.	Recognition rates with the added laser cue.	90
Table 4.4.	Recognition rates with missing color cues.	91
Table 5.1.	Place Learning.	97
Table 5.2.	Tree Construction.	98
Table 5.3.	Recognition Rates.	110
Table 5.4.	Recognition Rates for Unsupervised Labels.	112
Table 5.5.	Label Descriptions.	114
Table A.1.	Technical Details of APES.	121

LIST OF SYMBOLS

A_f	Fovea
b	Bubble space point
$\mathcal{B}$	Bubble space
B	Bubble surface
c	Robot's position
$\mathcal{C}_{cp}$	A set of cue descriptors
f	Robot's viewing direction
$\mathcal{F}$	Robot's viewing direction state space
$g_b(o)$	Bump function
I	Bubble descriptor
I_c	Cue descriptor
M_p	Number of base points in a place
$\mathcal{M}_s$	Short-term memory
N_c	Number of cues
N_s	Number of fixations
N_v	Number of visual filters
$\mathcal{P}$	Set of places
$q(b, t)$	Observation vector
$S_{c,p^{l-1}}$	Distance matrix for cue weights
u_k	Control input
$\mathcal{V}$	Visual features
w_{1_j}	Weight coefficients in saliency term
$W_{p^{l-1}}$	conditional probability matrix for cue weights
x	Robot's base point
$\mathcal{X}$	Robot's base point space
Z_p	normalization term for composite matrix

α	Robot's heading
Δ	Tree based on bubble descriptors
$\varphi_{k,p}$	Artificial potential function
κ_{c,p^l}	Cue weight
ρ_i	Riemannian metric of the response to the i^{th} feature
ω_k	Visual filter frequency

LIST OF ACRONYMS/ABBREVIATIONS

AHC	Agglomerative Hierarchical Clustering
APF	Artificial Potential Function
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
BuS	Bubble Space
CDSF	Cue Descriptor Semantic Forest
CDST	Cue Descriptor Semantic Tree
CPAM	Coloured Pattern Appearance Model
HCT	Hull Census Transform
HIST	Histogram
HSI	Hue Saturation Intensity Color Space
IK	Inverse Kinematics
PID	Proportional-Derivative-Integral
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

1. INTRODUCTION

This thesis is concerned with robotic place recognition. From the robot's perspective, the goal of place recognition is to determine its whereabouts - given its knowledge of places it has visited previously. Place recognition is crucial to map building and spatial reasoning with mobile robots [1]. Historically, map building started with the geometric modeling of free-space in the early eighties [2]. However, as metric representations were shown to have problems in large unstructured environments with high computational requirements [3], roboticists came to realize the importance of more abstract representations [4]. Topological maps - motivated by this - describe space as a collection (graph) of places (nodes) linked by adjacency relationships (edges) [5]. This is still an active research area as there is no consensus regarding the way they are built [5]. A key element in topological maps is the place representation and recognition module [6] and place recognition is still an important aspect of the problem [7–9].

1.1. Problem Statement

Consider a robot as seen in Figure 1.1. Suppose the robot navigates through different places such as parking lot or offices as seen in Figure 1.1. The question is can the robot know where it is at a given moment? In this thesis, this problem was found to have four critical aspects to it: perception, representation, cues and learning.

- First, perception was considered. In particular, the problem of how the robot should look around in the current place is studied. Attentive vision has been shown to be a solution for robotic systems in 1992 [10]. Therefore, a solution based on attentive vision can be used to detect interesting points in the scene.
- Next, the problem of place representation is considered. Space representation is crucial to map building and spatial reasoning with mobile robots [1] and there is a growing interest in augmenting or possibly replacing purely metric maps with topological representations [5]. The proposed representation model should associate sensory features with the interesting points which are implicitly dependent

on robot pose. This representation should have a compact structure for ease of computation and storage. It should also be compatible with a machine learning method with reasonable recognition rates.

- As a third point, the proposed representation should be used with different types of sensory and semantic cues. Each cue conveys a separate aspect of the scene which may be either redundant or complementary in nature with individual limits on accuracy and availability. The integration should be simple regardless of type or number of cues.
- Finally, the problem of unsupervised place learning is considered. It should construct a knowledge base of the places in the environment in such a way that if the scene has not been observed before, then it should be inserted into the training set and the new data should be organized dynamically for obtaining an updated representation of the environment [6].

1.2. General Approach

In this perspective, the problem is broken into four stages as shown in Figure 1.2. The first stage addresses the "where to look problem" by using free viewing visual behaviour with attentive vision. The interesting sensory features are collected for place representation. The developed place representation model is capable of incorporating data from different sensors. This representation is then used together with previous place knowledge for recognition. For more robust and efficient recognition, this representation is extended to integrate sensory and semantic cues. If the place is found to be a new place, then it is learned in an autonomous manner and added to the already existing knowledge base.

1.2.1. Place Recognition & Attentive Vision

Perception is critical to place recognition. It is defined by Kant [11] as "representation accompanied by sensation". A robot endowed with vision capabilities collects images to recognize its environment. Processing all regions of an image for place recog-

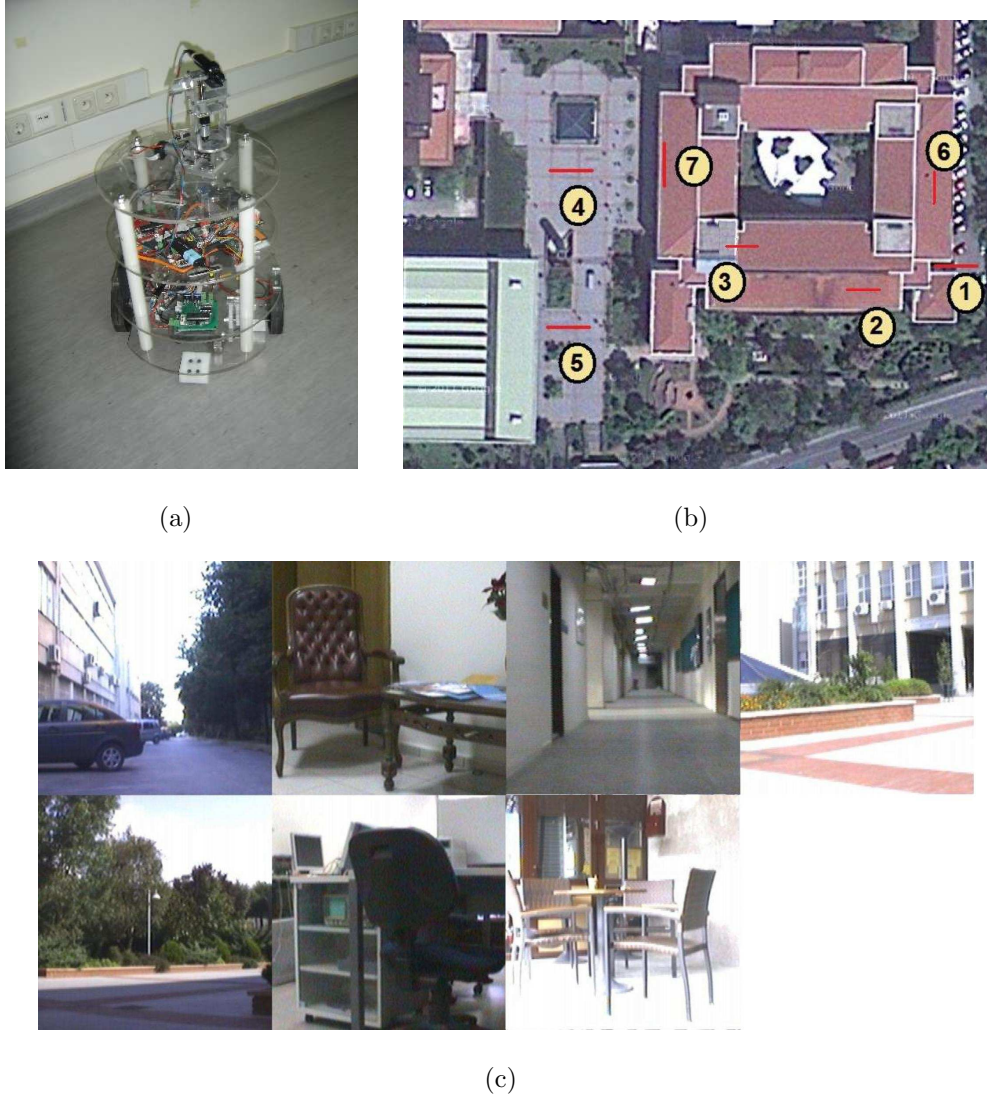


Figure 1.1. (a) APES robot; (b) The map of the site containing 7 different places. (c) Sample scenes from the site.

tion requires high computational complexity, however real-time operability requires low computational complexity. Therefore, the robot must concentrate on data containing maximum information. Attentive vision has been shown to be a solution for robotic systems [10]. An alternative closed-loop approach for attentive robots based on artificial potential functions has been proposed in [12]. This method has been shown to fixate to selected points in a minimal time. However, its feature set does not contain complex features. Therefore, a method with a complex feature set implemented with a closed-loop approach based on artificial potential functions can extract the salient regions of a place in a real-time system.

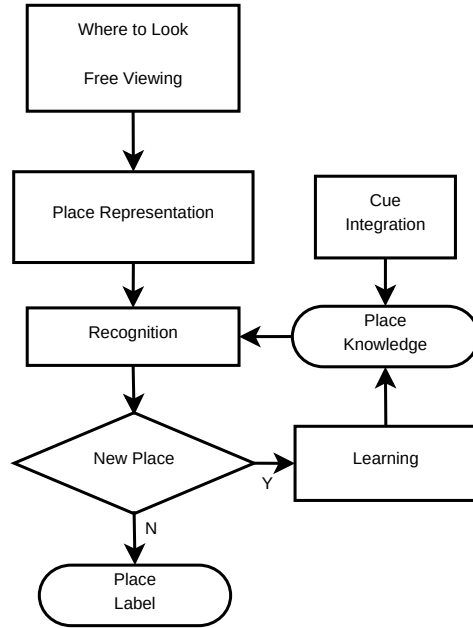


Figure 1.2. The general approach used in the thesis.

1.2.2. Place Representation in Topological Maps

Visual place representation in topological maps is an important problem for robotics. The images are acquired either via a two degrees of freedom camera head or an omnidirectional camera mounted on a mobile robot. This problem is of great interest to the machine vision and robotics communities [8]. The proposed approaches either represent each place directly based on visual data or first find a set of image features which are then used to generate the final representation. A second approach is taken in this thesis using a set of feature observations along with their associated viewing directions in this representation. The starting point is the “bubble memory” where visual feature values and their local S^2 –metric relations are simultaneously encoded from an egocentric perspective [13,14]. In contrast to most of the representations with either high computational complexity or insufficient representation ability, bubble space representation has low computational complexity while being rotationally invariant.

1.2.3. Cue Integration in Topological Place Recognition

A robot may not always be equipped with only a camera. It may also use other sensors/cues for recognizing places. It is known that new cues provide higher performance in place recognition [15,16]. In this thesis, the focus is a related aspect – namely integrative cue representation. The robot should be able to incorporate any type of cues - sensory feature or semantic. As such, introducing sensory feature cues should not require the internal representations to be rebuilt from scratch [17]. It should rather be possible to update the representation incrementally. Simultaneously, the representation should enable graceful degradation of performance in case of missing sensory cues or observations. In this thesis, an approach endowed with these features is developed.

1.2.4. Autonomous Learning & Topological Maps

Place representation models deal with two main problems: detecting a novel place and updating the model’s knowledge base. In this thesis, the usage of trees for place representation is motivated by graph-based algorithms in place recognition. Most of the place representation models collect the novel places in a supervised manner and use a static graph representation, i.e. their knowledge representation cannot be updated in an incremental manner [18,19]. On the other hand, some representation models detect the places in an unsupervised manner, however they cannot update their own knowledge base in a dynamic manner described in [20]; whereas some models detect the places in an unsupervised manner, but they cannot update their own graph representations in an incremental manner [21]. In this study, a method will be implemented which combines detecting novel places in an unsupervised manner with a dynamic representation.

1.3. Contribution

The major contributions of this thesis can be summarized as follows:

- (i) Attentive Vision: A novel integrated model of visual behaviors based on artificial potential functions is proposed. In this model, the robot can autonomously

switch between the behaviours of “idle looking”, “search”, “visual onset detection”, “tracking” and “remain fixated” simply by changing the parameters of the artificial potential functions accordingly - depending on the visual feedback from the environment.

- (ii) Place Representation: A topological model referred to as bubble space is developed. Sensory features associated with all the interesting points are encoded as bubble surfaces - in a manner that is implicitly dependent on robot pose. For compact representation, bubble surfaces are transformed into bubble descriptors that are rotationally invariant with respect to heading changes while being computable in an incremental manner as each new set of visual observations is made. Bubble descriptors are used for supervised place learning and recognition with support vector machines.
- (iii) Cue Integration: A representation referred to as cue descriptor semantic forests is proposed. The integration of different feature and semantic cues becomes very simple regardless of their type or their number while allowing relative and possibly partial semantic expressiveness via cue weights.
- (iv) Autonomous learning of places: A novel approach is proposed that enables the robot to learn places in topological maps in a completely unsupervised and incremental manner.

1.4. Organization of the Thesis

The rest of this thesis is organized as follows.

In Chapter 2, a model of how the robot looks around in the current place is developed. This model incorporates a plethora of different visual behaviors including free viewing, detecting visual onsets, search, remaining fixated and tracking. In this model, the generation of camera movements is based on a closed-loop dynamical systems as defined via a family of artificial potential functions. Experimental results with APES robot serve to show the performance properties of a robot engaged in each different visual behavior. For the interested reader, the APES robot is described in Appendix A with its user guide presented in Appendix B respectively.

Chapter 3 presents bubble space based representation of “places” (nodes) in topological maps. Bubble space is a representation that simultaneously allows for detailed (bubble surfaces) and holistic (bubble descriptors) description of places. It is based on bubble memory where visual feature values and their local S^2 -metric relations from robot’s viewpoint are simultaneously encoded on a deformable spherical surface. Bubble surfaces extend bubble memory to accommodate varying robot pose and multiple features. They are transformed into bubble descriptors that are rotationally invariant with respect to heading changes while being computable in an incremental manner as each new set of visual observations is made. The visual features used to obtain bubble surfaces are explained in Appendix C. Incremental computation of bubble surface coefficients is explained in Appendix D. Finally, the rotational invariance of bubble descriptors is proved in Appendix E.

In Chapter 4, the integration of different cues is considered and a novel approach based on cue descriptor semantic forests is developed. In this representation, each sensory feature related cue is encoded by a cue descriptor. Cue descriptors are amalgamated with an ordered set of semantic cues and cue weights via cue descriptor semantic trees - thus yielding a cue descriptor semantic forest. Comparative experiments on benchmark datasets provide insight on learning times, recognition rates and performance in case of missing or partially available sensory feature cues.

In Chapter 5, autonomous learning of places is studied and an unsupervised and incremental approach to the construction of bubble descriptors semantic trees is presented.

The thesis concludes with a brief summary and a discussion regarding future work as presented in Chapter 6.

2. ARTIFICIAL POTENTIAL FUNCTIONS BASED CAMERA MOVEMENTS IN ATTENTIVE ROBOTS

This chapter presents a model of how the robot can look around in the current place via switching among plethora of different visual behaviors including free viewing, detecting visual onsets, search, remaining fixated and tracking. In this model, the generation of camera movements is based on a set of closed-loop dynamical systems as defined via a family of artificial potential functions. Each movement from the current fixation point to the next is associated with an artificial potential function that encodes saliency and possibly inhibition depending on the visual behavior that the robot is engaged in. The novelty of this approach is that since the nature of resulting motion can vary from being saccadic to smooth pursuit, the full repertoire of visual behaviors all become possible within the same framework. The robot can switch its visual behavior simply by changing the parameters of the constructed artificial potential functions appropriately. Furthermore, automated reflexive changes among the different visual behaviors can be achieved via a simple switching automaton.

The organization of the Chapter is as follows: The rest of this section presents a brief summary of related literature. The motion dynamics based on artificial potential functions are presented in Section 2.2. The construction of the artificial potential functions is developed in Section 2.3. The scanpath algorithm is presented in Section 2.4. Experimental results with APES robot engaged in different visual behaviors are discussed in Section 2.5. The Chapter finally concludes with a brief summary.

2.1. Related Literature

Related literature from two areas is reviewed: computational attentive vision and attention in robotics.

2.1.1. Computational Attentive Vision

Inspired by visual findings, many researchers in computer vision and robotics have come to view attention to be integral to vision [22, 23]. Attentive systems operate in a loop of pre-attentive and attentive processing where the goal of pre-attention is to find salient regions in the incoming image while in attention, only these salient regions, referred to as fovea, are viewed in high resolution and are subjected to detailed scrutiny. Thus, limited resources are allocated to process only the most relevant parts of the incoming image as opposed to its whole.

Most models of pre-attention are based on the construction of a two-dimensional map that topographically encodes interesting regions as surveyed in [24, 25]. What differentiates these maps is the exact manner in which they are constructed. In saliency maps, a set of features are combined into a single representation. While some saliency maps are simple, others are relatively much more complex and incorporate a large number of features. For example, 42 feature maps including 6 intensity, 12 color, and 24 orientation features are combined individually first and then summed up for bottom-up saliency maps [26]. Detection of visual onsets is based on the computation of temporal differences of consecutive images filtered at various spatial scales [27].

Learned representation based on weights of the features stored in the long-term memory enable the generation of task related maps which are in turn used to bias saliency maps towards specific a priori learned targets for top-down pre-attentive behavior [28]. Another approach is based on a probabilistic framework where these maps are defined by likelihood that a set of features (that could also be a target) will be present at a given location [29–31]. For example, in attention-based information maximization, a set of independent component analysis coefficients extracted from each image are used to first construct coefficient distributions and then joint likelihoods which eventually yield an information map that encodes the saliency of each spatial location [30]. The novelty of this approach is that - in contrast to saliency maps - visual salience is not based solely on local features, but also their distribution on the whole image. Discriminant saliency based on the mutual information between a set of

features and class label extends decision-theoretic formulation work to top-down processing is presented in [29]. It should be noted that with some of these approaches, as being real-time is not a main concern, the computational complexity of saliency computation can be quite high [32]. Secondly, in most work, inhibition is accomplished by the mechanism that is responsible for computing the most salient location with notable exception [12] where the map encodes saliency and inhibition simultaneously.

Once a map is constructed (either explicitly or implicitly), the second issue pertains to determining the most salient location in this map. Most approaches propose search mechanisms that generate emergent sequences on a single image. A hierarchical approach is taken in [27] where the computation is based on spatial competition that is improved at each level of processing via feed-forward and feedback mechanisms. A neural model with built-in non-classical surround modulation effects conceives attention as the facilitation of certain set of neurons (whose outputs are set as determined by the saliency map) in a manner that highlights particular features at a particular position in the visual field [26]. An alternative closed-loop approach based on artificial potential functions has been proposed to model saccadic motion in [12]. Finally, explicit sequences where ‘memorized’ sequence representations are retrieved in a manner analogous to skilled motor behavior based on augmented hidden Markov models are presented in [33].

2.1.2. Robotic Vision Systems

Attentive vision has been deemed to be necessary for robotic systems [10] as low computational complexity is a major concern for real-time operability. Interestingly, attention has been found to be closely related to other visual tasks such as object recognition [34, 35]. Many robots or robotic heads endowed with attentive motion capabilities have been developed in order to collect the visual data accordingly. Separate mechanisms have been proposed [14, 36–45].

In most of this work, the body of the robot is stationary while the head is moving with some notable exception [46]. Meanwhile, the selection of the target points and

the motion necessary for moving to them are treated independently and an open-loop motion control is employed. An “outer” control loop first determines where the next target point is and then this point is given as an input to a second (“inner”) control loop that realizes the camera motion. In this Chapter, the search is considered to be realized via a single loop that realizes ”where” and ”motion” simultaneously.

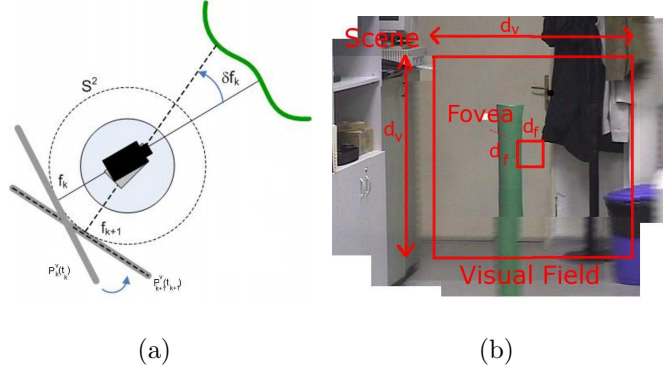


Figure 2.1. (a) One-dimensional view of the configuration space $\mathcal{F}$, fixation points $f_k, f_{k+1} \in \mathcal{F}$ and the associated image planes P_{f_k} and $P_{f_{k+1}}$. (b) A visual field $v(f, .)$ of size $d_v \times d_v$ defined on the image plane P_f centered around the fixation point f .

2.2. Camera Movement Dynamics

Consider a robot engaged in a specific visual behavior. Let its camera’s state space (pan and tilt movements) be denoted by $\mathcal{F} \subset S^2$ where S^2 is the unit sphere. Suppose that at instant t_k , the robot is fixated at $f_k \in \mathcal{F}$. The camera’s image plane P is a $d_v \times d_v$ sized region centered at the origin. Let $P_f = \{f\} \times P$ denote the image plane whose origin is tangent to $\mathcal{F}$ at f . The central much smaller $d_f \times d_f$ sized region in P is known as fovea A . Let $A_f \subset P_f$ defined as $A_f = \{f\} \times A$ denote the fovea associated with the current image plane P_f . While the robot is engaged in its visual behavior, its camera moves from one fixation point f_k to the next f_{k+1} as seen in Figure 2.1a and generates a scanpath with a set of fixation points $\{f_k\}$, $k = 1, \dots, N_S$ where N_S denotes the number of fixations made. The visual field $v : \mathcal{F} \times P \rightarrow R^n$ denotes the n -dimensional sensory data collected from each image plane P_f as seen in Figure 2.1b. In particular, in case of HSI (hue, saturation, intensity) color space representation, $n = 3$ and each visual field value is 3-dimensional $v(f, x) = [H_f(x) S_f(x) I_f(x)]^T$. The

hue component H_f is the primary color sensed associated with the mean wavelength, the colorfulness S_f is related to the width of its spectral density and the brightness I_f is related to intensity [47].

2.2.1. Camera Motion Dynamics

The camera state from the current fixation point f_k at time t_k to the next is given by $f : [t_k, t_{k+1}] \rightarrow \mathcal{F}$. The corresponding system dynamics is associated with an artificial potential function $\varphi_{k,p} : [t_k, t_{k+1}] \times P \rightarrow [0, 1]$. The idea of using artificial potential functions for programming robots has been pioneered in the context of navigation and obstacle avoidance and its use has since then gained acceptance in the robotics field [48, 49]. Each $\varphi_{k,p}$ encodes saliency and possibly inhibition on the current image plane $P_{f(t)}$ depending on the parameter vector $p \in R^{N_P}$. The negative gradient field $\varphi_{k,p}$ evaluated at the origin of $P_{f(t)}$ automatically gives rise to a control input for the camera's actuators. Thus, we let the camera dynamics be governed by:

$$\dot{f}(t) = -u_k(t, 0) \quad (2.1)$$

where $u_k : [t_k, t_{k+1}] \times P \rightarrow \mathcal{F}$ is the control input defined as:

$$u_k(t, x) = D_x \varphi_{k,p}(t, x) \quad (2.2)$$

Note that in order to move the camera, the gradient vector which is of Cartesian nature needs to be transformed to the angular velocities - namely $\dot{f}$. However, geometrically, it is easy to see that the angular velocity components are directly proportionally to the Cartesian velocity components [12]. The corresponding closed-loop control system is as shown in Figure 2.2. The robot's camera starts navigating away from the current fixation until

$$u_k(t, 0) = 0 \quad \text{where } t_k < t \leq t_{k+1}$$

Thus, camera motion is accomplished simply by “sliding” into the equilibrium point of the associated dynamical system. The equilibrium point is then designated as the next fixation point $f_{k+1} = f(t_{k+1})$. Note that the termination time t_{k+1} is defined implicitly by the time taken to reach to f_{k+1} . This closed loop system inherits the critical qualitative behavior of gradient trajectories. If $D_x u_k(t_{k+1}, x)$ has full rank at the newly arrived fixation point f_{k+1} , then the equilibrium point is an isolated singularity. Otherwise, it can have a nontrivial manifold of minima which means that small perturbations in the location of the current fixation point could result in large deviations on the location of the next fixation point and hence leading to different camera movements. The details of the Inverse Kinematics (I.K. in the Figure 2.2 can be found in Appendix A).

Note that in contrast to open-loop approaches where the next fixation point is given explicitly, in this approach, it is defined implicitly through system dynamics. Hence, the questions of where and how are solved in a single unified manner. As $\varphi_{k,p}$ is time-variant, this in turn increases the visual responsiveness of the robot. Hence, ideally, the robot is able to respond to scene and hence saliency changes automatically while a movement is in progress. Let it be noted that this is not the case in human vision where saccadic suppression (no perceiving during a saccade) removes the stimulus from awareness and hence visual events such as an object moved to a new position during a saccade become rarely noticed [50, 51]. However, this does not impede the stimulus from being processed by the visual system and hence can still affect what the observer sees after the eye movement [51]. Interestingly, in practice, the proposed system has saccadic suppression to a limited extent. This is because as the incoming images are not continuous, but rather acquired consecutively one after the other as explained in Section 2.4, the robot is blind to changes between consecutive images.

2.3. Artificial Potential Functions

Each $\varphi_{k,p}$ encodes saliency and possibly inhibition on the current image plane $P_{f(t)}$ for $t \in [t_k, t_{k+1}]$ depending on the robot’s visual behavior. The dependency on visual behavior is realized through the parameter vector $p \in R^{N_p}$. It consists of two

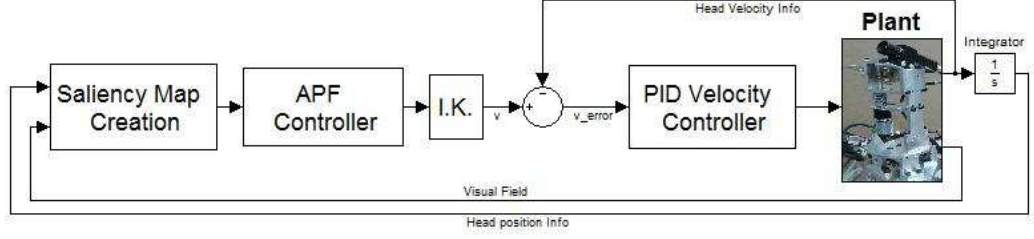


Figure 2.2. Closed-loop APF based camera movements. Note that the control system realizes “where” and “motion” simultaneously.

terms as follows:

$$\varphi_{k,p}(t, x) = \beta_p(t, x) \gamma_{k,p}(t, x) \quad (2.3)$$

The first term β_p encodes a set of weighted saliency features that draw attention on the current visual field. It is comprised of bottom-up and top-down terms that encode various features. The second term $\gamma_{k,p}$ incorporates the inhibition mechanism that is integral to saccades. This is unlike most saliency map formulations where inhibition is activated during the computation of the most salient location. The construction of these terms are done such that each APF is minimal at the potential next fixation points. This is preferred over being maximal since the flow field properties of gradient systems dictate that maximal points are unstable equilibrium points which means under little perturbations the system starts moving away from these points.

2.3.1. Saliency Features

The saliency term β_p encodes a set of “attractive” features in a product form:

$$\beta_p(t, x) = \prod_j \beta_j^{w_j}(t, x) \quad (2.4)$$

Each β_j measures a different saliency feature and should be relatively simple to

compute in order to minimize computational complexity and thus be real-time applicable. Hence, the simplest saliency features possible are used. Of course, more complex formulations for both bottom-up and top-down saliency terms could be used such as complex information maps for bottom-up saliency and task-related saliency maps as explained in Section 2.1.1 could also be used - with the understanding that there is a fundamental trade-off between increased complexity and lower computation. Of course, with more efficient algorithms or improved (faster, yet having low power requirements) hardware this can be possible. Furthermore, let us also note APF formulation incorporates inhibition so that saliency of recent locations is automatically decreased. This is unlike most saliency map formulations where the map does not have built-in inhibition information, but rather inhibition is activated during the computation of the most salient location. The weights w_{1_j} are components of the p parameter vector. Each determines the contribution of the corresponding feature to the overall saliency term β_p - namely whether the particular feature is turned on or not and if turned on, how dominating it is. Different saliency features need to be used for different visual behaviors and can be controlled by setting the weight values w_{1_j} appropriately. In broad categorization, saliency features can be categorized broadly into bottom-up and top-down features.

2.3.1.1. Bottom-up Features. Bottom-up (BU) features are those that will draw attention without any particular vision cognition task at mind while being relatively simple to compute. Three bottom-up features are considered - namely corners, color and motion. However, let it be noted that this set could easily be extended to include more features - including more complex features such as Cartesian and Non-Cartesian filters with varying frequencies, joint likelihood of features as well as top-down task-related features.

The first feature β_1 is BU corner [52]. Corner detection operator D_c can be implemented using an efficient and fast algorithm with high repeatability rate such

as [53]. Hence, BU corner saliency is expressed as:

$$\beta_1(t, x) = \frac{1}{1 + D_c(v(f(t), x))} \quad (2.5)$$

If a pixel x in the current image plane $P_{f(t)}$ has a corner, $\beta_1(t, x)$ attains a minimum value. Otherwise, its value is close to 1.

The second saliency feature β_2 is BU Color. BU Color is another feature that is known to draw attention [54]. It can be mathematically formulated based on measuring the amount of contrast in hue values H_f as:

$$\beta_2(t, x) = \frac{1}{1 + \nabla H_{f(t)}(x)^T \nabla H_{f(t)}(x)} \quad (2.6)$$

The greater is the color change at a point, the smaller is $\beta_2(t, x)$.

BU Motion is another dominant feature in drawing attention [55] and is measured by the function β_3 . We assume that the robot's motion is slow compared to its head fixations so that the robot acquires images almost in a continual manner.

A dynamic region in a given scene will in general induce a corresponding change in the visual field¹. This is encoded by the difference of a pair of consecutive visual field map values $v(f(t), \cdot)$ and $v(f(t - \delta t), \cdot)$ – in particular via the changes in intensity components $I_{f(t)}$ and $I_{f(t - \delta t)}$. The visual field may be changing due to two primary reasons – robot's body and head motion which is self-known or moving objects in the scene. Of course, both can occur simultaneously. The robot needs to weed out the effects of its self-motion in its changing visual field. If the robot's body is translating, as the translation amount can be assumed to be small with sufficiently frequent sampling of the visual field, it can be ignored and hence two consecutive visual fields will almost be identical. However, the effects of the rotational movement of robot's

¹Of course, excluding ambiguous cases such as that leading to the aperture problem.

camera (regardless of whether due to head or body motion) will need to be taken into account. As the visual field represents a small region at the center of the image plane, lens distortion effects can be neglected and the rotation of the camera is approximately equivalent to a translation T_m of the visual field.

If $I_{f(t)}(T_m(x))$ denotes the visual field translated with this amount, this difference can be estimated by $\psi(t, x) = |I_{f(t-\delta t)}(x) - I_{f(t)}(T_m(x))|$. Assuming that rotational movements are also small, in case of only self motion, this difference will be small – namely $\psi(t, x) \leq \tau$ where τ is the threshold for the motion detectability. In case of dynamic scene, this difference will be large. Hence, the threshold τ enables the suppression of irrelevant salience. Thus, BU motion saliency can be measured as:

$$\beta_3(t, x) = \begin{cases} 1 & \text{if } \psi(t, x) \leq \tau \\ \frac{1}{\psi(t, x)} & \text{otherwise} \end{cases} \quad (2.7)$$

In regions of the image with motion, $\beta_3(t, x)$ will become smaller.

2.3.1.2. Top-down Features. Top-down (TD) features are those that will draw attention with a particular visual cognition task at mind while again being relatively simple to compute. One very common feature is TD color. However, again, TD features can be easily extended to incorporate a richer set. TD Color is defined differently from BU color. Again it is based on the hue component of the incoming visual field. However, instead of being based on contrast, it measures similarity with the target hue color H^g quantitatively [56]. One simple measure is to use the function β_4 defined as:

$$\beta_4(t, x) = \frac{1}{1 + (H_{f(t)}(x) - H^g)^2} \quad (2.8)$$

2.3.2. Inhibition

The inhibition mechanism is known to consist of inhibition of return to the previous fixation and short term memory [57, 58]:

- Inhibition of return: The current fixation point f_k is inhibited.
- Short-term memory: The N_m previous fixations $\{f_{k-1}, \dots, f_{k-N_m}\}$ are inhibited.

The second term $\gamma_{k,p}$ encodes both simultaneously:

$$\gamma_{k,p}(t, x) = \gamma_{k_1}^{w_2}(t, x) \quad (2.9)$$

where the function γ_{k_1} is defined as:

$$\gamma_{k_1}(t, x) = \frac{1}{1 + \log \prod_{l=0}^{N_m} \|x - \pi_t(f_{k-l})\|^2} \quad (2.10)$$

Here $\pi_t : \mathcal{F} \rightarrow P_{f(t)}$ is the projection operator that maps each f_k on to the image plane $P_{f(t)}$. Note that $\pi_t(f(t)) = 0$. If the projection of any of the previous fixation points f_i where $i = k - 1, \dots, k - N_m$ is not within $P_{f(t)}$, the corresponding term is simply omitted in the computation of $\gamma_{k,p}$. The parameters w_2 and N_m are also components of p parameter vector. Here, w_2 determines whether inhibition is turned on or not while N_m determines the size of short-term memory.

Consider a sample visual field $v(t_k, \cdot)$ as seen in Figure 2.3a – with the current visual field $v(t_k, \cdot)$ with f_k (black square at the center) and f_{k-1} (black square at the bottom) superposed on it. Let β_p comprise of only TD color term seen in Figure 2.3b. Thus $w_{1_j} = 0, j = 1, 2, 3$, $w_{1_4} = 0.2$ and $N_m = 6$. Without any inhibition – namely when $w_2 = 0$, both the current and previous fixation points are very salient as seen in Figure 2.3c. Here, the color scale is such that increased darkness means greater saliency. Figure 2.3d shows $\varphi_{k,p}$ evaluated on the image plane $P_{f(t_k)}$ with inhibition of return turned on as $w_2 = 0.8$. The previous and current fixation points are inhibited

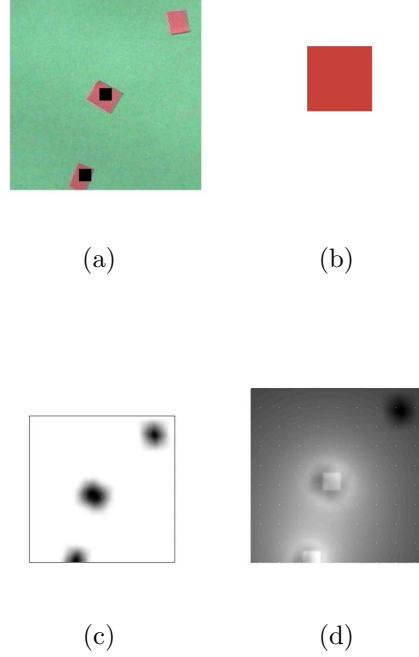


Figure 2.3. The construction of a sample $\varphi_{k,p}$ evaluated on $P_{f(t_k)}$ using TD color as feature saliency - (a) Current visual field $v(t_k, \cdot)$ (b) Target color; (c) $\varphi_{k,p}$ evaluated on $P_{f(t_k)}$ with $w_{1_j} = 0, j = 1, 2, 3, w_{1_4} = 0.2, N_m = 6$ and inhibition turned off - namely $w_2 = 0$; (d) $\varphi_{k,p}$ evaluated on $P_{f(t_k)}$ with the same parameter values except now inhibition is turned on with $w_2 = 0.8$.

(and hence are local maxima) while a darker region not previously attended is a local minimum and hence is potentially the next fixation point.

2.4. Scanpath Algorithm

Several practical issues need to be considered for transforming the APF framework to a scanpath algorithm that can be implemented on a robot. First, while in theory, as the camera moves in the interval $[t_k, t_{k+1}]$, its visual field from $v(f_k, \cdot)$ to $v(f_{k+1}, \cdot)$ is continuous, in practice, it is piecewise constant as it can be updated only N_k times at intervals of δt with $t_{k+1} = t_k + N_k \delta t$ as shown in Figure 2.4. Consequently, the function $\varphi_{k,p}$ is piecewise constant for each interval $[t_k + n\delta t, t_k + (n+1)\delta t]$ where $1 \leq n \leq N_k$. Note that the value of N_k is not necessarily constant and may change between consecutive fixations.

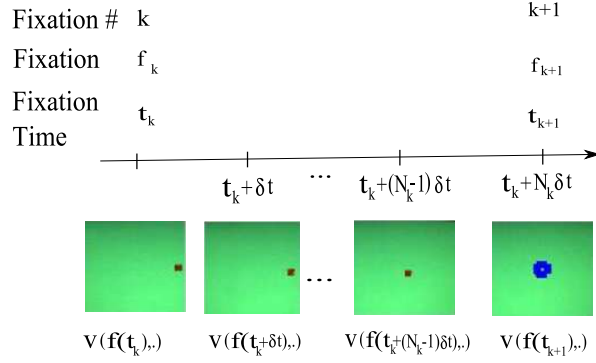


Figure 2.4. Time evolution of visual fields from k th fixation to the next. The robot starts at fixation point f_k and through a series of N_k movements, ends up at the next fixation point f_{k+1} . The intermediate visual fields $v(f(t_k + n\delta t), \cdot)$ where $1 \leq n \leq N_k$ are as shown.

Secondly, the dynamical system as defined by Equation 2.1 reaches an equilibrium point when $u_k(t, f(t)) = 0$ for some $t > t_k$. As is well known, numerical computation of gradient is susceptible to noise since it is essentially a high-pass filter applied on the image. In order to make this computation numerically more robust, the computation of gradient is done via averaging the gradient vectors over the fovea $A_{f(t)}$. Similarly, the control input is computed by taking the average of gradient vectors on the whole of $P_{f(t)}$. In addition to also having increased robustness, this also helps perturbing the system away from the current fixation f_k which is a maximal point of $\varphi_{k,p}(t_k, \cdot)$ which is the case when inhibition of return is active. As the associated control input $u_k(t_k, 0) = 0$, the camera system cannot start moving away from f_k . If the computed vector turns out to be zero, previous control input is applied. If the robot's previous velocity is also zero, the robot moves its camera with a random velocity input. Hence, the resulting algorithm for the movement dynamics for $t \in [t_k, t_{k+1}]$ is mathematically

expressed as:

$$\dot{f}(t) = \begin{cases} 0 & \text{if } t > t_k, \sum_{x \in A_{f(t)}} u_k(t, x) = 0 \\ - \sum_{x \in P_{f(t)}} u_k(t, x) & \text{if } \sum_{x \in P_{f(t)}} u_k(t, x) \neq 0 \\ \dot{f}(t - \delta t) & \text{if } \sum_{x \in P_{f(t)}} u_k(t, x) = 0 \\ \text{random} & \text{if } t = t_k, \dot{f}(t - \delta t) = 0 \end{cases} \quad (2.11)$$

Note that the control inputs are normalized before being applied to the camera motors. An index function $i : R^{\geq 0} \rightarrow Z^{\geq 0}$ defines when each movement ends and hence the fixation point is switched to the next point given as:

$$i(t) = \begin{cases} k + 1 & \text{if } \sum_{x \in A_{f(t)}} u_k(t, x) = 0 \\ k & \text{otherwise} \end{cases} \quad (2.12)$$

Thus, $\{f_{i(t)}\}$ yields the sequence of fixation points associated with the robot's scanpath.

Finally, a short-term memory of $\mathcal{M}_s$ with max size N_m is constructed. It is initialized to be empty. $\mathcal{M}_s$ operates as a first-in, first-out type of queue with updates at every fixation t_k with $k = 0, \dots, N_S$ as follows:

$$\mathcal{M}_s = \begin{cases} \mathcal{M}_s \cup \{f_k\} & \text{if } |\mathcal{M}_s| < N_m \\ \mathcal{M}_s \cup \{f_k\} \setminus \{f_{k-N_m}\} & \text{otherwise} \end{cases} \quad (2.13)$$

The movements continue until $i(t) = N_S$. Note that if $N_S = \infty$, the robot will look around continually. An example of a single movement with the corresponding visual fields is shown in Figure 2.5a. Here, the salient point is to the right of the initial

visual field. The system starts moving and at each step the visual field is updated. The resulting $f(t)$ for $t \in [t_0, t_1]$ projected onto the initial visual field P_{f_0} is as shown in Figure 2.5b.

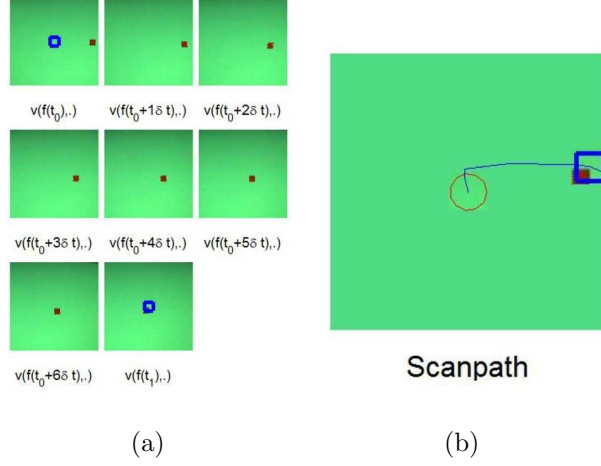


Figure 2.5. (a) Sampled visual field map values $v(f(t), \cdot)$ for $t \in \{t_0, t_0 + \delta t, \dots, t_0 + N_0 \delta t\}$ in a movement from f_0 to f_1 ; (b) The resulting trajectory $f(t)$ for $t \in [t_0, t_1]$ projected onto P_{f_0} where f_0 is at the center of the red circle and f_1 is at the center of blue square.

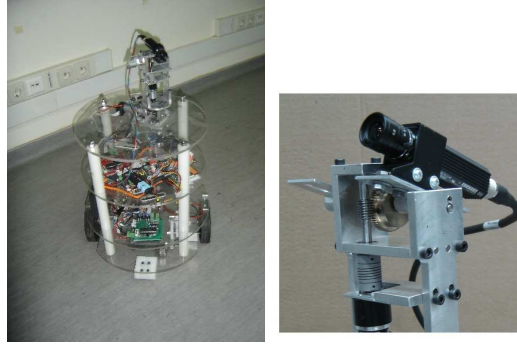


Figure 2.6. APES Robot. Complete robot and the camera head.

2.5. Experimental Results

A series of experiments have been conducted using APES seen in Figure 2.6. APES has a visual field size $d_v = 300$, fovea size $d_f = 30$. The configuration space $\mathcal{F}$

Table 2.1. APF parameter values for different visual behaviors. The terms in the top row are the weights associated with the following features: w_{1_1} -BU Corner, w_{1_2} -BU Color, w_{1_3} -BU Motion, w_{1_4} -TD Color, w_2 -Inhibition, N_m -Memory Size.

Visual Behaviors	p	w_{1_1}	w_{1_2}	w_{1_3}	w_{1_4}	w_2	N_m
Free Viewing	p_1	2	2	0	0	1	6
Detecting visual onsets	p_2	1	1	1	0	1	0
Search	p_3	1	0	0	1	1	6

is defined as:

$$\mathcal{F} = [-35^\circ, 35^\circ] \times [-35^\circ, 35^\circ]$$

The accuracy is $\pm 2^\circ$ in both the pan and tilt angles. The visual field map update period $\delta t = 0.5$ second. The control input as given by Equation 2.2 is given in the image plane coordinates. However, from geometry, it is easy to see that these angular velocity components are directly proportional to the Cartesian velocity components [12]. Hence, the control input is subjected to a linear transformation that normalizes its magnitude with respect to maximum velocity values in its two degrees of freedom. The normalization is based on maximum control input which can be computed based on extremum visual field conditions. The technical details of the APES robot can be found in Appendix A and the procedure of preparing the robot for the experiments can be found in Appendix B.

In the experiments, the robot is engaged in one particular visual behavior by setting the parameter values as given in Table 2.1.

Whenever possible, fixations are compared with N_I salient regions labeled manually by human observers using two different measures. For each image, two subjects

select the interesting areas independently and those that are selected by both are labeled as salient regions. The first measure is hit rate (HR) – namely the similarity of robot’s fixations with those that are manually determined. HR can be mathematically defined as $HR = \frac{N_A}{N_S}$ where N_A denotes the number of fixations that fall within the manually labeled regions. Note that $HR \rightarrow 1$ as more of the robot’s fixations are within the manually labeled salient regions. Let it be noted that the false positive rate is given by $\frac{N_S - N_A}{N_S}$. The second measure is miss rate (MR) – namely the percentage of the manually labeled salient regions that are missed out by the robot’s scanpath. Letting $N_U \leq N_I$ denote the number of regions missed out, $MR = \frac{N_U}{N_I}$. As $MR \rightarrow 0$, the robot is more ensured to have not missed fixating on a salient region. It should be noted that in general $HR + MR \neq 1$ - namely their sum is not necessarily unity.

Secondly, whenever applicable, two measures for evaluating responsivity performance have been used. The first is the number of fixations $N_R \leq N_S$ made until fixating on the object of interest (a visual onset or a detected target). The smaller N_R is, the faster is the detection capability of the robot. A second measure is the average duration of the camera movements when homing on the object of interest as measured by average catch duration $CD = \frac{1}{N_S} \sum_{i=1}^{N_S} N_i$. Again, the smaller is this value, the quicker is the robot in fixating on the object of interest and CD can be viewed as a measure of agility.

2.5.1. Free Viewing Experiments

In the free viewing experiments, both the robot’s body and the scene are set to be stationary. The robot engages in free viewing behavior on a set of scenes shown to it on a monitor. The robot makes $N_S = 15$ saccades. As both the bottom-up corner and bottom-up color weights are active, the robot is expected to look at regions containing corners or of color different from their surroundings.

For a start, a benchmark experiment for testing the accuracy of the robot’s fixation behavior is done. The robot is shown an image on the monitor as shown in Figure 2.7a and tested to see how many of the red squares it will look at. The robot’s

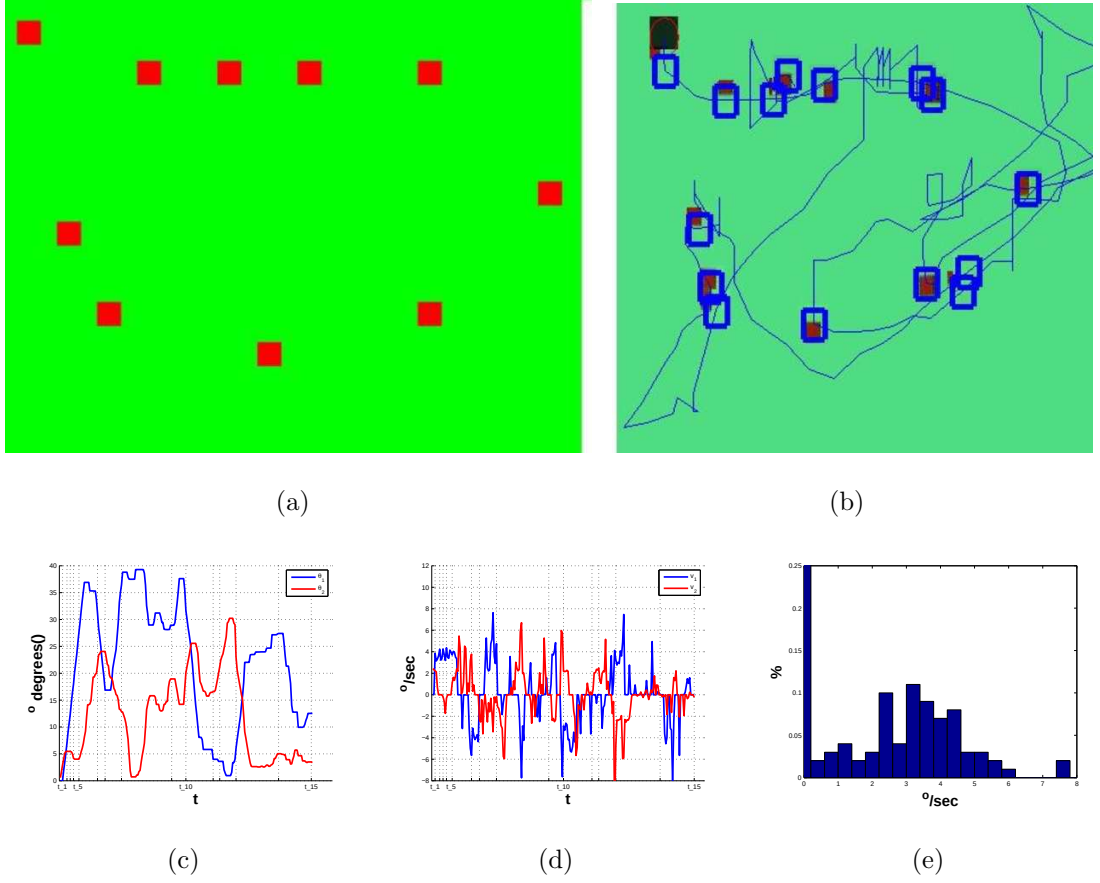


Figure 2.7. Free viewing. (a) Scene shown to APES; (b) The corresponding scanpath trajectory projected onto P_{f_0} where f_0 is at the center. (c) Time evolution of f (pan and tilt angles); (d) Time evolution of velocity $\dot{f}$; In both graphs, the ticks on the x-axis represent t_k . (e) Histogram of ∞ -norm of $\dot{f}$.

scanpath seen in Figure 2.7b includes all the red squares. Some squares are fixated upon more than once. It is observed that the mechanical $\pm 2^\circ$ accuracy of the robot's saccade mechanism causes few of the fixations to be slightly off. The results of Figure 2.7 reveal big variations in the pan-tilt angles with peak velocity magnitudes reaching nearly $8^\circ/sec$. In the histogram of ∞ -norm of the velocity vectors, it is observed that $\|\dot{f}\|_\infty \geq 3^\circ/sec$ for around 46% of the total scanpath duration. Hence, camera movements have relatively high speeds. These results suggest that the motion associated with free viewing is of saccadic nature as expected.

Next, the robot is shown images of various indoor and outdoor scenes as seen



(a) Scenes 1,2,3



(b) Scenes 4,5,6



(c) Scenes 7,8,9



(d) Scenes 10,11,12

Figure 2.8. Scenes shown to the robot in the free viewing experiments.

Table 2.2. Free viewing results with $N_S = 15$ fixations.

Scene #	N_I	N_A	N_U	HR	MR
1	6	15	1	1.00	0.17
2	4	12	1	0.80	0.25
3	6	13	1	0.87	0.17
4	4	11	1	0.73	0.25
5	7	14	1	0.93	0.14
6	11	15	3	1.00	0.27
7	4	14	0	0.93	0.00
8	4	13	1	0.87	0.25
9	9	14	3	0.93	0.33
10	9	14	2	0.93	0.22
11	8	15	2	1.00	0.25
12	10	15	2	1.00	0.20

in Figure 2.8 in order to evaluate performance under very different scene conditions. Table 2.2 shows results for each of the twelve scenes. The average HR across all the twelve scenes is calculated to be 0.92 with variance 0.0074 while the average MR is 0.21 with variance 0.0071. Thus, for still scenes, via free-viewing using only bottom-up corner and color saliency with inhibition turned on, the robot is able to fixate on most of the manually determined salient regions.

We also investigated the effect of the number of saccades N_S and the initial fixation f_0 on HR and MR . In Figure 2.9a, as the number of fixations N_S increases, HR first increases and then starts to decrease. This is because the robot starts looking at points not manually labeled to be salient. On the other hand, MR decreases around 30% as seen in Figure 2.9a. Thus, HR is not affected significantly by increasing N_S while MR can have a significant improvement as the number of fixations N_S is increased. In Figure 2.9b, the effects of different starting points are shown. It should be observed that salient regions occur towards the center in these scenes. As seen, when

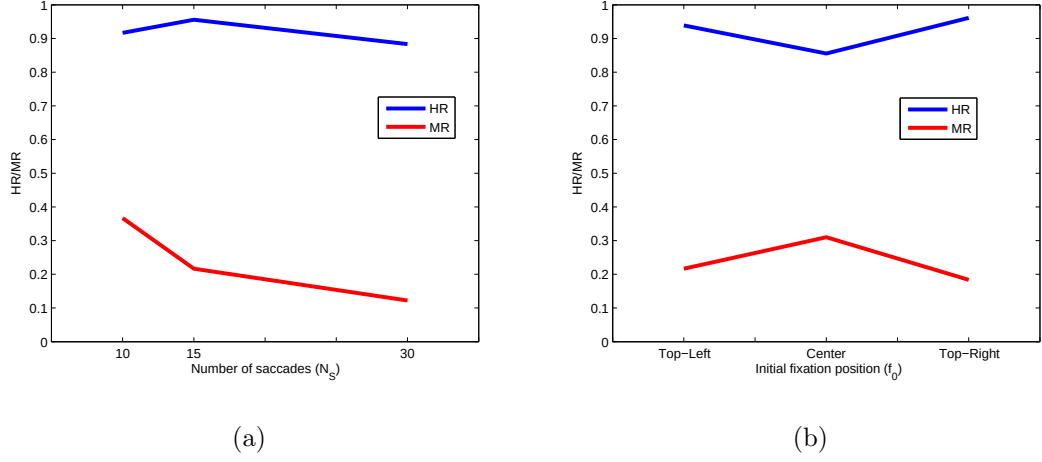


Figure 2.9. HR and MR in free viewing: (a) Variation with respect to the number of fixations N_S ; (b) Variation with respect to different starting points f_0 .

the robot starts free viewing from the salient extremity, both HR and MR performance are comparatively better than starting from the center. This is because the scanpath is likely to cover a greater region with the same or even smaller number of fixations as it does not have to pass through the center several times. However, in both, the variation is less than 10% which means that sensitivity to variations in the initial fixation point is minute.

Finally, feedback-based free viewing behavior is compared in stationary scenes with an open-loop approach using Saliency Toolbox - a commonly used tool in computer vision [59] with accessible codes. Similar saliency features (corner and color) are selected with $N_S = 15$. It should be noted that the operational conditions are very different since while APF approach results are obtained from a robot with a moving camera, Saliency Toolbox results are obtained via computer simulations. Sample scanpaths for one of these images are as shown in Figure 2.10. As seen in Table 2.3, HR is comparatively better for APF whereas the two approaches have similar MR.



Figure 2.10. Comparative fixations under free viewing - (a) Scene; (b) APF ; (c) Saliency Toolbox.

Table 2.3. Comparative results in free viewing – Saliency Toolbox (ST) vs APF with $N_S = 15$.

Scene #	1		12	
N_I	6		10	
Approach	ST	APF	ST	APF
N_A	9	15	14	15
N_U	1	1	2	2
HR	0.60	1	0.93	1
MR (%)	0.17	0.17	0.20	0.20

2.5.2. Detecting Visual Onsets

In the next set of experiments, while the robot’s body continues to be stationary, the scene is made dynamic with visual onsets introduced to the scene while the robot is engaged in a saccade. The APF parameters are set so that BU motion (with $\tau = 40$) and TD color saliency features are selected. The translation amount T_m is computed via geometric transformation from spherical to Cartesian coordinate system based on the amount of movement data obtained from the encoders of the camera pan-tilt motors. The robot is able to respond and change its saccade trajectory towards the newly

appearing object. This is in contrast to a robot engaged in an open-loop saccade that would totally ignore such a change until the end of the saccade.

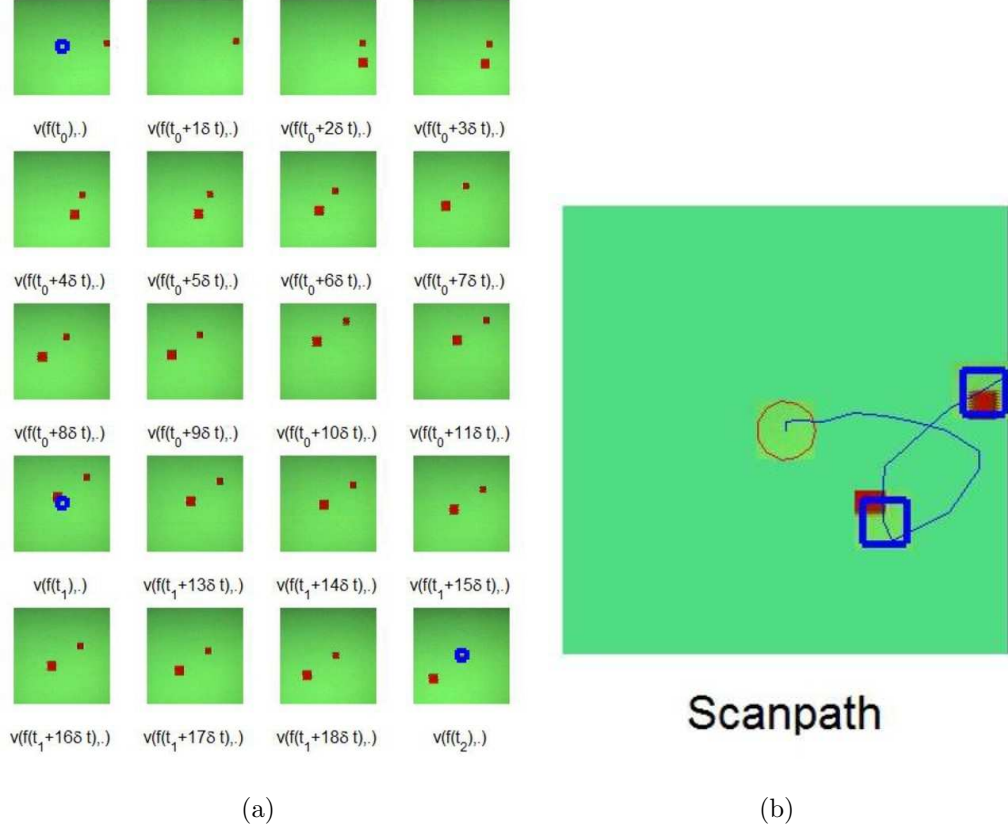


Figure 2.11. Detecting a red visual onset on a monitor: (a) Sampled visual field map values $v(f(t), \cdot)$ for $t \in \{t_0, t_0 + \delta t, \dots, t_0 + N_0 \delta t\}$ during the first saccade and for $t \in \{t_1, t_1 + \delta t, \dots, t_1 + N_1 \delta t\}$ during the second saccade where $t_1 = t_0 + N_0 \delta t$. (b) The resulting scanpath trajectory $f(t)$ for $t \in [t_0, t_2]$ projected onto P_{f_0} . The robot starts saccading towards the red square on the right.

First, the robot is shown a video on the monitor with a red dot in its visual field as seen in Figure 2.11a. As the robot starts its saccade towards that point, a bigger red dot appears on the screen after a brief time. As seen, the robot changes its saccade trajectory and fixates on the visual onset. The second saccade starts from this point and ends at the first red square. The visual field map is updated 12 times during this movement. In the next saccade, the robot fixates on the original target. The resulting scanpath trajectory is as seen in Figure 2.11b.

Table 2.4. Results in detecting visual onsets while a saccade is in progress. The variation of N_R and catch duration CD with respect to onset proximity to the origin – namely the starting fixation point.

Onset Distance (px)	Exp. Setup	N_R	CD (# Frames)
76.30	Screen	1	7
100.66	Screen	1	9
103.25	Lab	1	20
117.05	Lab	1	20
152.32	Screen	1	20
196.47	Screen	1	23

Following, the robot engages in the same visual behavior in a real environment as shown in Figure 2.12. Just as the robot starts saccading towards the red cup but before the saccade is completed, another red object appears. The saccade trajectory changes automatically and ends on this newly appearing red object.

A set of experiments is conducted via varying the “nearness” of the visual onset. This is measured by its distance in pixels to the origin – namely the initial fixation point. The results as presented in Table 2.4 show that $N_R = 1$ regardless of distance – that is the robot is able to change its original saccade trajectory and fixate on the visual onset in the first saccade. However, as the distance of the visual onset increases, so does the catch duration CD .

2.5.3. Search Experiments

In the search experiments, both the robot’s body and the scene continue to be stationary. The robot engages in search for a preset feature on a set of four different scenes shown to the robot as seen in Figure 2.13. Different features are selected for each different scene as explained in the same figure. A sample search as seen in Figure 2.14.

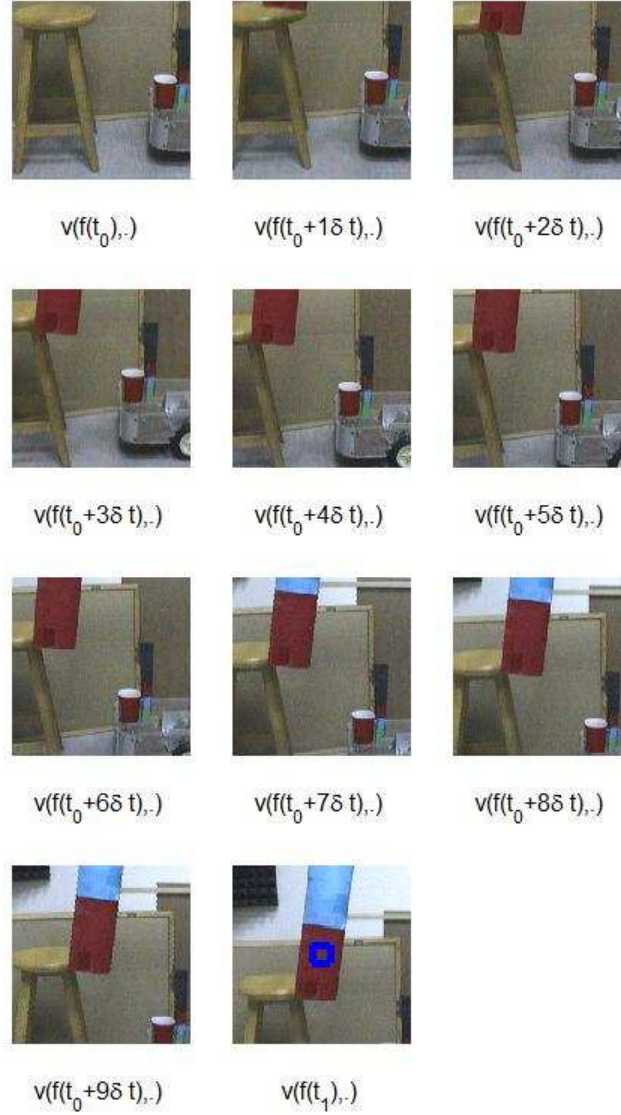


Figure 2.12. Detecting a red visual onset in the lab. Sampled visual field map values $v(f(t), \cdot)$ for $t \in \{t_0, t_0 + \delta t, \dots, t_0 + N_0 \delta t\}$ in a movement from f_0 to f_1 .

A set of experiments is conducted via varying the proximity of the searched feature to the robot's starting fixation. Performance is again based on two measures – the number of fixations N_R made until finding the searched feature and the catch duration CD . As seen in Table 2.5, in all the experiments, the robot finds the region of interest in one saccade. CD increases as the distance of the searched feature increases. As there are two yellow regions in the third scene, as they both attract attention, the saccade duration increases more than expected.

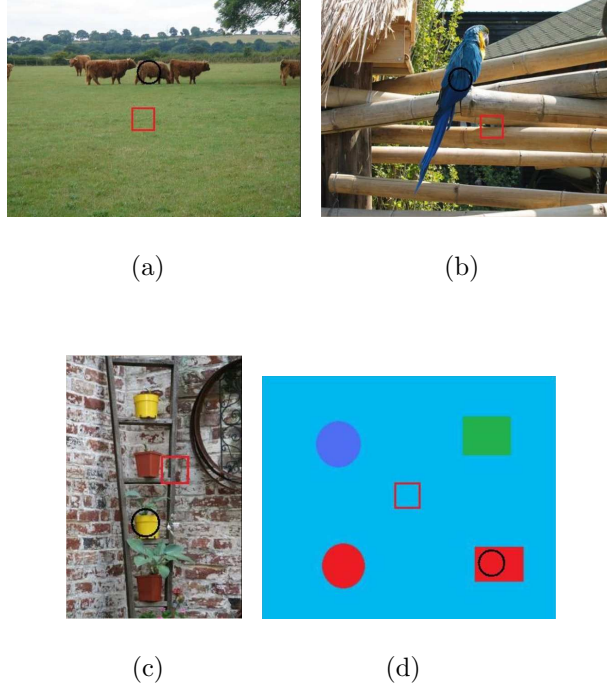


Figure 2.13. Search experiment scenes $v(f(0), \cdot)$ with the searched feature varying follows. (a) Image 1 (Red and corner); (b) Image 2 (Blue); (c) Image 3 (Yellow); (d) Image 4 (Brown). The circle indicates f_0 whereas the square indicates f_1 – both projected onto P_{f_0} .

2.5.4. Summary

The experiments serve to demonstrate that a robot employing the proposed approach can exhibit different visual behaviors in real-time with a set of different features or any combination of. In the free-viewing experiments, the hit rate HR values are above 0.87 for most of the scenes which indicate that most of the regions viewed by the robot were also marked as being salient by a human subject. Miss rate MR values are between 0.17 – 0.33 which indicate that most of the regions marked by a human subject are also covered by the robot’s scanpath. Comparative results with Saliency Toolbox [59] reveal very similar performance measures for one mode of visual behavior – namely free viewing.

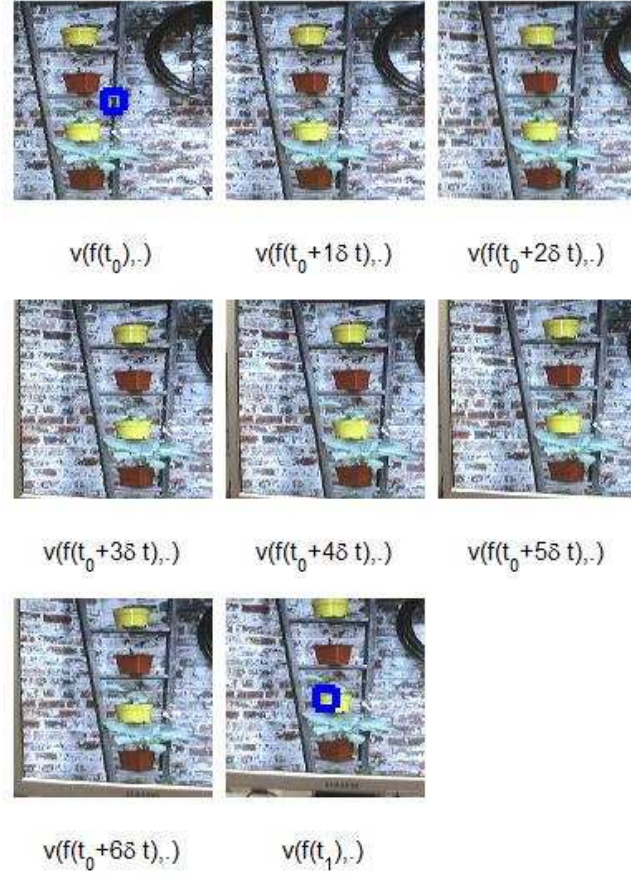


Figure 2.14. Search for yellow: Sampled visual field map values $v(f(t), \cdot)$ for $t \in \{t_0, t_0 + \delta t, \dots, t_0 + N_0 \delta t\}$ in a movement from f_0 to f_1 .

2.6. Conclusion

This chapter describes a novel approach to scanpath generation based on a family of artificial potential functions. Each movement from the current fixation point to the next is associated with an artificial potential function and is achieved simply by “sliding” into the equilibrium point of the associated negative gradient field. The construction of the artificial potential functions depends on a multitude of bottom-up and top-down saliency features and possibly inhibition depending on the visual behavior at hand. The contribution of this approach is that a variety of different features including corners, color and motion can be selected in a pre-attentive stage in mobile robots. The closed-loop nature allows for increased visual responsiveness

Table 2.5. Results of search experiments: Variation of N_R and CD with respect to distance.

Dist (px)	Scene #	N_R	CD (# Frames)
35	1	1	3
44	2	1	4
64	3	1	8
100	4	1	6

since in contrast to open-loop approaches, detecting changes in the visual field during a movement become possible which enables the robot to compute the next fixation in a faster manner. Experimental results on APES robot serve to demonstrate the resulting performance of the robot engaged in different visual features.

3. BUBBLE SPACE AND PLACE REPRESENTATION IN TOPOLOGICAL MAPS

Place representation is crucial to map building and spatial reasoning with mobile robots [1]. There is a growing interest in augmenting or possibly replacing purely metric maps with topological representations [5]. The focus of this chapter is on the problem of visual place representation in topological maps. The images are acquired either via a two degrees of freedom camera head or an omnidirectional camera mounted on a mobile robot. This problem is of great interest to the machine vision and robotics communities [8]. The proposed approaches either represent each place directly based on visual data or first find a set of image features which are then used to generate the final representation. The second approach is taken and a set of feature observations along with their associated viewing directions are used in the representation.

This chapter presents bubble space (BuS) - a novel representation for “places” (nodes) in topological maps. The starting point is bubble memory where visual feature values and their local S^2 -metric relations are simultaneously encoded from an egocentric perspective [13, 14]. The novelties of bubble space representation are two-fold:

- First, in bubble space, bubble surfaces extend bubble memory to accommodate two new variables – varying robot pose and multiple features. Bubble surfaces enable the representation of all visual features in a manner that is implicitly dependent on robot pose while preserving their local S^2 -geometry.
- Second, bubble surfaces are transformed into bubble descriptors. Bubble descriptors are holistic representations that are rotationally invariant with respect to heading changes while being computable in an incremental manner.

Hence, bubble space simultaneously provides for detailed (bubble surfaces) and holistic (bubble descriptors) representations of places. Bubble descriptors are used in place learning and recognition with support vector machines (SVM) - which is a stan-

dard supervised learning approach. During learning, as the robot collects visual data at various robot poses and viewing directions from each different place, it simultaneously constructs the set of bubble surfaces and then the associated bubble descriptors in an incremental manner. These are then encoded in its memory using multi-class SVM. For recognition, the robot is given visual input of one of these places with different viewing conditions such as its pose, illumination or a combination of both. Its decision making is based on comparing the currently constructed bubble descriptors with those of previously learned places in the hierarchical SVM framework. The results show that bubble space based representation can achieve compatible recognition rates with a lower standard deviation and significantly lower codebook extraction time.

The rest of this chapter is organized as follows. First, a review of the related literature is given in Section 3.1. Section 3.2 describes the mathematical framework of bubble space. The transformation of bubble surfaces into bubble descriptors is presented in Section 3.2.2. Then learning and recognizing places using bubble descriptors with multi-class SVM is explained in Section 3.3. Experimental results with the benchmark COLD data set [16] and New College data set [60] are presented in Section 3.4 where detailed analysis on recognition rate is provided and the approach is compared with the state-of-the-art descriptors. The chapter concludes with a brief summary.

3.1. Related Work

There is a general consensus that space representation is based on maps that encode knowledge pertaining to both local surroundings and that extending beyond the current sensory horizon. Map models can be categorized into two major groups: metric and topological. Metric models such as two-dimensional occupancy grids [61, 62], feature-based maps [63, 64] or submap based approaches [65] enable explicit modeling of free-space and thus ease of fusing data from different sensors or locations. However they have problems in large unstructured environments due to the fixed grid-size, accompanying computational burden due to potentially thousands of states in large environments and loop closure without any knowledge regarding the robot's initial position or large pose errors [3]. Alternatively, topological maps are allothetic char-

acterizations of places [5] where space is usually described as a collection (graph) of places (nodes) linked by adjacency relationships (edges). However, there is little consensus about what topological maps are or the way they are built [5]. The meanings of nodes and edges in a topological map vary depending on the algorithms as well as the application.

A key element in topological maps is the place representation and recognition module [6]. The criticality lies in the efficient representation of the sensory input so that it can be used reliably in the matching process associated with recognition of previously visited and learned places. One approach to place representation is to use metric maps for small-scale space and a topological representation for building maps of large-scale space such as the ATLAS framework [66]. However, the partitioning of the environment into an hybrid representation is challenging while local metric maps suffer from problems of metric maps. Topological representations can be categorized into object-based and appearance based approaches. In object-based approaches, visual features are defined based on the occurrence statistics of different objects in different places [67].

However, these approaches often do not generalize to new environments – mainly due to the fact that object detection is still an unsolved problem with respect to handling the intra-class variations in the appearance of objects [7]. The alternative approach is based on global configurations in the observed scenes without explicitly detecting and recognizing objects – also known as appearance based approaches. These methods are classified into two general categories: context based and landmark based [7].

Context based approaches encode the incoming images directly. Amongst context-based methods are using relatively large 11×11 pixel image patches [68], decomposition into eigenimages [69] or computing spatial envelope properties after applying various transforms such as Discrete Fourier Transform [70] or wavelet image decomposition [71]. Another approach is to construct descriptors such as color histograms [6], the coloured pattern appearance model (CPAM) encoding chromatic and achromatic spatial pat-

terns [72] or the histogram of oriented uniform patterns via partitioning the image into sub-blocks and then selecting the most informative image sub-blocks [7].

Alternatively, landmark-based approaches first extract a set of image features which is then used in generating the final representation [9, 73]. The most commonly used features are Harris-Laplace features [74], SIFT descriptors [75], SURF descriptors [76] and fingerprints [77]. While local features represent a limited area of the image, they usually provide robustness against common image variations. However, matching is sensitive to local variations in the incoming images. In order to overcome this problem, these features are further transformed into other descriptors. Most methods are based on partitioning the image into increasingly fine sub-blocks, generating a vocabulary of visual words from local image features and then building a bag-of-features representation [78]. The incoming features are then matched against this vocabulary resulting in a response vector that encodes the frequency of each visual word in the image. While having large number of sub-blocks and visual words lead to good performance in image representation, their major drawback is high dimensionality [79, 80]. The performance of bag-of-features is improved via a model that builds an intermediate representation based on different approaches such as probabilistic Latent Semantic Analysis (pLSA) algorithm [80] or spatial pyramid [79]. Hierarchical organization schemes such as tree-structured Bayesian network and Chow Liu algorithm have been used for expediting the matching process [81, 82]. Still, there are several challenges associated with learning visual vocabularies at large scale such as the exorbitant number of features and criticality of properly designing words [82].

Therefore, a higher performance can be observed when localization information is used with place recognition. Furthermore, only appearances of features are used and their spatial coordinates are discarded [83].

As the information content of a representation is enhanced by integrating spatial knowledge, numerous works have addressed this issue. The sensory egosphere projects images on to a spherical surface placed around the robot [84]. This idea is used as a database structure where each point on the sphere refers to a data structure rep-

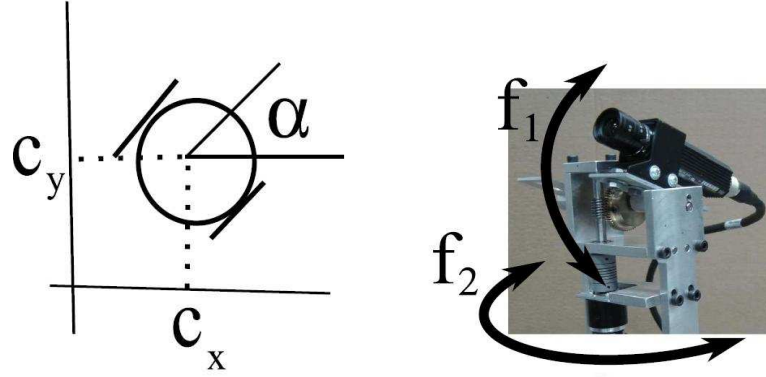


Figure 3.1. Robot's base point in $x \in \mathcal{X}$ and robot's viewing directions

$$f = [f_1 \ f_2]^T \in \mathcal{F}.$$

representing the sensory inputs from that direction [85] or for representing visual and depth data directly [86]. Egocentric representations have also been used via projection onto spheres which are then compactly represented via spherical harmonics [87–89]. Semantic descriptors that are based on the convex hull of image features encode spatial relations between neighboring features [8]. A different approach – called a bubble memory – simultaneously encodes visual feature values and their local S^2 –metric relation from robot's viewpoint [13, 14]. This chapter presents a new approach to place representation based on bubble space. In bubble space, bubble surfaces extend bubble memory to varying robot pose and multiple features. Furthermore, bubble descriptors which are obtained from bubble surfaces enable holistic representation of each place in a rotationally invariant manner. Both bubble surfaces and bubble descriptors can be computed incrementally with each new visual observation.

3.2. Bubble Space

Consider a robot positioned at location $c \in R^2$ in a two-dimensional workspace with a heading $\alpha \in S^1$ as shown in Figure 3.1. Define its base as $x = \begin{bmatrix} c^T & \alpha \end{bmatrix}^T$. The base space² is defined to be the set of all possible robot poses $\mathcal{X} = R^2 \times S^1$. As the

²The base space can be generalized to a three-dimensional workspace. In this case, $\mathcal{X} = R^3 \times S^3$. However, in this chapter, we assume a two-dimensional workspace.

robot moves in $\mathcal{X}$, any of its two-dimensional location or heading may change. At each base point $x \in \mathcal{X}$, the robot is assumed to have the ability to direct its optical axis to any direction within its physical limits. Let the robot camera state space (pan and tilt movements) be denoted by $\mathcal{F} \subset S^2$. It is assumed that $\mathcal{F}$ is a finite set as determined by the resolution of the camera head. Assume without loss of generality that the axis of rotation of pan angle is parallel to the axis of heading rotation. The bubble space $\mathcal{B} = \mathcal{X} \times \mathcal{F}$ is an abstract representation of the robot's base along with its viewing directions. Each bubble space point $b \in \mathcal{B}$ is defined as $b = [x^T f^T]^T$ where $x \in \mathcal{X}$ and $f \in \mathcal{F}$. As the robot moves in $\mathcal{B}$, any of its two-dimensional location, heading or the optical axis may change. The robot's base point is given by $\pi : \mathcal{B} \rightarrow \mathcal{X}$ - defined as the projection of b onto $\mathcal{X}$ as $\pi(b) = x$. The section is a continuous map $h : \mathcal{X} \rightarrow \mathcal{B}$ such that $\forall x \in \mathcal{X}, \pi(h(x)) = x$. The image of a section h - namely $Im(h(x))$ - is the set of all possible viewing directions from a given base x .

3.2.1. Bubble Surfaces

Assume that at time t , the robot is at base $x \in \mathcal{X}$. In bubble space, what the robot sees depends on its current bubble space point $b \in \mathcal{B}$. For each viewing direction $f \in \mathcal{F}$, it applies a set of visual filters $\mathcal{V}$ with $|\mathcal{V}| = N_v$. The choice of $\mathcal{V}$ will vary depending on the task at hand. Let $q(b, t) = [q_1(b, t), \dots, q_{N_v}(b, t)]^T$ denote the observation vector that consists of all individual responses where visual response value $q_i(b, t) \in [\rho_{min}, \rho_{max}]$ with $0 < \rho_{min} < \rho_{max}$. Now, for each visual filter $i \in \mathcal{V}$, visualizes the robot to be surrounded by an hypothetical spherical surface that is deformed at each f by an amount that is dependent on the visual filter response in that direction. This surface is referred to as bubble surface $B_i(x, t)$. The first argument indicates its dependency on base point while the second argument indicates its dependency on time. As this is done for each visual feature $i \in \mathcal{V}$, a set of N_v bubble surfaces $B_i(x, t)$ is generated.

In general, at any base x , the robot will look only at a subset of viewing directions $Im(h(x))$. Let N_s denote the cardinality of this subset. These viewing directions may correspond to the locations of extracted features or salient points. As the robot looks at

each of these viewing directions over time, for each visual feature i , it makes N_s visual observations which are then encoded by a bubble surface $B_i(x, t)$ after being deformed exactly at the corresponding N_s points. Note that as such, there is a non-uniform sampling of viewing angles. The relative geometry of these viewing angles and hence the associated feature values encode spatial information regarding the features. For example, when the robot is at a particular base as seen in Figure 3.2a with the visual field from a particular viewing direction as shown in Figure 3.2b, the bubble surface associated with one sample feature after it has looked at N_s different viewing directions is as shown in Figure 3.2c. Hence, each set of bubble surfaces enables the integration of spatially distinct features along with their relative geometry as seen from one particular base point. In this perspective, bubble surfaces are detailed representations for each place since they simultaneously encode visual features and their relative S^2 —geometry.

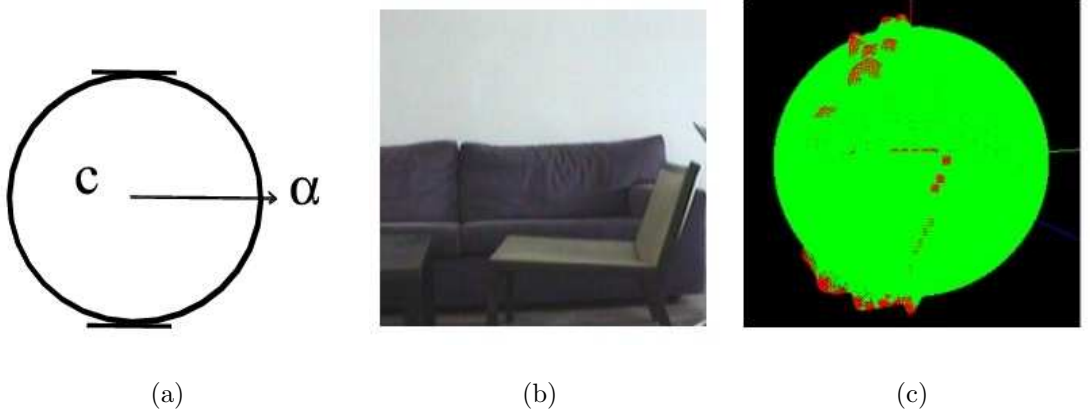


Figure 3.2. (a) Consider a robot at a given base point $x = [c^T \alpha]^T$; (b) The place as seen through one particular viewing direction f ; (c) A bubble surface encoding response to a particular visual filter after $N_s = 100$ different viewing directions.

Mathematically, each bubble surface $B_i(x, t)$ is a deformed sphere embedded in R^3 with an intrinsic parametrization:

$$B_i(x, t) = \left\{ \begin{bmatrix} f \\ \rho_i(b, t) \end{bmatrix} \mid \forall f \in \mathcal{F} \text{ and } b = [x^T f^T]^T \right\} \quad (3.1)$$

where $\rho_i : \mathcal{B} \times R^{\geq 0} \rightarrow [\rho_{min}, \rho_{max}]$ is a Riemannian metric that encodes the response

to the i^{th} visual feature. The parameters³ $0 < \rho_{min} < \rho_{max}$ define the lower and upper bounds on individual feature values. The initial bubble surface $B_i(x, 0)$ is defined as a S^2 sphere with radius $\rho_0 \in [\rho_{min}, \rho_{max}]$ – namely $\rho_i(b, 0) = \rho_0$. As the robot looks around, for each viewing direction, an observation $q(b, t)$ is made, each bubble surface is deformed at the corresponding bubble point. If the robot’s observations include filter responses with identical type ξ , but a range of frequencies [90], only one particular feature from each filter type is selected to generate a set $\mathcal{M}(b, t)$ as:

$$\mathcal{M}(b, t) = \{i \mid \forall j \in \mathcal{V} \text{ s.t. } \xi(i) = \xi(j), q_i(b, t) \geq q_j(b, t)\}$$

The bubble surface is first deformed at b via deforming at b depending whether $i \in \mathcal{M}$ or not.

$$\rho_i(b, t + \delta t) = \begin{cases} q_i(b, t) & \text{if } i \in \mathcal{M}(b, t) \\ \rho_i(b, t) & \text{otherwise} \end{cases}$$

In addition, as visual features tend to dominate in their local neighborhoods, the bubble surface is also deformed in $N_\epsilon(b)$ - ϵ -neighborhood of b , as defined by the standard Euclidean metric - via introducing a local bump at b . As this is equivalent to modifying the metric locally on this deformable surface, the corresponding bubble metric ρ_i is defined as:

$$\rho_i(o, t + \delta t) = \rho_i(o, t) + g_b(o)q_i(b, t + \delta t)$$

where the bump function $g_b : \mathcal{B} \rightarrow [0, 1]$ is any continuous decreasing function that

³Note although in this work, they are taken to be identical for all the visual features, in general, they could be different for each visual feature.

satisfies the following two conditions:

$$g_b(o) = \begin{cases} 1 & \text{if } \|b - o\| = 0 \\ 0 & \text{if } \|b - o\| = \epsilon \end{cases}$$

Interestingly, the bump function $g_b(o)$ alleviates the inaccuracy of the pan-tilt mechanism to some extent via spreading the surface deformation in a neighborhood. Hence, the parameter ϵ is closely related with its accuracy. Again, consider the robot at a particular base point as shown in Figure 3.3a before it starts looking around. In Figure 3.3b, with a viewing direction that is to the left of its heading direction, it encodes its observation via deforming the bubble surface accordingly. With a second viewing direction that is now to the right of the heading direction, a new deformation is added onto the surface as shown in Figure 3.3c. As can be observed, bubble surfaces are not invariant to heading changes as the relative geometry of visual features on the bubble surface with respect to the heading direction will shift in a manner that is dependent on the amount of heading change.

Bubble surfaces are represented compactly based on the coefficients of double Fourier series [91]. Consider a bubble surface $B_i(x, t)$ associated with each feature $i \in \mathcal{V}$ as defined by Equation 3.1. The surface can be explicitly represented by the double Fourier series as:

$$\rho_i(b, t) = \sum_{m=0}^{H_1} \sum_{n=0}^{H_2} \lambda_{mn} z_{xi, mn}^T(t) e_{mn}(f)$$

The parameters λ_{mn} are predefined as given in Appendix D. Define $f \in \mathcal{F}$ as $f = [f_1 \ f_2]^T$. For each (m, n) , the vector $e_{mn}(f) \in R^4$ consists of an orthonormal set

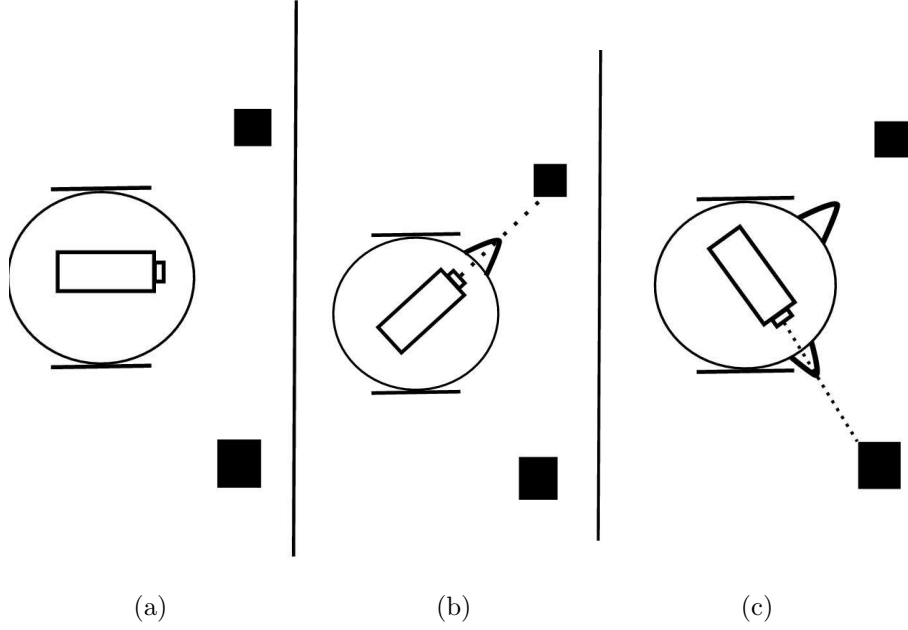


Figure 3.3. The deformation of bubble surface from a particular robot base: (a) Initial undeformed bubble surface; (b) With one viewing direction; (c) After two viewing directions.

of trigonometric basis functions as:

$$e_{mn}(f) = \begin{bmatrix} \cos(mf_1)\cos(nf_2) \\ \sin(mf_1)\cos(nf_2) \\ \cos(mf_1)\sin(nf_2) \\ \sin(mf_1)\sin(nf_2) \end{bmatrix}$$

The corresponding vector $z_{xi,mn}(t) \in R^4$ of coefficients of double Fourier series associated with base point x at time t is defined as:

$$z_{xi,mn}(t) = \begin{bmatrix} \eta_{xi,mn}(t) & \beta_{xi,mn}(t) & \mu_{xi,mn}(t) & \nu_{xi,mn}(t) \end{bmatrix}^T$$

The exact definitions of its components are given in Appendix D. As all the initial bubble surfaces $B_i(x, 0)$ are spherical with radius ρ_0 , all the $z_{xi,mn}(0)$ vectors

can easily be computed as given in Appendix D. Hence, each bubble surface can be reconstructed exactly by the collection of vectors $z_{xi,mn}(t)$. We refer to this collection as bubble surface coefficients. Note that while the series representation is possibly an infinite sum, in practice we can use finite sum where H_1, H_2 are the number of harmonics. Bubble surface deformations for harmonics greater than H_1, H_2 will not be discriminating since they correspond to high frequency spatial deformations and will mostly be due to noise. For the bubble surface as shown in Figure 3.4a, the reconstructed surfaces with $H_1, H_2 = 2$ and $H_1, H_2 = 6$ are as shown in Figure 3.4b and Figure 3.4c respectively.

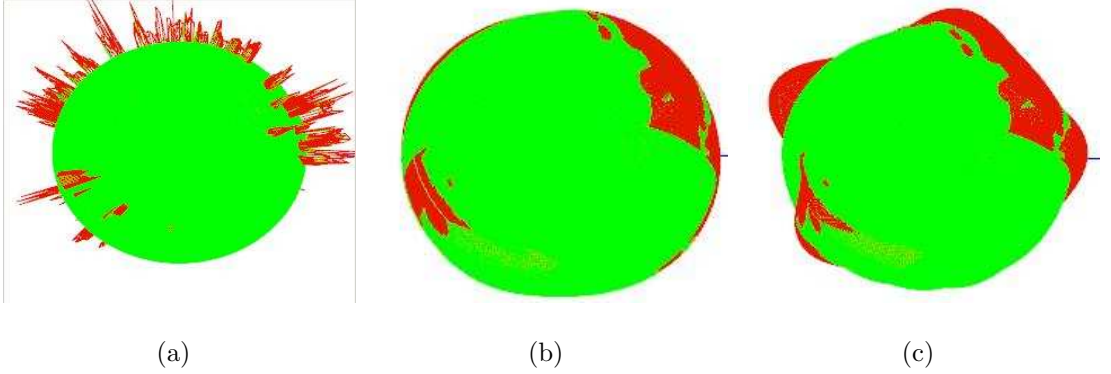


Figure 3.4. (a) Original bubble surface; (b) Reconstructed bubble surface with $H_1 = H_2 = 2$; (c) Reconstructed bubble surface with $H_1 = H_2 = 6$.

As the robot changes its viewing direction to f' at time $t + \delta t$, the bubble surface changes only at this point $b' = [x f']^T$. This in turn implies that bubble surface coefficients can be computed to first order approximation in an incremental manner after each new visual observation. Incremental computation enables the updating of the representation to incorporate each new set of visual feature observations and hence saves computational time. The details are presented in Appendix D.

3.2.2. Bubble Descriptors

Bubble descriptors enable holistic representation of each bubble surface while being rotationally invariant and computable in an incremental manner. Holistic representations enable global representation of each place. Rotational invariance is desired since the robot should be able to recognize a place correctly even if it is observed from different headings. As discussed, incremental computation enables flexibility in integrating new visual feature values.

Bubble descriptors are derived from bubble surface coefficients - namely the coefficients of double Fourier series. In the base space $\mathcal{X}$, rotations are introduced by the changes in the robot's heading - namely $\delta\alpha \in S^1$. While bubble surface coefficients are not rotationally invariant with respect to changes in robot's heading, a series of rotational invariants can be defined as:

$$I_{i,mn}(x, t) = z_{xi,mn}^T(t) z_{xi,mn}(t)$$

For the interested reader, the proof is given in Appendix E. A special case is with $m = 0$. In this case, $\beta_{0n} = \nu_{0n} = 0$ which yields the following invariants:

$$I_{i,0n}(x, t) = \eta_{i,0n}^2(x, t) + \mu_{i,0n}^2(x, t)$$

where $n = 0, \dots, H_2$. Hence, each set of bubble surfaces $B_i(x, t)$ with $i \in \mathcal{V}$ can be compactly represented by a vector $I(x, t) \in R^{N_I}$ where $N_I = N_v(H_1 + 1)(H_2 + 1)$ as:

$$I(x, t) = [I_{1,00}(x, t), \dots, I_{N_v, H_1 H_2}(x, t)]^T$$

This vector is referred to as a bubble descriptor. The initial values of bubble descriptors $I_{i,mn}(x, 0)$ are easily computed based on the initial values of the associated bubble surface coefficients $z_{xi,mn}^T(0)$. Let it be noted that while the use of bubble

descriptors in place representation is new to the best of the author's knowledge, various similar invariants have been used extensively for object retrieval and recognition as discussed in Section 3.1.

Bubble surface coefficients can be computed to first order approximation in an incremental manner with each new observation as explained in Appendix D. This in turn implies that each bubble descriptor can be computed incrementally:

$$\begin{aligned} I_{i,mn}(x, t + \delta t) \cong & I_{i,mn}(x, t) + 2z_{xi,mn}^T(t)\delta z(b', t, \delta t) \\ & + \delta z(b', t, \delta t)^T \delta z(b', t, \delta t) \end{aligned}$$

After each new observation, as the bubble descriptors are updated in an incremental manner, the associated computational complexity depends on the number of the observations directly.

3.3. Learning & Recognizing Places

Suppose that the robot is to learn a set of places $\mathcal{P}$ with cardinality N_p . Furthermore, suppose that the robots looks around in each place $p \in \mathcal{P}$ from M_p different base points. For clarity, the place learning and recognition problem is defined as follows:

- **Learning:** In each place, for each base point, the robot generates a bubble descriptor $I(x_j(p))$. Please note that the second argument will be omitted and $I(x)$ will be used for notational simplicity. Hence, each place is associated with a set of bubble descriptors $\mathcal{I}_p$:

$$\mathcal{I}_p = \{I(x_j(p)) \mid j = 1, \dots, M_p\}$$

Once it visits all N_p places, using the complete set of labeled bubble descriptors $\cup_{p \in \mathcal{P}} \mathcal{I}_p$ as training data, it then deploys a standard learning approach.

- **Recognition:** The robot is given visual data from a random base point in any of these places with possibly changed viewing conditions such as pose and/or illumination. It then tries to recognize this place using its previously learned knowledge.

Thus, place recognition is the problem of assigning a place to one of previously learned places or possibly designating it as a new (previously unknown) place based purely on sensory data. Let us remark here that this problem is related to kidnapped robot and loop-closure detection problems - both of which are concerned with finding positions in already mapped areas for SLAM applications [92]. The kidnapped robot problem is defined as recovering the robot's position in a given map after an arbitrary blind displacement [64]. On the other hand, loop closure is defined to be the problem of detecting when the robot has returned to a previously visited location in its travel trajectory [93,94]. The associated approaches use additional information such as prior information regarding initial position, odometry, motion model or the sequentiality of images. As such, place recognition can be used in kidnapped robot loop closure procedures for coarse positioning in metric or topological SLAM [95].

In learning mode, any standard pattern learning approach may be utilized. In this work, multi-class support vector machine decision trees [96] are used where the bubble descriptors are hierarchically organized in a decision tree in a supervised manner. Although a simpler pattern learning and recognition method could have been used, SVM is preferred since it has become a standard approach that can continue to be used as visual data and the number of places become much more extensive. The advantage of decision trees, having a relatively low computation complexity of $O(\log(N_p))$ at recognition phase, enables real-time decision making for robotic tasks as the number of places gets large. Since SVM is a binary decision process, the decision tree is also binary where each node acts as an SVM [97]. Its construction is as follows:

- The structure of the tree is decided *a priori* in a supervised manner.
- The top node of the tree represents all N_p locations and thus is associated with all the bubble descriptors $\cup_{p \in \mathcal{P}} \mathcal{I}_p$ used in learning.

- For each node n_k , let $\cup_{p \in \mathcal{P}} \mathcal{I}_p(n_k)$ denote the associated set of bubble descriptors
- In all nodes n_k except the leaves, SVM is used to divide the associated bubble descriptors into two child nodes via computing the parameters w_{n_k}, d_{n_k} of a separating hyperplane $w_{n_k}^T I(x(p)) + d_{n_k} = 0$.
- The leaves of the tree represent classified places.

In recognition mode, when the robot enters a place p , as it starts looking around, it constructs a bubble descriptor $I(x(p))$. Note that if $p \in \mathcal{P}$, this is a previously learned place; otherwise it is a new place. The recognition module uses the SVM-based binary tree constructed during learning. At each node n_k of the tree, the current bubble descriptor $I(x(p))$ is associated with a decision value $c(I(x(p)), n_k)$ as follows:

$$c(I(x(p)), n_k) = \sum_{I_i \in \cup_{p \in \mathcal{P}} \mathcal{I}_p(n_k)} \zeta_i y(I_i) K(I(x(p)), I_i) + d_{n_k}$$

The value $y(I_i) \in \{-1, 1\}$ specifies whether a bubble descriptor belongs to left or right child of that node. The parameters ζ_i are obtained via solving dual Lagrangian form of the aforementioned optimal hyperplane problem. The kernel function is a radial basis function defined as:

$$K(I(x(p)), I_i) = e^{-\gamma \|I(x(p)) - I_i\|^2}$$

where $\gamma \in R$ is an a priori defined parameter. The decision value is used to compute the conditional probability of being associated with a given node defined as [98]:

$$pr(y(I(x(p))) = 1 \mid n_k) = \frac{1}{1 + e^{\varsigma_1 c(I(x(p)), n_k) + \varsigma_2}}$$

where ς_1 and ς_2 are computed by minimizing the negative log likelihood of learning data [99]. Letting τ denote a priori selected threshold value, the processing in the

recognition module is as follows:

- Start traversing down the tree starting at the top node.
- At each node n_k ,
 - (i) If $pr(y(I(x(p)))) = 1 \mid n_k \geq \tau$, continue to the children of that node.
 - If $sign(c(I(x(p)), n_k)) = 1$, continue to the right node;
 - Otherwise proceed to the left node.
 - (ii) If $pr(y(I(x(p)))) = 1 \mid n_k < \tau$, then label as being unclassified.
- This is repeated until either a leaf node is reached or unclassified decision is made.

3.4. Experimental Results

This section presents experimental results aimed at evaluating bubble space representation in place learning and recognition. In the learning mode, for each different place, the robot is given visual data from a variety of base points which it then uses to construct the associated bubble descriptors. After it goes through all the different places, it generates the hierarchical SVM tree as described in Section 3.3. In the recognition mode, the robot is given visual data from one of the places it has learned. This visual data differs either in robot pose, viewing illumination, time or any combination there of. Hence, it is not the same as any of the visual data used during learning. Its task is to recognize the current place based on previous learning or designate it as a new (previously not learned) place.

In all experiments, the robot has only visual data as sensory input. It is assumed that it has no access to any motion, relative motion or any other sensing information. In particular, at each base point, it is given a panoramic image and makes $N_v = 56$ visual observations at each different viewing direction. The details of visual filters are presented in Appendix C for completeness. It starts constructing $N_v = 56$ bubble surfaces with surface deformation parameters $\rho_{min} = 0.5$, $\rho_{max} = 1.5$, $\rho_0 = 1$ and $\epsilon = 2^\circ$ respectively. It stops this process after it has gotten visual observations from $N_s = 40$ different viewing directions. These viewing directions are determined automatically via

looking around in a free viewing mode as described in detail in [100]. Simultaneously, the corresponding bubble descriptors are constructed with $H_1 = H_2 = 6$ number of harmonics, however only the first 40 rotational invariants are used since the remaining are observed to be nondiscriminating. Hence, each bubble descriptor is of dimension $56 \times 40 = 2240$. The incremental update of double Fourier coefficients is done with $N_1 = N_2 = 360$. The SVM parameters are obtained via libSVM library [101]. The threshold parameter τ of the SVM is set to a relatively low value ($\tau = 0.3$) which is obtained in a supervised manner via considering all incorrectly recognized places. All the processing is done on a processor with an Intel Xeon W3250 @2.67 GHz, 6 GB RAM.

3.4.1. Indoors Experiments: COLD Data Set

The first set of experiments is done with COLD (COsy Localization Dataset) data set [16] that has been extensively used to evaluate previous approaches [8, 95]. The COLD data set consists of visual data from three different laboratories with different illumination conditions (the cloudy, sunny and night conditions). The first and third laboratories both have five different places: printer area (PA), corridor (CR), two-person office (2PO1), washroom (TL) and stairs area (ST) while the second laboratory has all these places except the stairs (ST) area. For the Saarbrücken Laboratory, the sunny image sequence does not exist. A robot with an omnidirectional camera travels a standard route that passes through each of these places under each illumination condition three different times. The omnidirectional images are converted into panoramic images before being input to the robot. In each place, the robot is given images from $150 < M_p < 350$ base points – depending on the available data. The base increments associated with each frame in each sequence are available in the COLD database [16], however as stated earlier, the robot does not use this information in any way.

The experiments are similar to those that were done in recent and related work [8, 95]. The robot first learns each place at each laboratory in one illumination condition via constructing the corresponding bubble descriptors at a set of base points using one route sequence. Once it completes this, it generates the SVM tree in a supervised

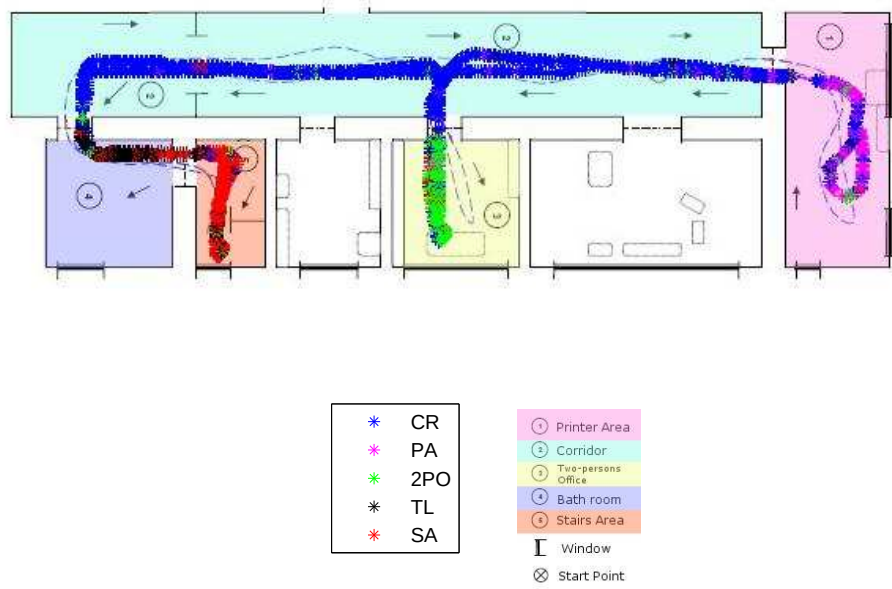


Figure 3.5. A sample recognition task Freiburg Lab. The learning is done in sunny illumination while recognition task is done under cloudy illumination. The overall accuracy for the shown sequence is 82%. Color coded stars on the path indicate what the robot recognized each place as.

manner. In recognition mode, it is input images at a random base point from other sequences which it should classify as one of previously learned places or not seen. As explained, hence visual data is from different bases, with different illumination conditions or both. A sample recognition performance under cloudy viewing condition for a path in Freiburg Lab with learning under sunny viewing illumination is as shown in Figure 3.5.

In order to study the effect of reduced bubble descriptor dimension, the experiments are also repeated with a reduced bubble descriptor (R-BuS). In the reduced set, the visual feature set is reduced to 4 – namely. vertical, horizontal and 30° aligned Cartesian filters and chromatic channel b of CIELAB color space. Furthermore, the number of harmonics is also reduced as $H_1 = H_2 = 2$. Hence the dimension of reduced bubble descriptor is $4 \times 9 = 36$.

Table 3.1. The comparison of four approaches through the corresponding codebook extraction per frame, vector dimension, average total extraction time, SVM model time, total learning time, recognition rate and standard deviation of recognition rates with all routes in COLD data set.

	R-BuS	BuS	e-HCT	CPAM
Codebook extraction per frame (time, ms)	87	268	690	6500
Dimension	36	2240	29	128
Average total extraction time (min)	1.74	5.36	13.80	130
Average SVM model time (sec)	1.10	30.12	0.90	15.60
Total learning time (min)	1.76	5.86	13.82	130.26
Recognition Rate (%)	64.3	70.9	68.9	72.3
Recognition Rate Std. Dev. (%)	12.6	12.7	14.8	20.2

Table 3.1 presents a comparative study with the other state-of-the-art descriptors similar to [8]. Two other representations are chosen - namely e-HCT (extended Hull Census Transform) [8] and CPAM (Colored Pattern Appearance Model) [72] that have been shown to have better performance in comparison to SURF (Speeded-Up Robust Features) [76] used together with HIST (Histogram) [6] or classic BoW (Bag-of-Words) [8, 78]. The codebook extraction time does not include extraction time of features; only feature coding information time is included. The codebook extraction time per frame for BuS representation is around 268 millisecond as compared to 690 millisecond for E-HCT ⁴. This is very short compared to other approaches – even though it is of much higher dimension. This can be attributed to the fact that bubble descriptors can be computed directly in an incremental manner after each visual observation with simple computations as explained in Appendix D. Hence, in turn, the average extraction time per sequence which corresponds to the time taken to extract all the necessary descriptors to model an environment, is also much shorter as seen in the table. Of course, with the R-BuS representation, this is much shorter. As the number

⁴The results as presented in [8] are slightly lower as the computations there were done on a faster processor.

of frames contained in the standard sequences vary between 1000 to 1800 while that in the extended sequences vary between 1400 to 2500, average total extraction time is around 5.36 minutes as compared to 13.8 minutes for the e-HCT descriptor. With the R-BuS representation, this goes down to 1.74 minutes. Average SVM model time is slightly longer as expected since the descriptors have higher dimension. However, since this is of much lower magnitude compared to that of average extraction time per sequence, total learning time is also much lower than that of other methods. At the same time, average recognition rates around 70% are comparable with those of the state-of-art descriptors. We also study the associated standard deviation as a smaller value means lesser susceptibility to changes in the operating conditions and hence will be an indicator of relatively more consistent recognition performance. It is observed that bubble space representation is associated with a standard deviation of 12.7% which is lower than the other two approaches.

The experimental results on recognition rates include a comparative study similar to [8]. The results are given separately for each laboratory as seen in Table 3.2, 3.3 and 3.4. On the regular Freiburg data, the average performance of BuS is better with a lower standard deviation. The worst deviation as measured by the difference of the lowest and average recognition rates is significantly lower as compared to the other approaches. This indicates increased robustness against different viewing conditions. For example, for the Freiburg data set, this value is 21.79% for BuS representation as compared to 30.84% of e-HCT model. When using R-BuS representation, the performance is not as high as expected, however this performance is still better than some of the commonly used approaches. Interestingly, both standard deviation and worst deviation remain to be low. These results suggest that robustness against different viewing conditions continue to hold even in case of reduced set of bubble descriptors.

In order to study scalability, the robot is also made to navigate in an extended route in each laboratory. The extended routes associated with the first two laboratories both contain eight places while that of the third laboratory contains six places. Hence, there are three additional places in the route of the first laboratory, four additional places in that of the second laboratory while that of the third laboratory has one

Table 3.2. Recognition rates for Freiburg Lab (COLD data set) with varying learning and recognition scenarios.

Learning		Test	Freiburg			
			R-BuS	BuS	e-HCT	CPAM
Standard Route	Cloudy	Cloudy	77.8	90.0	77.4	81.7
		Night	63.6	81.3	63.1	58.6
		Sunny	80.1	82.4	67.4	81.0
	Night	Cloudy	63.8	69.3	67.2	25.0
		Night	56.4	77.1	87.2	81.4
		Sunny	45.3	59.8	45.5	18.2
	Sunny	Cloudy	78.5	81.8	75.6	82.2
		Night	53.0	59.8	58.3	53.9
		Sunny	85.3	89.5	92.3	95.4
	Average		67.1	76.7	70.4	64.2
	STD		14.0	11.4	14.5	27.3
	Worst Dev		21.8	16.9	25.0	46.0
Extended Route	Cloudy	Cloudy	51.3	58.9	66.3	72.1
		Night	48.6	55.2	59.7	64.2
		Sunny	51.6	62.9	44.9	53.8
	Night	Cloudy	44.6	44.8	55.6	35.5
		Night	61.2	72.5	80.5	84.0
		Sunny	31.5	33.8	25.9	25.5
	Sunny	Cloudy	60.0	60.5	57.3	62.4
		Night	47.9	48.7	44.9	47.2
		Sunny	58.4	63.2	75.6	84.7
	Average		50.6	55.6	56.8	58.8
	STD		9.2	11.6	16.8	20.4
	Worst Dev		19.1	21.8	30.8	33.3

Table 3.3. Recognition rates for Saarbrücken Lab (COLD data set) with varying learning and recognition scenarios.

Learning		Test	saarbr			
			R-BuS	BuS	e-HCT	CPAM
Standard Route	Cloudy	Cloudy	73.0	84.8	83.0	89.1
		Night	66.7	80.1	71.2	81.5
	Night	Cloudy	63.9	74.8	53.6	85.5
		Night	82.3	84.5	93.2	99.2
	Average		71.5	81.0	75.2	88.8
	STD		8.2	4.7	17.0	7.6
	Worst Dev		10.8	6.3	21.7	10.4
Extended Route	Cloudy	Cloudy	63.7	72.3	64.9	73.4
		Night	47.8	51.6	47.7	58.0
	Night	Cloudy	48.1	51.4	38.4	70.9
		Night	65.4	74.0	87.0	98.5
	Average		56.3	62.3	59.5	75.2
	STD		9.6	12.5	21.4	17.0
	Worst Dev		9.2	11.7	27.5	23.3

Table 3.4. Recognition rates for Ljubljana Lab (COLD data set) with varying learning and recognition scenarios.

Learning		Test	Ljubljana			
			R-BuS	BuS	e-HCT	CPAM
Standard Route	Cloudy	Cloudy	78.8	82.2	82.8	89.1
		Night	68.0	74.5	75.1	74.9
		Sunny	73.1	76.6	71.7	81.4
	Night	Cloudy	53.4	61.5	67.5	75.4
		Night	82.7	85.6	91.3	99.1
		Sunny	50.1	61.5	65.1	68.5
	Sunny	Cloudy	75.6	81.1	78.0	85.7
		Night	70.2	73.1	73.2	70.6
		Sunny	80.1	90.1	87.9	92.6
	Average		70.2	76.2	77.0	81.9
	STD		11.5	9.9	8.9	10.5
	Worst Dev		20.1	14.7	14.3	17.2
Extended Route	Cloudy	Cloudy	74.3	81.3	77.5	85.0
		Night	74.8	76.9	71.6	67.9
		Sunny	67.1	69.3	68.4	74.6
	Night	Cloudy	61.2	65.5	63.4	64.9
		Night	83.4	88.6	89.2	96.7
		Sunny	63.5	67.5	62.5	60.7
	Sunny	Cloudy	67.8	73.1	71.3	77.6
		Night	63.2	71.2	69.7	66.2
		Sunny	71.2	76.6	82.2	87.6
	Average		69.6	74.4	72.9	75.7
	STD		7.1	7.3	8.7	12.1
	Worst Dev		13.8	14.2	16.3	21.0

additional place. The robot is again made to travel in each route three times for the three different illumination conditions - hence again there are 9 different sequences generated. It is observed that as the number of places increase, average recognition rates decrease slightly - similar to all the previous descriptors. However, the standard deviation of BuS representation continues to be the smallest. Hence, while recognition rates are comparable with the other approaches, the recognition performance of BuS representation is relatively less sensitive to changes in the viewing illumination as measured by the standard deviation of recognition rates or the worst deviation attained.

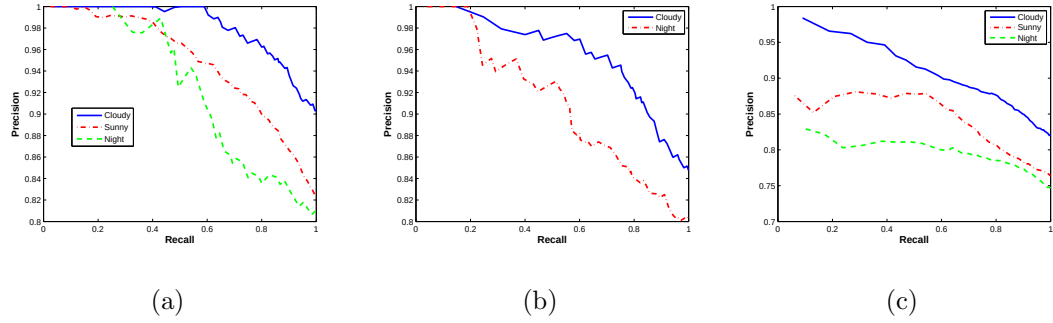


Figure 3.6. Precision recall curves for varying test scenarios with COLDF standard route data set with learning under cloudy illumination. (a) Freiburg Lab, (b) Saarbrücken Lab, (c) Ljubljana Lab.

The trade-off between recall and precision rates at standard route under cloudy illumination is also studied via varying the threshold parameter $\tau \in [0.3, 1.0]$ as shown in Figure 3.6. Again, data sets obtained at different bases, illumination conditions or both are used in recognition testing. The resulting recall rates for specified precision levels are summarized in Table 3.5. It is observed that at 100% precision (with $\tau \in [0.8, 1]$), the maximum achievable recall rate is around 59.2%. However, for 90% precision (with $\tau \in [0.3, 0.85]$), the recall rate increases to around 100%. Unfortunately, to the best of author's knowledge, there is no similar study available in place recognition for comparative purposes. As expected, the performance degrades with testing under different illumination conditions as compared to learning. For example, for Freiburg Lab, at 99% precision, around 60% recall rate goes down to 17% for sunny illumination and 26% for night illumination. For Saarbrücken Lab, again 99% precision, the recall

Table 3.5. Recall rates for specified precision levels with COLD data set as given in Figure 3.6.

			Precision (%)		
Recall (%)	Lab	Test	100	99	90
	Freiburg	Cloudy	59.2	60.4	100
		Sunny	17.2	19.3	80.2
		Night	25.3	30.3	61.7
	Saarbrücken	Cloudy	14.8	25.2	85.8
		Night	19.2	21.5	56.5
	Ljubljana	Cloudy	-	9.1	60.8
		Sunny	-	-	-
		Night	-	-	-

rates are 17% and 20% respectively for cloudy and night illumination testing. It should be noted the maximum attainable precision is lower than 100% for Ljubljana Lab. Although the recognition rates are high, false positives occur due to high similarity of places.

3.4.2. Outdoors Experiments: New College Data Set

The second set of experiments is done using New College data set [60]. This is a benchmark outdoors visual data with a route of approximately two kilometers coverage as shown by the lighter (yellow) lines in Figure 3.7. Let it be noted that while this data has been used in the experimental evaluation of localization tasks [102,103], to the best of author’s knowledge, it has not been used in place learning and recognition tasks. For this, the associated route is divided into seven places whose boundaries are as indicated by the the dark bold lines in the same figure. Each place is then assigned a different semantic label and has visual data from base points varying between $400 \leq M_p \leq 800$. A robot travels down this route through the first three places three different times.

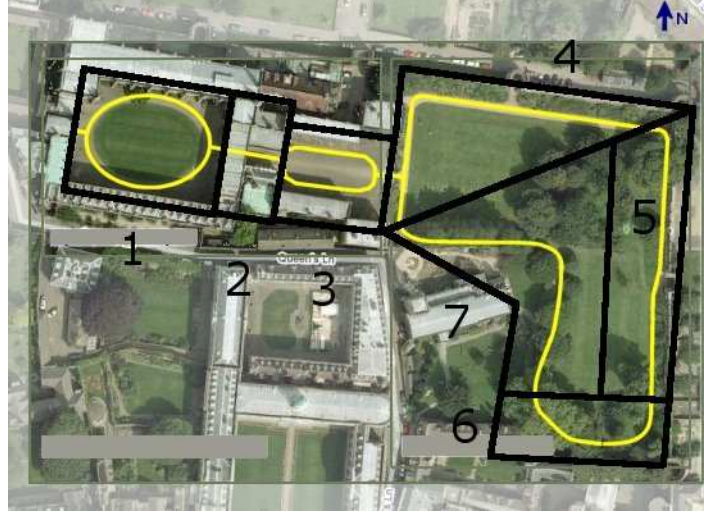


Figure 3.7. A route in New College data set segmented into 7 places.

This is followed by following a route through the next four places which is repeated twice. Both are done in the early hours of the day. These experiments are repeated in the evening once again. Learning is done using data from one complete path while recognition experiments are done using the remaining data sets for early morning data. As only one complete path of evening data is available, this data is sampled for learning while the remaining samples are used in the recognition experiments. Sample visual data used in learning varying in base points and illumination are as shown in the left column of Figure 3.8. The visual data used in recognition vary with respect to base points or illumination or both as shown in the right column of the same figure.

Recognition rates using only day, only evening or both in both learning and recognition phases are given in Table 3.6. Recognition rate is highest when both learning and testing are done with day illumination. In evening test conditions, recognition rate goes down to 59.4%. In case of reversed learning and test illumination conditions, performance goes down by 10% as expected since learning a place in darker illumination will in general contain much less visual data. Finally, when learning is done using a combination of uniformly distributed samples from all of the places under all conditions, recognition rates go up to 82%. Hence, as expected, recognition performance in similar viewing illumination is much higher compared to that in different illumination.

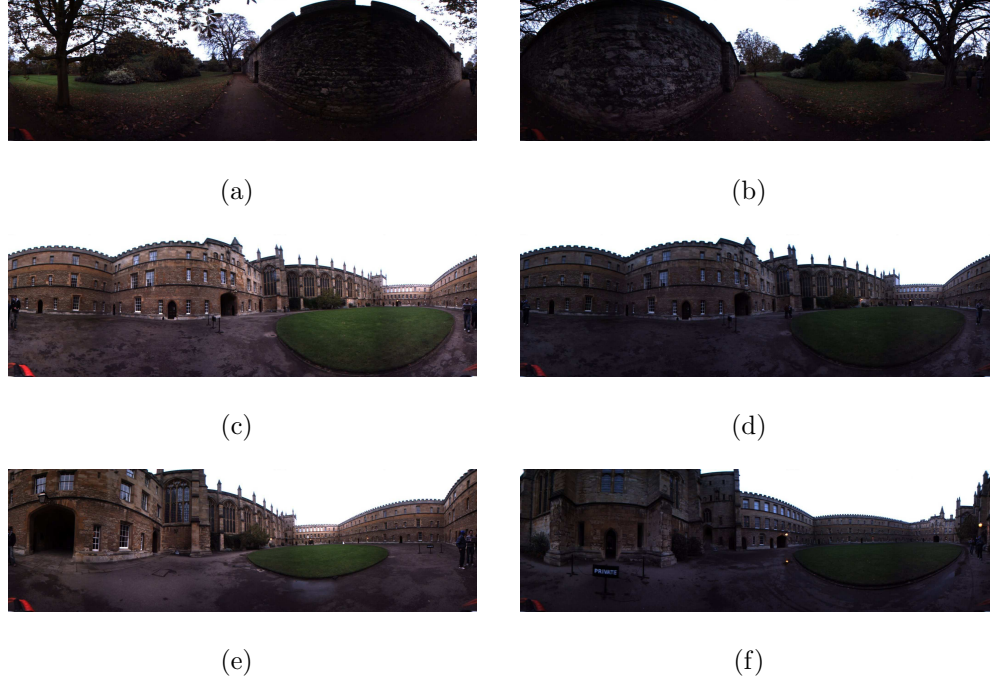


Figure 3.8. Sample images from New College data set where samples from learning set are on the left column while test images are on the right column. (a) and (b) With different viewpoints; (c) and (d) With different illumination levels; (e) and (f) Both viewpoints and illumination levels varying.

In order to understand the underlying performance, recognition rates pertaining to specific places are also studied. The confusion matrix associated with similar learning and test illumination conditions is as given in Table 3.7. Interestingly, the first three places which are comprised of man-made structures are confused with each other while the last four places which are of natural scenery tend to be confused with each other. This may be partly attributed to the discrimination power of the Cartesian and non-Cartesian features. The confusion matrix associated with learning in day illumination while testing in evening illumination is as shown in Table 3.8. Again, it is observed that the misclassifications occur among scenes that would also appear similar to an human observer such as different park views or different street views.

The recall and precision curves for different learning and recognition scenarios are obtained via varying the threshold parameter $\tau \in [0.3, 1.0]$ as shown in Figure 3.9. The resulting recall rates for specified precision levels are summarized in Table 3.9.

Table 3.6. Recognition rates for varying learning and recognition scenarios with New College data set.

Learning	Test	Recognition Rates (%)
Day	Day	85.2
Day	Evening	59.4
Evening	Day	51.6
Day+Evening	Day+Evening	82.3

Table 3.7. Confusion Matrix: Learning and recognition are both done under day illumination with New College data set.

		Recognized as (%)						
		P1	P2	P3	P4	P5	P6	P7
Actual	P1	97.2	1.4	1.4	0.0	0.0	0.0	0.0
	P2	18.1	76.2	4.4	1.3	0.0	0.0	0.0
	P3	7.1	2.4	88.2	2.3	0.0	0.0	0.0
	P4	1.2	1.3	1.2	87.6	5.2	1.3	2.2
	P5	1.0	0.0	0.0	9.8	76.9	7.7	4.6
	P6	1.3	0.0	0.0	1.3	14.6	78.4	4.4
	P7	5.2	0.0	0.0	9.1	11.1	10.2	64.4

At 100% precision (with $\tau = 0.98$) the maximum achievable recall rate is around 18%. However, for 90% precision (with $\tau = 0.50$), the recall rate increases to around 89% for learning and testing in similar illumination conditions and around 5% (with $\tau = 0.98$) for learning and testing in completely different illumination conditions. For comparative purposes, two recent related work are considered [102, 103]. However, in both of the papers, the problem considered is the loop closure problem. Hence,

Table 3.8. Confusion Matrix: Learning is done under day illumination while recognition is done under evening illumination with New College data set.

		Recognized as (%)						
		P1	P2	P3	P4	P5	P6	P7
Actual	P1	88.2	6.4	2.1	3.3	0.0	0.0	0.0
	P2	34.3	54.5	3.3	2.2	1.3	1.1	3.3
	P3	27.2	3.1	63.3	5.2	0.0	0.0	1.2
	P4	16.3	1.1	10.3	58.4	9.6	2.1	2.2
	P5	14.1	0.0	1.2	11.1	48.5	13.0	12.1
	P6	0.0	0.0	0.0	2.6	38.1	53.2	6.1
	P7	1.4	1.1	2.2	25.6	24.2	7.1	38.4

the reasoning in both use additional information including the motion model or the sequentiality of the images. However, in our work, none of these are used. With this in mind, the recall rate is reported as 47% at 100% precision in [102] and as 70% at 100% precision in [103] whereas our recall rate is 18% at 100% precision. However, at 99% precision, a recall rate of 42% is attained in our study – which is close to that obtained in [102]. The results suggest that BuS representation can be used for coarse placement in loop closure in SLAM problems due to its performance on recognition and computation time.

Finally, we extend the outdoor experiments by considering the case of unknown places. For this, the learning is done with only six of the places in day illumination. The resulting recall rates for specified precision levels are summarized in Table 3.10. It is observed that at 100% precision (with $\tau = 0.95$), the maximum achievable recall rate is around 18.9%. However, for 90% precision (with $\tau=0.45$), the recall rate increases to around 91%.

Table 3.9. Recall rates for specified precision levels as given in Figure 3.9.

			Precision (%)		
Recall (%)	Learning	Test	100	99	90
	Day	Day	18.4	42.2	89.1
	Day	Evening	-	-	6.4
	Evening	Day	-	-	3.3
	Day + Evening	Day + Evening	12.2	32.4	82.1

Table 3.10. Recall rates for specified precision levels with an unknown place.

			Precision (%)		
Recall (%)	Learning	Test	100	99	90
	Day	Day	18.9	53.6	91

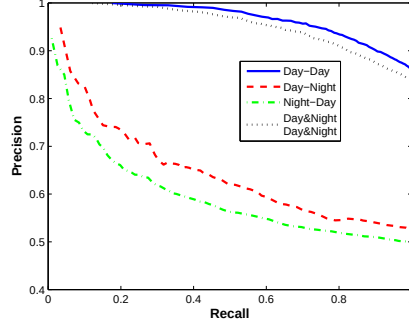


Figure 3.9. Precision-recall curves for varying learning and recognition scenarios with New College data set.

3.4.3. Both Indoors and Outdoors: Comprehensive Data Set

Even though the data sets are among the most extensive available to the best of author's knowledge, the number of places they contain is still fairly small. In an effort to amend this, in the last set of experiments, we consider all the indoors COLD and outdoors New College data sets simultaneously. Since the first data set contains 3 different buildings with 14 places altogether and the second data set contains 7 places, there are 21 different places. The robot learns these places by being given visual data associated with each different place – all in cloudy (COLD data set) or early day (New College data set) viewing conditions. For convenience, both are referred to as cloudy illumination. In the recognition stage, visual data sets that are either taken in different times at different robot poses, illumination conditions or both. Note that New College data set has only two illumination conditions – day (cloudy) or evening (night). The recognition rates are presented in Table 3.11. It is observed that the recognition rates are still fairly high - 81% in same illumination similar to only COLD or only New College data set experiments. In case of different illumination conditions, performance deteriorates to some extent depending on the change in illumination as expected. Note that interestingly, the resulting performance is higher than that in extended route experiments with COLD data set. This is attributed to the greater visual diversities among different indoors laboratories and outdoors environment. The precision-recall curves generated via varying the threshold parameter $\tau \in [0.3, 1.0]$ are presented in

Figure 3.10 for comprehensive data set. For cloudy condition the precision is at 99 % upto 20 % recall rate. Again, for sunny illumination condition, the results are better than the illumination at night.

Table 3.11. Recognition rates for the comprehensive (COLD + New College) data set.

Learning	Recognition	Recognition Rates (%)
Cloudy	Cloudy	81.4
Cloudy	Sunny	75.2
Cloudy	Night	57.1

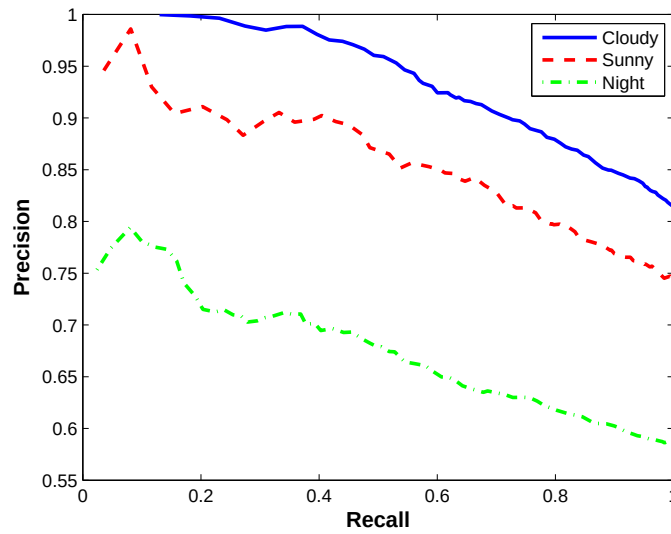


Figure 3.10. Precision recall curves for the comprehensive (COLD + New College) data set.

3.4.4. Summary

The experimental results on two different data sets (indoors and outdoors) along with the comprehensive data set comprised of both suggest that bubble space representation enables comparatively high recognition rates with acceptable precision under similar learning and testing illumination conditions. In both data sets, recognition

rates are around 80%. Under different illumination conditions, performance is not as high – as is also the case in the other state-of-the art approaches. It is observed that degradation in performance is greater in New College data set. This is attributed to the fact that outdoors evening illumination conditions are comparatively dimmer as compared to indoors evening illumination conditions. Furthermore, comparative results with COLD data sets indicate much lower variation in performance and computation time. Interestingly, as the dimension of bubble descriptor is reduced considerably (by a factor of 60), the recognition performance degrades by around 7%. However, recognition rates remain relatively high. Finally, recognition rates continue to be around 80% even as the number of places are increased as shown by the extended COLD data set and comprehensive data set results. The advantages of BuS representation include the following: i) Flexibility in integrating visual features; ii) Both detailed and holistic representation of places; iii) Construction in an incremental manner as new visual observations are made; iv) Rotational invariance with respect to heading changes; v) Relatively high recognition rates under varying viewing conditions with relatively lower standard deviation and significantly lower codebook extraction time. To the best of author’s knowledge, although research on place representation has been very active and has made significant progress in recent years, we are not aware of descriptors that have all these features simultaneously.

3.5. Conclusion

In this chapter, we present bubble space which is a novel representation for “places” (nodes) in topological maps. It is based on bubble memory where visual feature values and their local S^2 –metric relations from robot’s viewpoint are simultaneously encoded on a deformable spherical surface. In bubble space, bubble surfaces extend bubble memory to accommodate two new variables – varying robot pose and multiple features. The usage of multiple robot poses and features provides a thorough modeling of each place as suggested by the experimental results. Second, for learning and recognition, bubble surfaces are transformed into bubble descriptors which are holistic representations of places while being rotationally invariant and computable

in an incremental manner with each new set of visual feature observations. Thus, bubble space simultaneously provides detailed (bubble surfaces) and holistic (bubble descriptors) representations of places. The proposed representation is evaluated in a place learning and recognition task with multi-class SVM based decision-making on two benchmark visual data sets – the indoors COLD dataset and the outdoors New College data set. Two aspects of our experimental results are particularly noteworthy. First, it is observed that BuS representation enables high recognition rates with low standard deviation under strict testing in stable and varying illumination conditions. Second, the codebook extraction time per frame is significantly lower. These results suggest that the representation is practical for usage in various real-time robotic tasks such as navigation and map building. Furthermore, the recall rates at 99% precision convey that the BuS representation can be used in conjunction with localization.

4. CUE DESCRIPTOR SEMANTIC FORESTS FOR TOPOLOGICAL PLACE RECOGNITION

Topological mapping for autonomous robot systems has been receiving increased research interest [6, 9, 82, 104–106]. In most of the proposed approaches, the robot uses a multitude of cues for generating a description of each place - regardless of whether cue integration is explicitly addressed or not. Each cue conveys a separate aspect of the scene which may be either redundant or complementary in nature with individual limits on accuracy and availability. In this chapter, the problem of cue integration for place recognition is considered. Cue integration has been studied generally [107, 108] and applied in areas such as visual segmentation and tracking [109–113]. In more recent work, the effectiveness of cue integration has also been demonstrated in place recognition [15, 16]. In all these work, performance is shown to improve considerably via the integration of information provided by different cues.

In this chapter, our focus is another related aspect – namely integrative cue representation. Cue representation is closely related to the efficiency of place learning and recognition. Such representation should be able to incorporate any type of cues - sensory feature or semantic. As such, introducing sensory feature cues should not require the internal representations to be rebuilt from scratch [17]. Rather it should be possible to update the representation incrementally. Simultaneously, the representation should enable graceful degradation of performance in case of missing sensory cues or observations. The contribution of this chapter is a new integrative cue representation - namely cue descriptor semantic forests (CDSF) that can address these issues effectively.

A CDSF is a collection of cue descriptor semantic trees - where each tree organizes data associated with one cue descriptor in a semantic hierarchy. There are two new aspects of this representation: cue descriptors and cue descriptor semantic trees. A cue descriptor is conducted by a distinct group of bubble surfaces associated with different sensory features - building on our previous work on bubble space for topological place

representation [19]. They are obtained via decomposing bubble descriptors into vectors with much smaller dimensions [114]. As such, each cue descriptor preserves the relative S^2 geometry of the features - similar to bubble descriptors. However, now, instead of one bubble descriptor that is associated with the whole set of bubble surfaces, a set of cue descriptors exists where each is associated with a small set of bubble surfaces. Next, for each cue descriptor, a cue descriptor semantic tree (CDST) is constructed. The semantics of each tree is based on an ordered list of semantic cues in a supervised manner. A CDST organizes the discriminant functions in a tree hierarchy based on increasingly expressive set of semantic cues and associates each sensory feature cue with a cue weight that measures its relative semantic expressiveness. As each sensory cue is associated with one particular cue descriptor semantic tree, the resulting representation is a cue descriptor semantic forest. With CDSF, the integration of different sensory feature and semantic cues becomes very simple regardless of their type or their number without requiring special processing for data adaptation while enabling relative semantic expressiveness via cue weights. With this new representation, topological place recognition is accomplished by an iterative process via traversing all the CDSTs concurrently. At each level of the CDSTs, a new semantic cue is determined via the weighted accumulation of sensory feature cues in a manner that depends on the so far collected set of semantic cues - which is then incremented with the newly determined semantic cue. This process is repeated until all the semantic cues have been uncovered. Extensive experimental results on benchmark datasets reveal the following:

- Shorter training times as new sensory feature cues become available.
- Robustness in case of completely missing or partially changed (occlusions) sensory feature cues.

The outline of this chapter is as follows: First, related literature is reviewed in Section 4.1. Sensory feature cue descriptors are described in Section 4.2. The formulation of CDSF representation is presented in Section 4.3 followed by its learning in Section 4.4. Cue integration for place recognition using CDSF is explained in Section 4.5. The performance of CDSF in both indoors and outdoors place recognition is demonstrated in Section 4.6 via an extensive set of experiments including a com-

parative study with a previously proposed representation, the addition of new sensory cues, the case of missing sensory cues and occlusion. The chapter concludes with a brief summary.

4.1. Related Literature

Cues are primarily of two types: sensory feature and semantic. In this section, we review each from the perspective of cue integration for place recognition.

4.1.1. Sensory Feature Cues

Sensory cues are low-level features obtained from the sensory data via simple processing such as filter responses. They may be from a single sensing modality as alternative interpretations of the data from the same sensor may contain different information [115]. Alternatively, they may have been obtained from different modalities such as cameras, laser, lidar and sonars where each modality captures a different aspect of the environment [115]. These cues are also commonly referred to as low-level cues. Cue integration has been initially based on sensory cues. Their representational form reflects the features used in the map model. Typically, SIFT [75] and SURF [76] are computed as local features [116–118]. Alternatively, global features such as gist of a scene [71], fingerprints [77], CRFH [119], PACT [120] or bag-of-words [121] are used global sensory cues [117]. Robustness can be improved by using a mixture of local and/or global cues [15, 117].

Once multiple cues are available, the next aspect pertains integration of these cues into a composite measure [122–124]. Of course, integration is closely related with representation. Within robotics community, the simplest approach is based on building a new representation that combines different cues in a single feature vector and using a standard pattern recognition approach [17, 19, 125, 126]. An incremental learning approach as proposed in [17] adapts its decision function in order to integrate the new information. While these approaches may lead to good performances in specific tasks, they have a couple of shortcomings [113, 127]. As the number of cues increases, so does

the dimension of the feature vector which implies longer re-learning and recognition times and greater memory requirements [128]. As additional cues become available, feature representations change which implies that all the learning must be repeated from scratch. Finally, one cue being unreliable or unavailable will have adverse effects on performance.

Alternatively, incoming sensory data are organized as a group of cue vectors of smaller dimensions rather than a single feature vector of much bigger dimension [113–115]. Voting based approaches [112,129] are easy to implement, however as information provided by a minority of cues are completely ignored, they may lead to poor performance in cases when this information is vital. Minimum-variance unbiased estimators are simple linear integrators where cues - weighted on basis of their reliability - are summed [130]. In discriminative accumulation, cue integration is based on weighting and summing margins of each cue obtained from the associated pre-learned support vector machines (SVM) [113,115]. A measure of confidence is introduced as a means to better fuse different cues and also provide the consumer of the information with some information about the certainty in the end result [15]. This framework is extended to a probabilistic Bayesian reasoning where each cue is represented by a posterior probability distribution [130–132]. Bayesian schemes, while providing for more accurate cue integration, are associated with high computational complexity which can be problematic in real-time applications [133].

4.1.2. Semantic Cues

Complex environments have many aspects that are hard to capture based purely on sensory feature cues. This problem is addressed via the introduction of semantic cues such as Spatial Semantic Hierarchy [104] or [134]. Semantic cues encode some abstraction of spatial knowledge and specify different aspects of its ontology such as its geographical location and appearance. They are usually obtained from either raw data or sensory cues or even previously computed semantic cues [135] via more complex processing and/or supervision. In place recognition, semantic cues have been introduced in two ways – spatial semantics and object-derived semantics. Spatial se-

mantics include geographic locations and the classification of spaces (rooms, corridors, halls) [134, 136] or information about the connectivity of space such as offices being typically connected to corridors [105]. In object-derived semantics, objects are used in order to extract spatial semantics [134, 137–140]. In the hierarchical extension, information from low level sensor data are combined with high level objects and spatial properties based on factor graphs [141, 142]. In fact, this idea can be extended even further as cues have a multi-dimensional and ordered nature as suggested by the taxonomy of places [143]. In this taxonomy, each place is defined by an ordered list of semantic attributes. From place recognition perspective, each attribute can be used to narrow the possibilities associated with a current place - thus leading to ordered integration of multiple semantic cues. Although this approach has been adopted in scene classification - which albeit being related, is nevertheless different than place recognition⁵. In these approaches, low-level features are used in conjunction with multiple semantic scene attributes [135, 144–146]. The representation proposed in this chapter is motivated by this idea. In cue descriptor semantic forests, multiple sensory feature cues are combined with an ordered set of semantic cues. First, each sensory feature related cue is encoded by a cue descriptor that accommodates all the respective observations while preserving their relative S^2 geometry. Secondly, each cue descriptor semantic tree enables the amalgamation of one type sensory cue with an ordered set of semantic cues via an hierarchy of discriminant functions and cue weights. Then CDSF is used with a combination of sensory feature cue accumulation and semantic cue propagation for place recognition.

4.2. Sensory Feature Cue Descriptors

Sensory feature cue descriptors are formulated in bubble space. In this Section, the formulation of cue descriptors is presented.

⁵In scene classification, in contrast to place recognition problems, few good exemplary images are used during learning and testing.

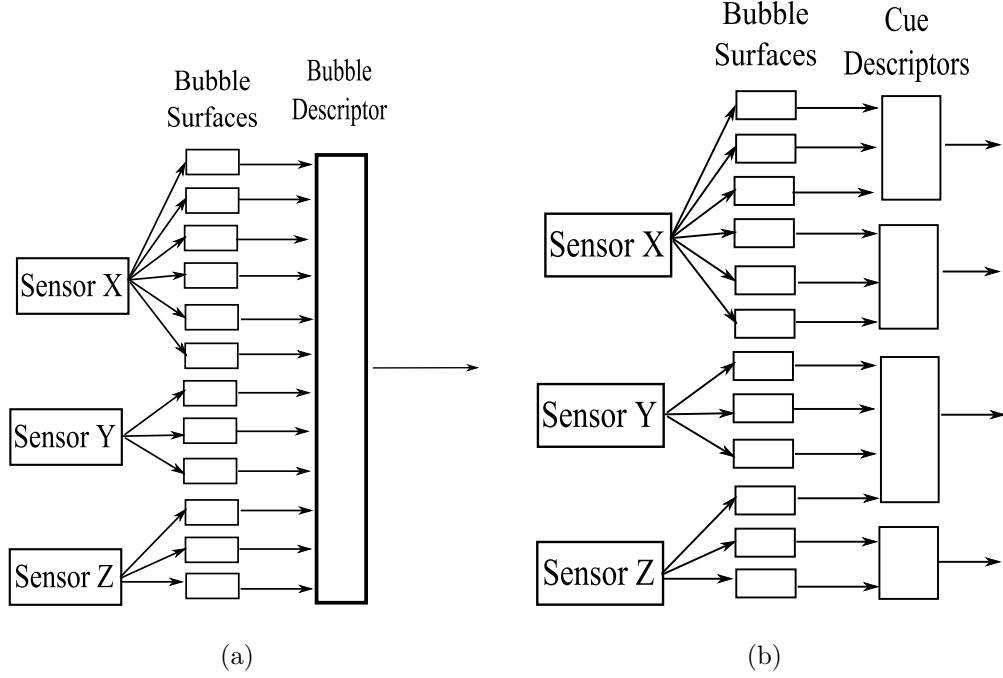


Figure 4.1. Representation in bubble space. (a) A single bubble descriptor (BuS); (b) A multitude of cue descriptors (CD).

4.2.1. Cue Descriptors

If the sensory features can be categorized into $N_c \ll N_v$ groups, then accordingly the bubble surfaces can be grouped into N_c subsets where each subset consists of bubble surfaces whose features are related. In this perspective, each bubble descriptor $I(x_j)$ is really an aggregation of N_c different components - again each component corresponding to one different cue as:

$$I(x_j) = \sum_{c=1}^{N_c} e_l \otimes I_c(x_j)$$

where the vectors e_l are the unit vectors in $\mathbb{R}^{N_c}$ and $\otimes$ is the Kronecker product [147]. Hence, it can be broken into smaller vectors $I_c(x_j) \in \mathbb{R}^{M_c(H_1+1)(H_2+1)}$ that embed invariants associated with one group of sensory features. We refer to $I_c(x_j)$ as a cue descriptor associated with one group of M_c sensory filters. Mathematically,

$$I_c(x_j) = [I_{c,00}(x_j) \dots I_{c+M_c, H_1 H_2}(x_j)]^T$$

The set of cue descriptors is then defined as $\{I_c\}_1^{N_c}$. Consequently, instead of a single vector with a very large dimension, a set of N_c cue vectors with much smaller dimensions are generated as shown in Figure 4.1b.

4.3. Cue Descriptor Semantic Forest

A cue descriptor semantic forest is a collection of cue descriptor semantic trees (CDST). Each CDST enables cue descriptors to be supplemented with semantic cues. In this section, we first discuss semantic cues. This is followed by the formulation of CDSTs.

4.3.1. Semantic Cues

Semantic cues are in general abstract concepts that specify particular properties of each place. In particular, spatial semantics is considered. As an example, one semantic cue may be the geographical description (e.g. the building where the place is in) and the second cue may describe its appearance (e.g. the type of the place, i.e. kitchen, corridor, bathroom, etc.). Let N_L be the number of semantic cues. Each cue p_l , $l = 1, \dots, N_L$ has values in $\mathcal{P}_l$ - a finite discrete set of values with cardinality $\|\mathcal{P}_l\| = N_l$

An l -dimensional semantic cue vector p^l consists l of these semantic cues. It is defined as $p^l = [p_1 \dots p_l]^T$. These cues are assumed to have a natural order among them - as suggested by the taxonomy of places [143]. The natural order is based on informativeness or specificity. The semantic cue vector p^l is constructed in a manner as to reflect this order in an increasing degree of specificity. Hence, the $l-1$ th semantic cue is assumed to have less informativeness or specificity as compared to the l th semantic cue. For example, as geographical description is less specific than appearance, the ordering will start with $l = 1$ being the geographical location and $l = 2$ being the appearance.

We assume that each place is associated with the maximum N_L number of seman-

tic cues – namely $p = p^{N_L}$. Thus, with the set of semantic cues consisting of geography and appearance, each place is denoted with a semantic cue vector $p = [p_1 \ p_2]^T$ where $p_1 \in \mathcal{P}_1$ with cardinality $\|\mathcal{P}_1\| = N_1$ and where $p_2 \in \mathcal{P}_2$ with cardinality $\|\mathcal{P}_2\| = N_2$. Hence, the number of possible different places $\|P\| = \prod_{i=1}^{N_L} N_i$.

4.3.2. Cue Descriptor Semantic Trees

The structure of the CDSTs is designed to encode the different semantic cues and their ordering. If the semantic cue vector $p = p^{N_L}$ is an N_L dimensional vector, then each CDST is organized according to N_L levels. Each level l , $l = 1, \dots, N_L$ is associated with one semantic cue and has at maximum $\prod_{j=1}^l N_j$ nodes. Each node is paired with one particular l -dimensional semantic cue vector p^l . The order of the levels reflect that of the semantic cues. Consider a node at level l with a semantic cue vector p^l . As nodes are related to each other via parent-children relationship, parent attributes are propagated to the children levels. Thus, if p^{l-1} denotes the (partial) semantic cue associated with its parent, then $p^l = \begin{bmatrix} p^{l-1} \\ p_l \end{bmatrix}$. Note that $p^0 = \emptyset$. For example, consider the CDST in Figure 4.2. Nodes at first level are associated with the first semantic cue having $N_1 = 3$ values. In the second level, 1-dimensional semantic cues are augmented with the second semantic cue - having $N_2 = 3$ values. Hence, there can be at maximum $\|P\| = 9$ places (terminal nodes). However, if only a subset of these places have been visited, the number of terminal nodes will be less - in this case $N_p = 8$.

Furthermore, each node at level l , $l = 1, \dots, N_L$ is associated with a discriminant function d_{c,p^l} and a cue weight $\kappa_{c,p^l} \in R^{\geq 0}$. The discriminant functions classify the cue descriptors associated with the parent nodes with respect to a particular value of l - semantic cue. They are constructed based on support vector machines (SVM) - a standard approach that can continue to be used as sensory data and the number of places become much more extensive. The cue weights encode the varying discriminative power of each cue descriptor in regards to (possibly partial) semantic cues. As such, they implicitly encode place specific prior knowledge. As the top node is associated

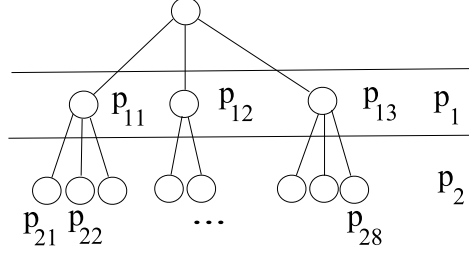


Figure 4.2. A CDST structure with $N_L = 2$.

with $p^0 = \emptyset$ (no semantic cue), it is also not associated with a classifier or a cue weight. Both the discriminant functions and the cue weights are obtained via two separate learning processes as explained in Section 4.4.

4.4. Place Learning

In place learning, the goal is to construct a CDSF. At each place $p \in \mathcal{P}$, the robot looks at its surroundings from a multitude of base points x_{pj} , $j = 1, \dots, M_p$. At each base point x_{pj} , it generates a set of cue descriptors $I_c(x_j)$, $c = 1, \dots, N_c$. Hence, for each cue c , a set of cue descriptors $\mathcal{C}_{cp}$ is generated:

$$\mathcal{C}_c = \bigcup_{p \in \mathcal{P}} \{I_c(x_{pj}) \mid j = 1, \dots, M_p\}$$

Place learning is based on the set $\mathcal{C}_c$. In this chapter, the CDSF is learned in a supervised manner. Thus, each cue descriptor $I_c(x_{pj}) \in \mathcal{C}_c$ is associated with a semantic cue vector $p'(x_{pj})$.

As introduced earlier, there are two aspects to learning. First, a cue descriptor semantic tree is constructed for each sensory feature cue. Secondly, each node in each CDST is associated with a cue weight. The cue weights encode the relative expressiveness of different sensory feature cues in determining a particular semantic cue vector p^l . For this, the set of cue descriptors $\mathcal{C}_c$ is partitioned into two subsets. The first group $\mathcal{C}_c^+$ is used for constructing the CDSF. The second group $\mathcal{C}_c^{++}$ is used in learning the cue weights.

4.4.1. CDSF Construction

A CDSF consists of N_c CDSTs. Each is associated with one particular sensory feature cue c , $c = 1, \dots, N_c$ and is constructed based on the cue descriptors $\mathcal{C}_c^+$ constructed in the learning phase. All the cue descriptors are assumed to be labeled with N_L -dimensional semantic cue vectors in a supervised manner. In particular, a one-against-all SVM at each node is constructed. The construction proceeds in a hierarchical and iterative manner - starting with the nodes at first level. It is repeated iteratively until level N_L is reached.

Consider a node at level l associated with semantic cue p^l . The corresponding discriminant function d_{c,p^l} encodes the parameters of an separating hyperplane in a one-against-all SVM [148]. It is learned based on the subset of $\mathcal{C}_c^+$ associated with its parent node. In particular, the set of cue descriptors associated with p^{l-1} are classified with respect to p_l th value of the l th semantic cue in a one-against-all SVM. Hence, at level $l = 1, \dots, N_L - 1$, a set of $N_{l-1} \times N_l$ discriminant functions are computed. The discriminant functions associated with the terminal nodes enable the classification of places with N_L -dimensional semantic cue vector. As this is done for each cue, a CDSF is generated.

4.4.2. Learning Cue Weights

Each node (except the top node) in each CDST is associated with a cue weight. As stated earlier, the cue weights encode the relative expressiveness of different sensory feature cues in determining a particular semantic cue vector p^l . In particular, we propose an approach where cue weights are learned in a supervised manner - motivated by merging multiple graphs into a single graph [149]. In this approach, for each particular set of semantic cues p^l , the cue weights are determined so that a transition probability matrix derived by combining similarity measures from different sensory feature cues approaches its ideal counterpart.

The learning process starts with $l = 1$ and proceeds to $l = N_L$. Consider semantic

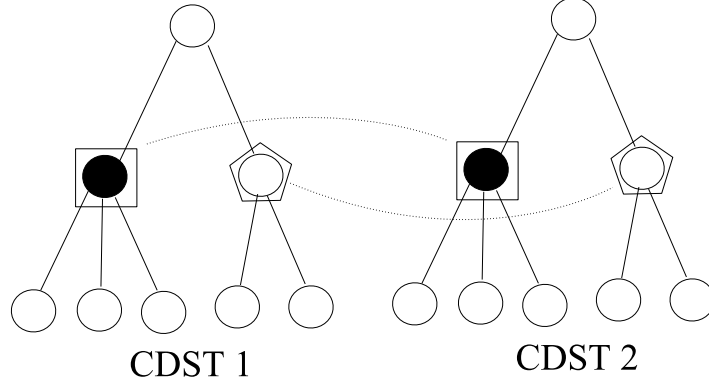


Figure 4.3. Corresponding nodes for level $l = 1$ in a 2-CDSF. One set is shown with black-filled circles while the second set is shown with the white circles.

cue vectors p^{l-1} , $l = 1, \dots, N_L$. Each CDST has a unique node that is associated with p^{l-1} . Of course, in case of no associated cues, there won't be such a node. These nodes - one from each CDST at maximum - are referred to as the corresponding nodes. For a sample CDSF, the corresponding nodes are as shown in Figure 4.3.

Next, consider a particular sensory feature cue c . Recall that the cue descriptors $\mathcal{C}_c^{++}$ are collected from M_p base points at N_p places. Let $M^{++} = \sum_{p=1}^{N_p} M_p$ denote the cardinality of $\mathcal{C}_c^{++}$. For each cue descriptor $I_c(x_{pj}) \in \mathcal{C}_c^{++}$ and for each $p^l \in \mathcal{P}_l$, define a distance value $s_{c,p^l}(x_{pj})$

$$s_{c,p^l}(x_{pj}) = \frac{1}{d_{c,p^l}(I_c(x_{pj})) + \chi}$$

where $p^l = \begin{bmatrix} p^{l-1} \\ p_l \end{bmatrix}$ and $\chi > 0$ is an a priori determined parameter. The results are collected together in a distance matrix $S_{c,p^{l-1}}$ of dimension $M^{++} \times N_l$. This is repeated for each CDST. Hence, a set of N_c matrices is defined for semantic cue p^{l-1} .

Following, the distance matrices $S_{c,p^{l-1}}$, $c = 1, \dots, N_c$ are integrated into a single composite matrix $W_{p^{l-1}}$ of dimension $M^{++} \times N_l$ with entries:

$$w_{p^l}(x_{pj}) = \frac{1}{Z_p} \tilde{w}_{p^l}(x_{pj})$$

Now, each sensory feature cue c and semantic cue p^l is associated with a cue weight $\kappa_{c,p^l} \in R^{\geq 0}$. The entries are defined using the cue weights κ_{c,p^l} , $c = 1, \dots, N_c$ via exponential decay and normalization:

$$\begin{aligned}\tilde{w}_{p^l}(x_{pj}) &= e^{-\sum_{c=1}^{N_c} \kappa_{c,p^l} s_{c,p^l}(x_{pj})} \\ Z_p &= \sum_{p^l \in \mathcal{P}_l} \tilde{w}_{p^l}(x_{pj})\end{aligned}$$

With these definitions,

$$w_{p^l}(x_{pj}) \in [0, 1] \quad \text{and} \quad \sum_{p^l \in \mathcal{P}_l} w_{p^l}(x_{pj}) = 1 \quad (4.1)$$

Hence, W_{p^l} is a conditional probability matrix.

As each cue descriptor $I_c(x_{pj})$ in the learning set $\mathcal{C}_c^{++}$ is associated with a semantic cue vector $p'(x_{pj})$, it is possible to construct the ideal conditional probability matrix $W_{p^{l-1}}^*$ of dimension $M^{++} \times N_p$ with entries:

$$w_{p^l}^*(x_{pj}) = \begin{cases} 1 & \text{if } p^l \prec p'(x_{pj}) \\ 0 & \text{otherwise} \end{cases}$$

where the symbol $\prec$ denotes subvector relation. Ideally, with optimal weights κ_{c,p^l} , the two matrices should be equal - namely $W_{p^l} = W_{p^l}^*$. As the cue weights κ_{c,p^l} are unknown, the problem of finding the optimal κ_{c,p^l} is defined as a minimization problem:

$$\kappa_{c,p^l} = \arg \min_{\kappa_{c,p^l}} ||W_{p^{l-1}}^* - W_{p^{l-1}}||$$

subject to the two constraints as defined by Equation 4.1. Using Kullback-Leibler

divergence,

$$\|W_{p^{l-1}}^* - W_{p^{l-1}}\| = \sum_{p=1}^{N_p} \sum_{j=1}^{M_p} \sum_{p_l \in \mathcal{P}_l} w_{p^l}^*(x_{pj}) \log \frac{w_{p^l}^*(x_{pj})}{w_{p^l}(x_{pj})}$$

The solution is determined by taking partial derivative of $\|W_{p^{l-1}}^* - W_{p^{l-1}}\|$ with respect to κ_{c,p^l} and setting it to zero. This yields a set of nonlinear equations:

$$\sum_{p=1}^{N_p} \sum_{j=1}^{M_p} \sum_{p_l \in \mathcal{P}_l} (w_{p^l}(x_{pj}) s_{c,l}(x_{pj}) - w_{p^l}^*(x_{pj}) s_{c,l}(x_{pj})) = 0$$

The cue weights κ_{c,p^l} can be approximated by linear approximation and iterative estimation. This procedure is repeated for each semantic cue vector p^{l-1} , $l = 1, \dots, N_L$.

4.5. Cue Integration for Recognition

In recognition mode, the robot is at a place and looks around from a base point x . It is assumed that this place is defined by a N_L -dimensional semantic cue vector $p^* = [p_1^* \dots p_{N_L}^*]^T$. Of course, the robot does not know p^* or its base point x . As it looks around, it collects a set of sensory feature cues and constructs the respective descriptors $\{I_c\}_{c=1}^{N_c}$. The goal of place recognition is to determine p^* using $\{I_c\}_{c=1}^{N_c}$ and the learned CDSF. There are two aspects to recognition: sensory feature cue accumulation and semantic cue propagation. At each level of the CDSTs, a new semantic cue is determined via the weighted accumulation of sensory feature cues considering the semantic cues collected so far. This is followed by incrementing the semantic cues with this new semantic cue and propagating it to the next level of CDSTs for determining the next semantic cue. This is repeated until all the semantic cues have been uncovered.

4.5.1. Sensory Feature Cue Accumulation

Consider a partial set of semantic cues $p^{(l-1)*}$ that are determined. The next semantic cue value p_l^* needs to be determined. For each $p_l \in \mathcal{P}_l$, consider the l -dimensional augmented semantic cue vector $p^{l'}$ defined as:

$$p^{l'} = \begin{bmatrix} p^{(l-1)*} \\ p_l \end{bmatrix}$$

The integration of all the sensory cues c , $c = 1, \dots, N_c$ is done via weighted accumulation using the discriminant functions and the cue weights. In particular, the robot decides on the p_l^* as:

$$p_l^* \in \arg \max_{p_l \in \mathcal{P}_l} \sum_{c=1}^{N_c} \kappa_{c,p^{l'}} d_{c,p^{l'}}(I_c)$$

Here, $\kappa_{c,p^{l'}}$ are cue weights associated with p_l and the cue set $c = 1, \dots, N_c$.

4.5.2. Semantic Cue Propagation

Each CDST is traversed concurrently with the other CDSTs. In this traversal, semantic cues determined upto level l are propagated to the next level in a hierarchical manner. This is done starting with the first semantic cue p_1 and going sequentially to p_{N_L} . At each level, p_l^* is determined using weighted cue accumulation as explained in Section 4.5.1. The semantic cue vector is then augmented as $p^{l*} = \begin{bmatrix} p^{(l-1)*} \\ p_l^* \end{bmatrix}$. This is then propagated to the next level $l + 1$ and the whole process is repeated. The resulting semantic cue is denoted as p^* . The case of two-dimensional semantic cue in Figure 4.4. First, at level 1, the value p_1^* of first semantic cue is determined using weighted accumulation of all N_c cues. Next, using this value p_1^* , the robot proceeds to find the value of the second semantic cue. It again computes the second semantic

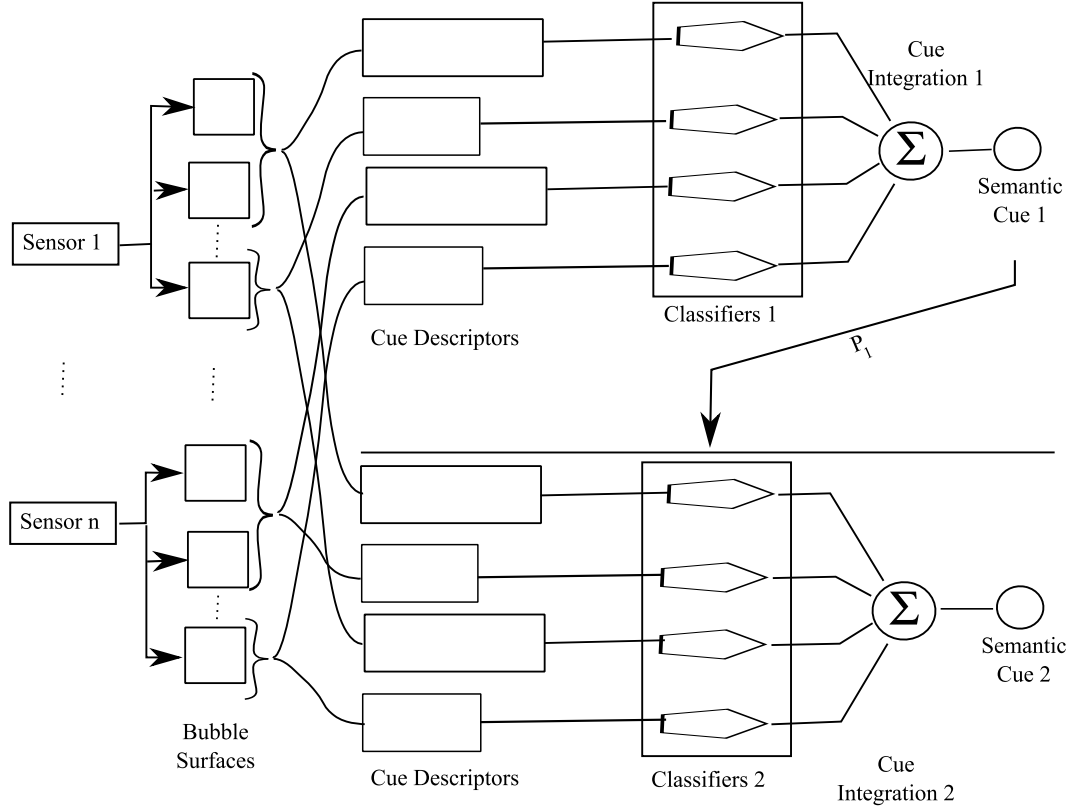


Figure 4.4. Hierarchical propagation of semantic cues.

cue value p_2^* via weighted accumulation. As this is the terminal node, the place label is $p^* = [p_1^* p_2^*]^T$. Therefore, instead of obtaining the semantic cue vector for each cue at one shot and combining these in order to find a final decision, semantic cues are collected in an incremental and ordered manner.

4.6. Experiments

This Section presents experimental results to evaluate the proposed representation. The experiments are similar to those that were done in recent related work [8, 19, 95] and last chapter where the task is to recognize the current place based on previous learning. The test data differs either in robot pose, viewing illumination, time or any combination of from the learning data. It is assumed that the robot has no access to any motion, relative motion or any other sensing information. Let it be remarked that as the performances of different representations including BuS representation are compared in a detailed manner in these studies, they won't be repeated here. Instead, we will present comparative performance of bubble descriptor (BuS) consisting sin-

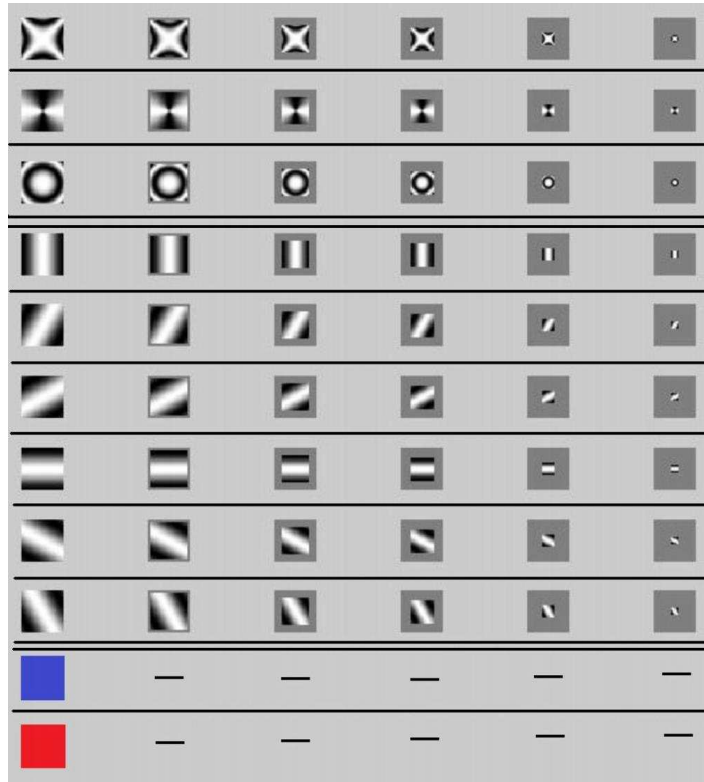


Figure 4.5. Cues that are used in the experiments. In 3-CDSF, these features are organized into three different cues as shown by the double solid lines. In 11-CDSF, eleven CDSTs are constructed where the cue descriptors are obtained via grouping these features into 11 groups as shown by the solid lines.

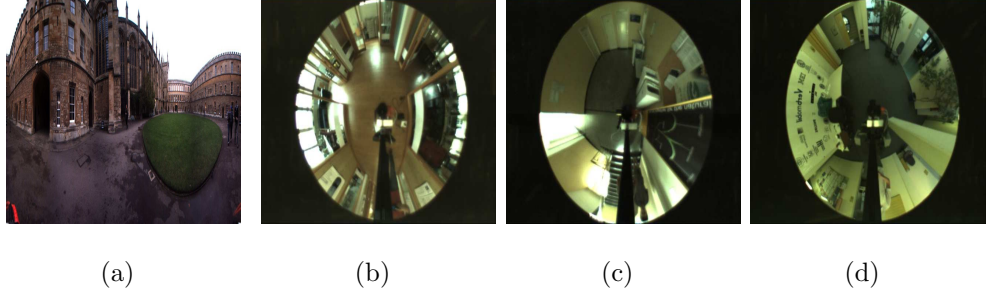


Figure 4.6. Sample images. (a) Panoramic image from New College dataset; (b), (c) and (d) Omni-directional images from Freiburg Lab, Ljubljana Lab and Saarbrücken Lab respectively.

gle large-dimensional vector with CDSF constructed with multiple smaller-dimension vectors.

The bubble surfaces are formed with sensory observations from $N_s = 40$ different viewing directions with parameters $\rho_0 = 1$, $\rho_{min} = 0.5$, $\rho_{max} = 1.5$ and $\epsilon = 2^\circ$. These viewing directions are determined automatically via looking around in a free viewing mode as described in detail in [100] and Chapter 2. The corresponding bubble descriptors are constructed with $H_1 = H_2 = 6$ number of harmonics, however only the first 40 rotational invariants are used since the remaining are observed to be nondiscriminating. There are 56 features used as shown in Figure 4.5, each bubble descriptor is of dimension $56 \times 40 = 2240$. In 3-CDSF, there are three CDSTs constructed corresponding to three sets of cues as shown in Figure 4.5 where the associated cue descriptors are of dimension 720, 1440 and 80 respectively. In 11-CDSF, the CDSF consists of eleven CDSTs where each cue descriptor is of dimension 240 except the last two which are of dimension 40. The value of χ is set to 40 in the experiments. The CDSF is constructed after separating the samples into two subsets. The first subset is used for constructing the CDST while the second set is used for learning the cue weights. All the processing is done on a processor with an Intel Xeon W3250 @2.67 GHz, 6 GB RAM.

4.6.1. Visual Cues - Indoors & Outdoors

In the first set of experiments, recognition rates of CDSF are compared with bubble descriptor (BuS). These experiments are done with the combined benchmark outdoors New College dataset [60] and indoors COLD dataset [16] - in an effort to make our dataset as extensive as possible. These two datasets were introduced in the previous Chapter. The scenes from both datasets can be seen in Figure 4.6.

Since the first dataset contains 7 places while the second dataset contains 3 different buildings with 14 places altogether, there are 21 different places. The robot learns these places by being given visual data associated with each different place – in early day (New College dataset) and cloudy (COLD dataset) viewing conditions. For convenience, both is referred to as cloudy illumination. Bubble descriptors (BuS) and CDSF are both constructed. It should be remarked while the two datasets contains different image formats (omni-directional vs panoramic), both bubble descriptors and CDSF are able to accommodate this with no problem. During learning, all places are associated with an ordered list of semantic cues of dimension $N_L = 2$. The first semantic cue is geographical description (building where the place is in, outdoors) with possible values New College, Freiburg Lab, Ljubljana Lab and Saarbrücken Lab. Hence its cardinality is $N_1 = 4$. The second semantic cue is appearance with $N_2 = 13$ possible values. Thus, each CDST consists of two levels as shown in Figure 4.7. While there is a total of 52 possible places, only 21 of these exist at the terminal nodes of the CDST.

In the recognition mode, the robot is given visual data that are taken at different times and robot bases, illumination conditions or both. The robot should classify the place associated with this data as one of previously learned places. Note that New College dataset has only two illumination conditions – day (cloudy) or evening (night). For recognition, one-against-one (*oao*) approach is used in BuS representation while 3-CDSF and 11-CDSF both use cue integration as explained in Section 4.5. The results are as shown in Table 4.1. It is observed that while BuS representation has good performance, the performance of 3-CDSF is very close or higher than that of BuS whereas that of 11-CDSF is somewhat lower than that of 3-CDSF. This suggests that

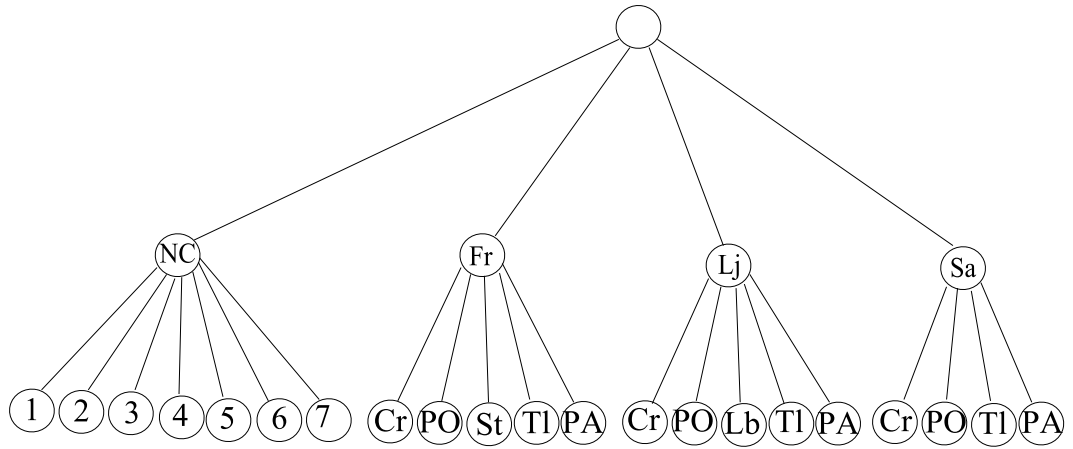


Figure 4.7. A CDSF. 1st Semantic cue: Geographical location: NC: New College, Fr: Freiburg Lab, Sa: Saarbrücken Lab, Lj: Ljubljana Lab; 2nd Semantic cue: Appearance: 7 places, Cr: Corridor, PO: Personal Office, Tl: Toilet, St: Stairs Area, Lb: Lab, PA: Printer Area.

Table 4.1. Recognition rates - BuS vs CDSF.

Test		Cloudy - Day	Sunny	Night	Average
New College	BuS	84.5	-	51.3	67.9
	3-CDSF	82.7	-	50.1	66.4
	11-CDSF	78.9	-	45.9	62.4
Freiburg	BuS	84.7	76	46.5	69.1
	3-CDSF	83.9	79.8	72.2	78.6
	11-CDSF	77.2	73.8	70.6	73.9
Ljubljana	BuS	75.7	72.6	62	70.1
	3-CDSF	74.3	75.1	69.3	72.9
	11-CDSF	73.2	72.9	66.6	70.9
Saarbrücken	BuS	78.3	-	71.3	74.8
	3-CDSF	64.6	-	64.4	64.5
	11-CDSF	56.7	-	61.1	58.9

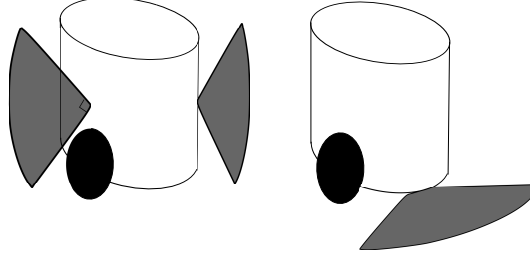


Figure 4.8. The locations of the 2D laser sensor in New College database. and in COLD database respectively.

increased structural integrity is advantageous in regards to recognition rates. It can be observed that in some places, i.e. New College, the average recognition rate for BuS is slightly better, while for indoors, 3-CDSF has a better performance with respect to BuS for Freiburg and Ljubljana. For Saarbrücken Labs, the BuS representation has a better performance probably due to combined complex structure of Saarbrücken Lab. At the same time, learning time with BuS is around 52.7 minutes while that of 3-CDSF is 43.3 minutes. Hence, there is a decrease of about 10 % that can be attributed to the decrease in SVM modeling time.

4.6.2. Adding 2D Laser Cue

In the next set of experiments, the same combined dataset is used and performance is compared when a new sensory cue becomes available. In this case, the new cue is two-dimensional laser data. In the New College dataset, the data is obtained from two laser sensors placed laterally on both sides of the robot while that in the COLD database is obtained from a laser range-sensor placed at the front of the robot as shown in Figure 4.8. As the laser data is, bubble surfaces are deformed only along the equator and its neighborhood depending on the pan range of the sensors. With this new sensory cue, BuS descriptor dimension increases to 2280 while the population of CDSF becomes $N_c = 4$ and $N_c = 12$ respectively. As the laser data from Ljubljana lab COLD is not available, the places associated with this lab are not considered and the CDSTs are constructed with 16 terminal nodes. With BuS representation, the

Table 4.2. Learning times with newly added laser cue.

	BuS	3-CDSF
Learning Time (s)	223	68

Table 4.3. Recognition rates with the added laser cue.

		Test			
		Cloudy - Day	Sunny	Night	Average
New College	BuS	92.6		67.9	80.25
	4-CDSF	93.7	-	78.7	86.20
	12-CDSF	91.9	-	76.9	84.40
Freiburg	BuS	91.1	82.9	79.6	84.53
	4-CDSF	94.7	92.2	75.6	87.50
	12-CDSF	91.5	87.6	74.4	84.50
Saarbrücken	BuS	70.4	-	66.2	68.30
	4-CDSF	53.4	-	69	61.20
	12-CDSF	51.8	-	69.9	60.85

SVM learning must be repeated from scratch. As shown in Table 4.2, the relearning time with BuS representation is approximately 223 seconds for 16 places. On the other hand, with CDSF, only the new CDST associated with the laser data needs to be constructed and cue weights $\kappa_{c,p^{l-1},l}$ need to be recomputed. Hence, relearning time is expected to be much less as compared to the BuS representation. Indeed this is the case with relearning time of 70 s for CDSF. Hence, there is a significant improvement in learning times - as the difference is on order of four times.

Of course, as expected, the recognition rates are improved with the added laser cue as shown in Table 4.3 with all the representations - except for the Saarbrücken Lab. With the added cue, the performance of 4-CDSF gets close to that of BuS while

Table 4.4. Recognition rates with missing color cues.

Test		Cloudy - Day	Sunny	Night	Average
New College	BuS	0	-	0	0.00
	3-CDSF	72.4	-	36	54.20
	11-CDSF	68.5	-	31.5	50.00
Freiburg	BuS	0	0	0	0.00
	3-CDSF	74.7	68.6	57.8	67.03
	11-CDSF	72.9	67	63.9	67.93
Ljubljana	BuS	73.2	72.1	69	71.43
	3-CDSF	71.5	71.2	66.1	69.60
	11-CDSF	72.9	72.2	66.1	70.40
Saarbrücken	BuS	0	-	0	0.00
	3-CDSF	49.3	-	53.2	51.25
	11-CDSF	32.5	-	44.4	38.45

surpassing in some cases. While the performance of cue descriptors with $N_c = 12$ cues is still lower, it is much closer to that of BuS. Surprisingly, for Saarbrücken Lab, the newly added laser data does not necessarily result in higher performance and in fact worsens it for the same illumination condition. This suggests that depending on content, additional cues may be sources of confusion. In summary, a CDSF with a small population size is preferable in regards to learning times - performance trade-off.

4.6.3. Missing Sensory Cues

In the next set of experiments, the same combined dataset is again used and the case of missing sensory cues is considered.

In recognition, it is assumed that color cues are not available (which may be due to low illumination or night-vision cameras). This implies that bubble surfaces associated with the color cues cannot be deformed. Both the BuS and CDSF representations

are built with undeformed bubble surfaces. Interestingly, there is dramatic performance difference between BuS and CDSF when color cues are missing as shown in Table 4.4. With BuS representation, all the places are found to be similar to Ljubljana lab and recognition performance deteriorates completely in other places. This can be contributed to the properties of SVM. Since SVM is modeled to compare the samples with all features available, a method that is using cue integration can be superior to it since the cue integration approach is more robust to mis-classifications of the missing features. In contrast, with CDSF, the recognition rates are reasonable and places are not confused with each other. As expected, overall recognition rates of CDSF decrease due to missing cue - with an average of 10-15%. Interestingly, the recognition rates of 11-CDSF also become closer to 3-CDSF - except for the Saarbrücken lab. This can be expected since with more cues, the effect of one missing cue will be relatively less. For Saarbrücken lab, while recognition rates with CDSF are still relatively high compared to BuS representation, the overall performance is relatively lower than the other places – in agreement with the previous results.

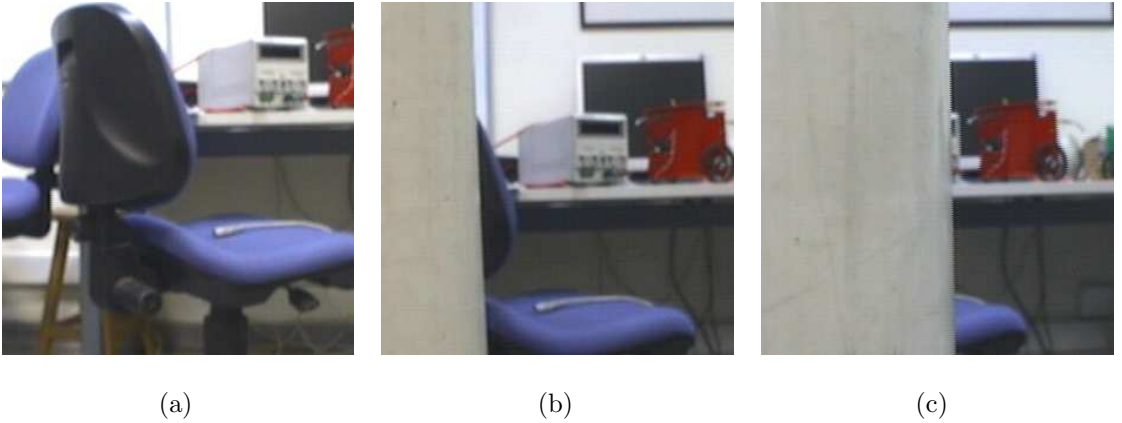


Figure 4.9. Sample occluded scene. (a) No occlusion. (b) 10 % Occlusion; (c) 20 % Occlusion.

4.6.4. Effect of Occlusion

In this set of experiments, the effect of occlusion on BuS and 3-CDSF is investigated. As the benchmark datasets do not contain scenes with occlusions, we use a dataset that is constructed with APES robot [150]. This dataset contains visual data from $N_p = 7$ different places within the Bogazici University North Campus as seen in Figure 1.1. This dataset is acquired from 7 sites including parking lot, depart-

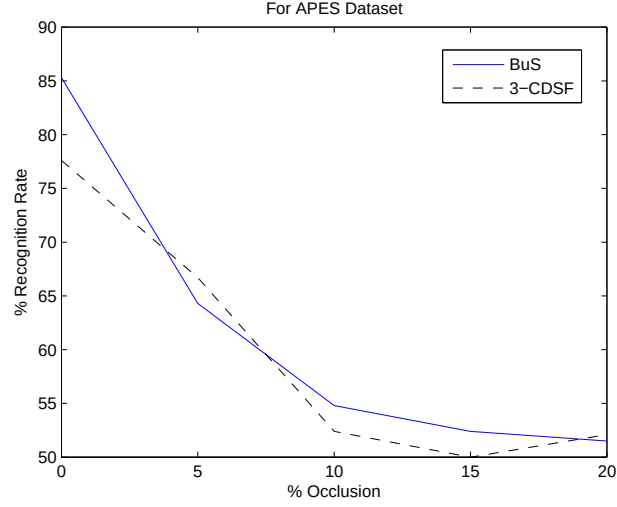


Figure 4.10. Recognition rates with varying occlusion.

ment office, corridor, science building, library, laboratory and science building entrance. Sample scenes from each site are as seen in Figure 1.1. The robot is trained at each place from $M_p = 9$ different base points as shown and it makes $N_s = 30$ fixations at each base point. Following, the robot is switched to operate in recognition mode. It then revisits each place again, but now the scenes are changed with occluding objects with varying amounts of occlusion as shown in Figure 4.9. The occlusion amount is controlled by the proximity of the new object placed in the scene. In contrast to previous set of experiments where one cue was completely missing, here all the sensory data is changed partially depending on the amount of occlusion. The recognition rates are shown in Figure 4.10. As expected, there is a gradual degradation of performance depending on the amount of occlusion in both of the cases. Interestingly, both follow a similar trend. For occlusions up to 20 % occlusion, the recognition rates of both BuS and 3-CDSF have a similar performance. It is observed that overall the performance BuS representation is slightly better - and hence these results are in agreement with those of Section 4.6.1. However, the performance differences between the 3-CDSF and BuS approaches vanish as the occlusion increases.

4.6.5. Summary

Experimental results with the combined two datasets suggest that CDSF representation has comparatively high recognition rates in large-scale environments with lower training times even with completely missing or partially changed (occlusions) data - in comparison with BuS representation. When only visual cues are used, the recognition rates are around 70 % for both representations with BuS being slightly better. Furthermore, it is observed that 3-CDSF performance is in general better than that of 11-CDSF. These two observations suggest that collecting similar features increases the recognition rates. With the addition of laser data, recognition rates increase as expected with average recognition rates of 85 %. Interestingly, 4-CDSF performance surpasses that of BuS representation in some cases. At the same time, learning times are 4 times faster than BuS representation. This is a significant improvement in the learning times. Hence, in contrast to recognition rates, using smaller sized cues the required time for training with a dramatic decrease in re-learning times. Then the robustness of CDSF under different conditions is investigated. In the first, one of the cues used in learning is missing completely during recognition phase. It is observed that while BuS representation fails completely, this is not the case for CDSF. Next, the two representations are compared under varying degrees of occlusion in the scene. Note that in this case, all the sensory features change accordingly. It is observed that performance degrades gradually as the amount of occlusion increases in both BuS and CDSF representations. As occlusion amount increases, the slight better performance of BuS diminishes and the recognition rates become about the same. In summary, there are several advantages of CDSF that extend the current state of the art as follows: i) Allow easy integration of different sensory modalities; ii) Shorter retraining and training times; iii) Robustness under missing or partially missing information of all sensory data.

4.7. Conclusion

This chapter presents cue descriptor semantic forests as a novel representation of different sensory and semantic cues for place recognition. This representation has two

aspects: Cue descriptors and cue descriptor semantic trees. Each sensory feature cue is encoded by a cue descriptor. Cue descriptors are supplemented by an ordered set of semantic cues and cue weights via cue descriptor semantic trees. As one cue descriptor semantic tree is associated with each sensory feature cue, the resulting representation is a cue descriptor semantic forest. The novelties of this representation are two-fold: First, as cue descriptors are constructed based on bubble space representation, as such, they incorporate any number of observations while preserving their relative S^2 geometry. Secondly, with the cue descriptor semantic tree structure, the integration of different feature and semantic cues becomes very simple regardless of their type or their number while enabling relative semantic expressiveness via cue weights. Comparative experiments on a benchmark dataset demonstrate that while learning times are decreased considerably with the introduction of new sensory cues, recognition rates are comparable to state-of-the art approaches in place recognition. Furthermore, recognition performance continues to be acceptable in case of missing or partially changed (occlusions) sensory feature cues.

5. UNSUPERVISED AND INCREMENTAL BUBBLE DESCRIPTOR TREE CONSTRUCTION IN TOPOLOGICAL MAPS

A mobile robot needs to recognize and classify the place it is in from a set of learned scenes for recognizing its environment [9, 105, 106]. If the place has not been observed before, then it should be inserted into the training set and the new data should be organized accordingly dynamically for obtaining an updated representation of the environment [6, 104]. Although recent appearance-based SLAM algorithms deal with large amount of data for detecting and including new locations into their maps [82], these methods lack the property of labeling a place with similar properties and finding representative classes of scenes.

The novel approach in this thesis can be described in two steps. First, it is assumed that the robot acquires visual data associated with a place that may be a previously learned place or a new one. The robot has to decide if it knows this place or not. In case it is a new place, it needs to update its knowledge of places accordingly. Secondly, we consider the organization of places learned using bubble descriptor semantic trees (BDSTs). The tree structure is incrementally modified as the robot decides it is in a new place – based on its current BDST. Using a tree-structure not only shortens the time necessary for classification [150], but it also allows updating the tree incrementally [151]. At each node of the tree except the leaf nodes, a model for classifying the corresponding node’s children is constructed.

Finally, bubble space representation is used for place representation since it can be used with different sensor modalities [114] and has a high recognition rate performance with a short computation time, therefore the capability of bubble space representation in a dynamic scene representation will be tested for the first time.

Table 5.1. Place Learning.

Time line \ Guidance	Supervised	Unsupervised
One-shot	[18, 105, 152] [19, 154, 155]	[21, 116, 153]
Incremental	[20, 156–158]	[159]

The rest of this chapter is organized as follows. First, a review of related work is given in Section 5.1. Then, we describe the incremental construction of BDSTs in Section 5.2. Experimental results are discussed in Section 5.4. The chapter concludes with a brief summary.

5.1. Related Literature

The related literature is presented from two perspectives. First, approaches in robotic place learning are discussed. This is followed by the discussion of tree construction methods for organizing data.

5.1.1. Place Learning

Robotic place learning approaches vary based on the provided external guidance (supervised vs unsupervised) and time line (one-shot vs incremental) as presented in Table 5.1. In a supervised approach, place labels are given externally whereas in unsupervised approaches, the robot needs to determine those itself. Secondly, the organization of learned places can be done at once or gradually over time.

Most approaches consider place learning in a supervised, one-shot manner. These are usually associated with appearance based topological mapping and include work such as [18, 19, 105, 152, 154, 155]. Common to all, the robot learns places after it has collected all the necessary data with semantic labels externally provided. Although the

Table 5.2. Tree Construction.

Time line \ Data Order	Dependent	Independent
One-shot	CLARANS [160]	AHC [161]
Incremental	BIRCH [162] AHC Mod. [163]	SLINK [151]

robot can then achieve high recognition performance, the robot needs guidance with respect to place labels and furthermore learning needs to be repeated from scratch when the robot goes to new places. More recent approaches have tried to tackle these issues. Some work have addressed unsupervised learning [21, 116, 153], however learning is done in a one-shot manner. On the other hand, there have been also incremental approaches as in [20, 156–158]; however these approaches require semantic labels to be externally provided. Interestingly, there are only few studies that propose both an unsupervised and incremental place learning. One such work is [159] that proposes an approach for the self-organization of a graph using neural-networks. However, it should be noted that the organization of the neural-network directly depends on sonar data and can only be used with a metric map.

5.1.2. Data Organization

Learning of places requires the organization of data. The most common approach is to use hierarchical clustering - in particular tree structures. This is a well studied area in pattern recognition and machine learning and various tree construction methods have been proposed for organization of large data sets. In all these approaches, the distance metric that measures the similarity between two classes plays a key role. Similarity measure determines the clustering performance. Three most widely used similarity measures are *single linkage*, *mean linkage* and *complete linkage*. In *single linkage*, the

maximum inter-cluster pairwise similarity is considered. *Mean linkage* considers the average inter-cluster pairwise similarity and *complete linkage* considers the minimum inter-cluster pairwise similarity.

These approaches vary depending on the effect of data order and time line of construction as shown in Table 5.2. A survey paper on the content-based indexing of large image datasets presents a comparative study of these methods with respect to new class detection and recognition rate performances [164]. It is observed that Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [162] is found to have the best performance. BIRCH is an hierarchical clustering method where clusters are detected in an incremental manner. However, the order of data is critical as changes in the data order can result in a different tree structure with possibly less optimal performance. Hence, even though it is advantageous when compared to non-incremental clustering methods like CLARANS [160], it is problematic if optimal data order cannot be guaranteed. While there are several methods that are independent of data order such as Agglomerative Hierarchical Clustering (AHC) [161], most are not incremental. Most incremental versions of these approaches such as the modified AHC [163]) do not support data order independency. There are various incremental tree construction methods. Incremental Tree Induction [165] uses incremental decision trees and spans the tree space to find a "best" tree structure according to a pre-defined metric between the nodes of the tree. Although this method is used in document categorization widely and successfully, it should be noted that it gives good performance when the features of the objects are conceptualized. However, in place recognition tasks, the features are usually numerical values. Therefore the success rate of using a decision tree in visual recognition tasks may be low. Fortunately, there is an algorithm that can be used with numerical features while being incremental and independent of data order - namely SLINK [151]. This is an approach which guarantees that the resulting tree has always the same structure regardless of the input order of classes and the clusters can be constructed incrementally.

5.2. Bubble Descriptor Semantic Trees

Suppose that the robot has visited places $\mathcal{P}$ with cardinality N_p . Initially, $\mathcal{P} = \emptyset$ with $N_p = 0$. As the robot navigates around and goes to new places, the set $\mathcal{P}$ expands accordingly with $N_p \gg 0$.

In each place, for each base point the robot generates a bubble descriptor $I(x_j(p))$. Please note that we will omit the second argument and use $I(x)$ for notational simplicity. Suppose that the robot looks around in each place $p \in \mathcal{P}$ from M_p different base points. We assume all the bubble descriptors are actually from the same place⁶ – albeit unknown to it. The average bubble descriptor is computed and taken as the representative of the generated bubble descriptors:

$$\bar{I}_p = \frac{1}{M_p} \sum_{j=1}^{M_p} I(x_j(p))$$

The robot needs to determine whether the place represented by $\bar{I}_p$ has been previously learned or new.

5.2.1. BDST Structure

Each BDST is constructed by cluster p with the associated representative bubble descriptor $\bar{I}_p$. A binary tree is constructed at the beginning. A dissimilarity matrix is used to construct the BDST. Euclidean distance is selected as the distance measure between the places p_i and p_k for the dissimilarity matrix, which can be explained mathematically as $d : N_p \times N_p \rightarrow R$, $d_{i,k} = \|\bar{I}_{p_i} - \bar{I}_{p_k}\|$ and $d(i, i) = 0$, for $i, k = 1, 2, \dots, N_p$ and $i \neq k$. We use *single linkage* similarity as our measure, since it creates a minimum spanning tree over data and has the ability to be modified incrementally when new clusters are available. Single linkage merges the nodes that are in minimum distance to each other until only one node remains. The nodes of the BDST is constituted by a

⁶This can be achieved via using place change or scene transition detection algorithms [166, 167].

vector $p = [p_1, p_2, \dots, p_{2N_p-1}]$. The leaf nodes of the BDST are the nodes without any child and they represent the places.

The bubble descriptors associated with a parent node p_i is indicated with $I(x(p_i))$ and it covers the bubble descriptors of its children such that $I(x(p_i)) = \begin{bmatrix} I(x(p_k)) \\ I(x(p_l)) \end{bmatrix}$ where p_k and p_l are the children of the node p_i .

It should also be noted that if the distance between any two inner nodes p_l and p_k is smaller than a threshold value (details are explained later), these nodes are merged to constitute a new node p_i and then p_l and p_k are destroyed. Therefore, the binary tree structure is not preserved and as a result each node can have more than two children. Hence, the labels of the bubble descriptors at the parent nodes can have more than two different values.

5.2.2. Recognition

To make a classification by using BDST, an SVM is trained at every node except leaf nodes. A one-against-all (*oaa*) approach is used for multi-classification. SVM is used for recognition since it is a standard approach that can continue to be used as sensory data and the number of places become much more extensive. As the leaf nodes are associated with one single label, it is not associated with a classifier.

At each node except the leaf nodes, the bubble descriptors $I(x(p_i))$ at that node are assigned with labels. These labels are associated with the child nodes to which these descriptors belong. All the bubble descriptors at the current node are used for constructing an SVM model.

5.3. Incremental BDST Update

5.3.1. Adding Nodes

SLINK [151] which is an hierarchical clustering method is used for tree update [168] and construction.

Let the set of all leaf nodes of the BDST be described as $p_i \in \mathcal{P}$, $i = 1, \dots, N_p$ as described previously and N_p is the number of places.

The dissimilarity function is a symmetric non-negative function $d : N_p \times N_p \rightarrow R$ as explained earlier. The tree can be characterized by a function $\Delta : [0, \infty) \rightarrow E(\mathcal{P})$, where $E(\mathcal{P})$ is the set of equivalence relations on $\mathcal{P}$, and Δ .

A tree on N_p places can be defined in $2N_p$ functions by using point representation. Let $P : 1, \dots, N_p \rightarrow 1, \dots, N_p$ and $L : 1, \dots, N_p \rightarrow 1, \dots, N_p \rightarrow [0, \infty)$ be pointer representations with the properties:

$$\begin{aligned} P(N_p) &= N_p \text{ and } L(N_p) = \infty \\ P(i) &> i \text{ \& } L(P(i)) > L(i) \text{ for } i < N_p \end{aligned} \tag{5.1}$$

Then, a pointer representation of the tree can be obtained where P, L defined are pointer representations for tree as shown in [151].

Now, assume a tree has been constructed with first N_p places. Then, the tree can be described with pointer representations P_{N_p} and L_{N_p} . When a new place $N_p + 1$ is inserted into the tree, the pointer representation P_{N_p+1} and L_{N_p+1} can be updated incrementally as follows. It should be noted this algorithm has a complexity of $O(N_p^2)$ for N_p classes.

```

 $P(N_p + 1) \Leftarrow N_p + 1$ 
 $L(N_p + 1) \Leftarrow \infty$ 
 $M(i) = d(i, N_p + 1)$  for  $i = 1, \dots, N_p$ 
for  $i$  increasing from 1 to  $N_p$  do
    if  $L(i) \geq M(i)$  then
         $M(P(i)) \Leftarrow \min\{M(P(i)), L(i)\}$ 
         $L(i) \Leftarrow M(i)$ 
         $P(i) \Leftarrow N_p + 1$ 
    else
         $M(P(i)) \Leftarrow \min\{M(P(i)), M(i)\}$ 
    end if
end for
for  $i$  increasing from 1 to  $N_p$  do
    if  $L(i) \geq L(P(i))$  then
         $P(i) \Leftarrow N_p + 1$ 
    end if
end for

```

Figure 5.1. SLINK Algorithm.

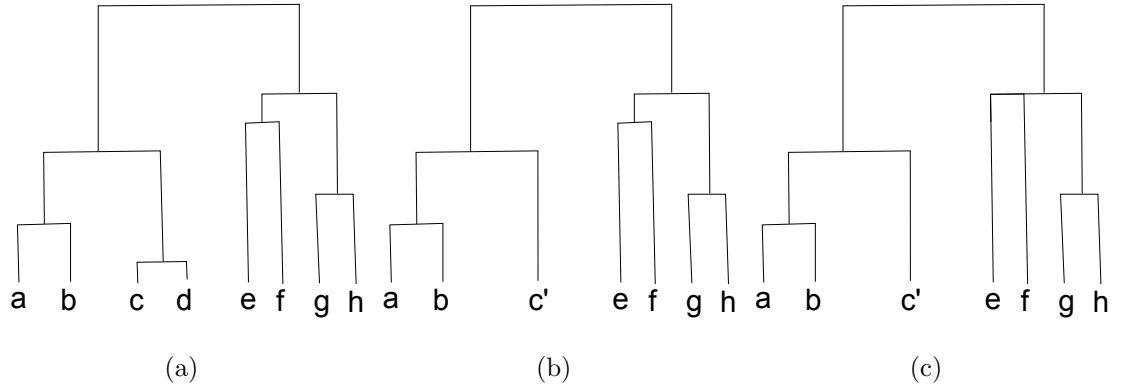


Figure 5.2. (a) The original tree. (b) After leaf node merging, nodes c and d merge into c'. (c) After inner-node merging.

It has been shown that this algorithm incrementally updates the tree and the final tree structure is independent of the input order of data [151].

5.3.2. Merging of Nodes

First, the leaf nodes that are close to each other under a threshold in the tree are combined together. Therefore, their parent node is replaced with the new combined node. By combining two leaf nodes together, the human evaluation of the tree is improved, furthermore a redundant classifier that will try to discriminate these two close nodes is eliminated, since these two leaf nodes are too similar to each other (e.g. the robot may return to the same location it has passed previously), so it is not necessary to make a discrimination between these two nodes. An illustration can be seen in Figure 5.2b.

Inner-nodes of the tree obtained in the previous section which are close to each other are combined together after the leaf nodes are combined. Merging closer inner-nodes together increases the human interpretation of the tree. For example the different locations of the same corridor can be merged in a single node. An illustration can be seen in Figure 5.2c. Merging of inner-nodes is made as follows:

- (i) Start from the inner-node p_{N_L} at the lowest level of the tree.
- (ii) If the parent p_{N_L-1} of the node p_{N_L} is at a distance less than τ_h , then connect

the right and left children of node p_{N_L} to its parent p_{N_L-1} and destroy the node p_{N_L} .

- (iii) Else mark p_{N_L} as visited.
- (iv) Go to the next unvisited inner node at the lowest level of the tree.
- (v) Finish the iteration when all of the inner-nodes are marked as visited.

5.3.3. Update of SVMs

After the tree is updated, the support vector machine (SVM) models are remodeled at the nodes whose structure has changed. Then, recognition is made by using these updated models. The details of updating.

If the members of a model at a node changes, then the the parameters of the corresponding separating hyperplane is also updated in a one-against-all SVM [148]. When a sample is introduced for recognition, classification starts from the root node. A *oaa* SVM classification determines the corresponding child node of the sample. The classification continues until a leaf node is reached.

5.4. Experiments

This Section presents experimental results to evaluate the incremental tree representation based on Bubble space. First, the performance of incremental tree construction approach based on bubble descriptors is tested. Both the recognition rates and the dependence on order of input data are investigated. In the second set of experiments, incremental and unsupervised tree construction is considered. The performance of an unsupervised incremental tree is compared with the results obtained in the first part of the experiments. In the last set of experiments, the resulting graphs from two sets of experiments are compared.

The experiments are similar to those that were done in recent related work [8, 19, 95] and previous chapters where the task is to recognize the current place based on previous learning. The learning process has been completed by using the data obtained

in Cloudy/Day illumination condition for all of the databases. The test data differs either in robot pose, viewing illumination, time or any combination of from the learning data.

The bubble surfaces are formed with sensory observations from $N_s = 40$ different viewing directions with parameters $\rho_0 = 1$, $\rho_{min} = 0.5$, $\rho_{max} = 1.5$ and $\epsilon = 2^\circ$.

These viewing directions are determined automatically via looking around in a free viewing mode as described in detail in [100] and Chapter 2.

The corresponding bubble descriptors are constructed with $H_1 = H_2 = 6$ number of harmonics, however only the first 40 rotational invariants are used since the remaining are observed to be non-discriminating.

There are again 56 features as explained in the Appendix C.

5.4.1. Incremental Tree Construction

In the first set of experiments, images that are labeled in a supervised manner according to the place they are captured from are used. The importance of input order is tested and the recognition rates of the resulting merged trees are evaluated. For test, the input data of the clusters to the system is given in a different order. The resulting tree was always the same regardless of the different orders. The changes resulted in 24 different input orders (e.g. order Fr-Lj-Sa-NC in one trial and order Fr-Sa-Lj-NC in another trial).

The corresponding place labels given in a supervised manner is shown in Figure 5.3 for the input order Fr-Sa-Lj-NC. As it can be seen there is a total of 21 places.

The resulting trees are shown in Figure 5.4. It can be observed that if the places are similar to each other, then their distance in the tree is smaller. Furthermore the places from the same location are in the same branch of the tree at most of the time.

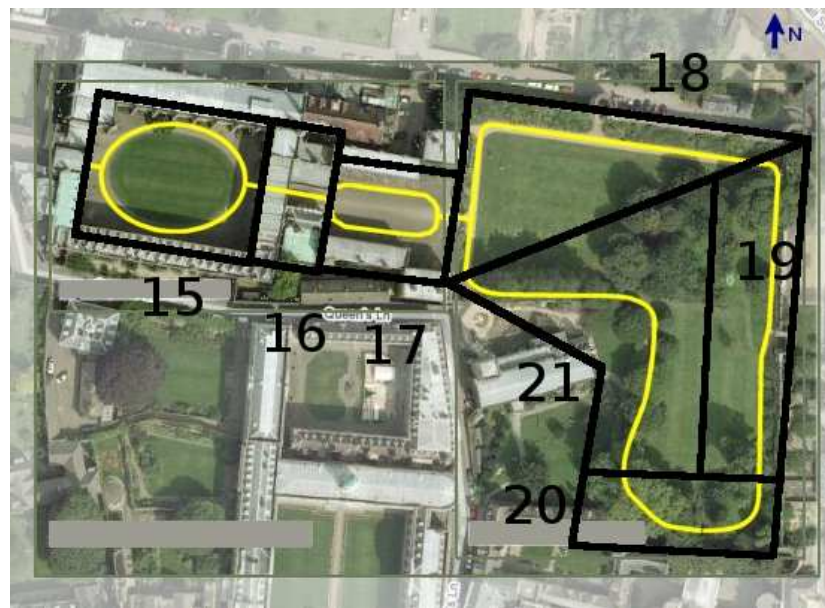
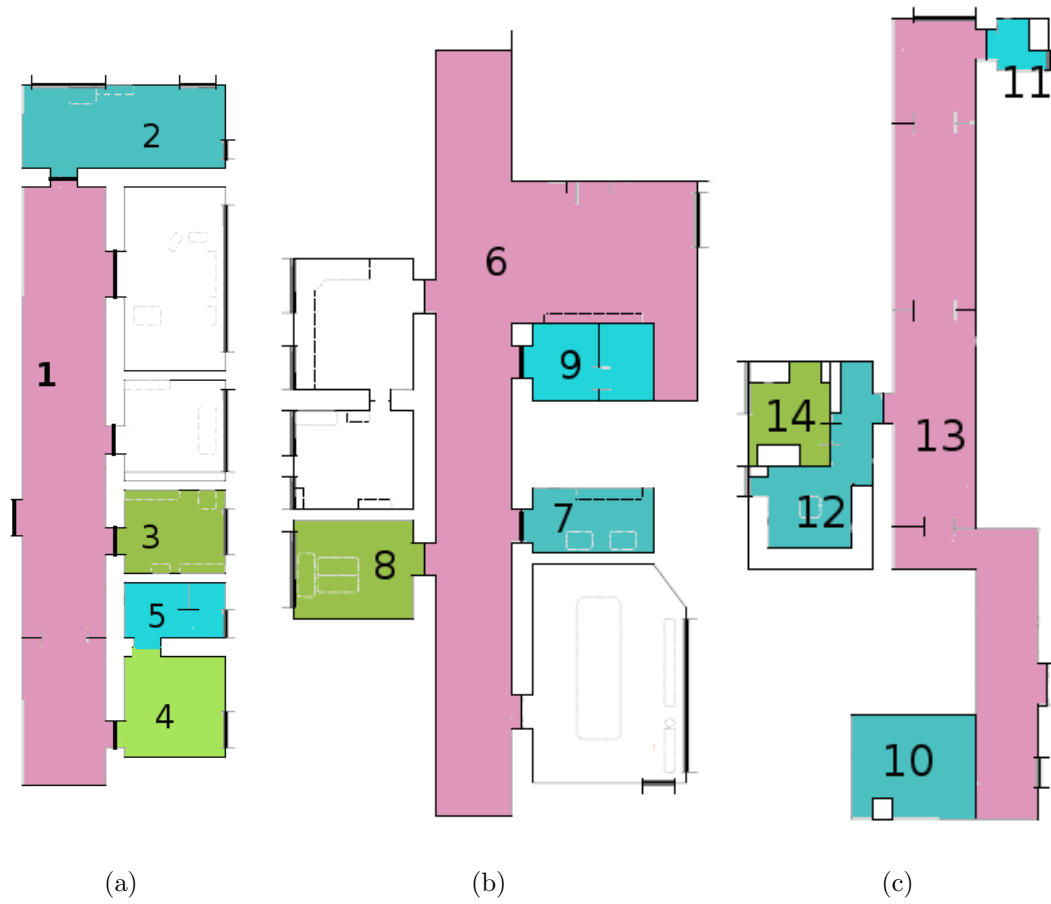


Figure 5.3. Labels from: (a) Freiburg Lab, (b) Saarbrücken, (c) Ljubljana; (d) New College.

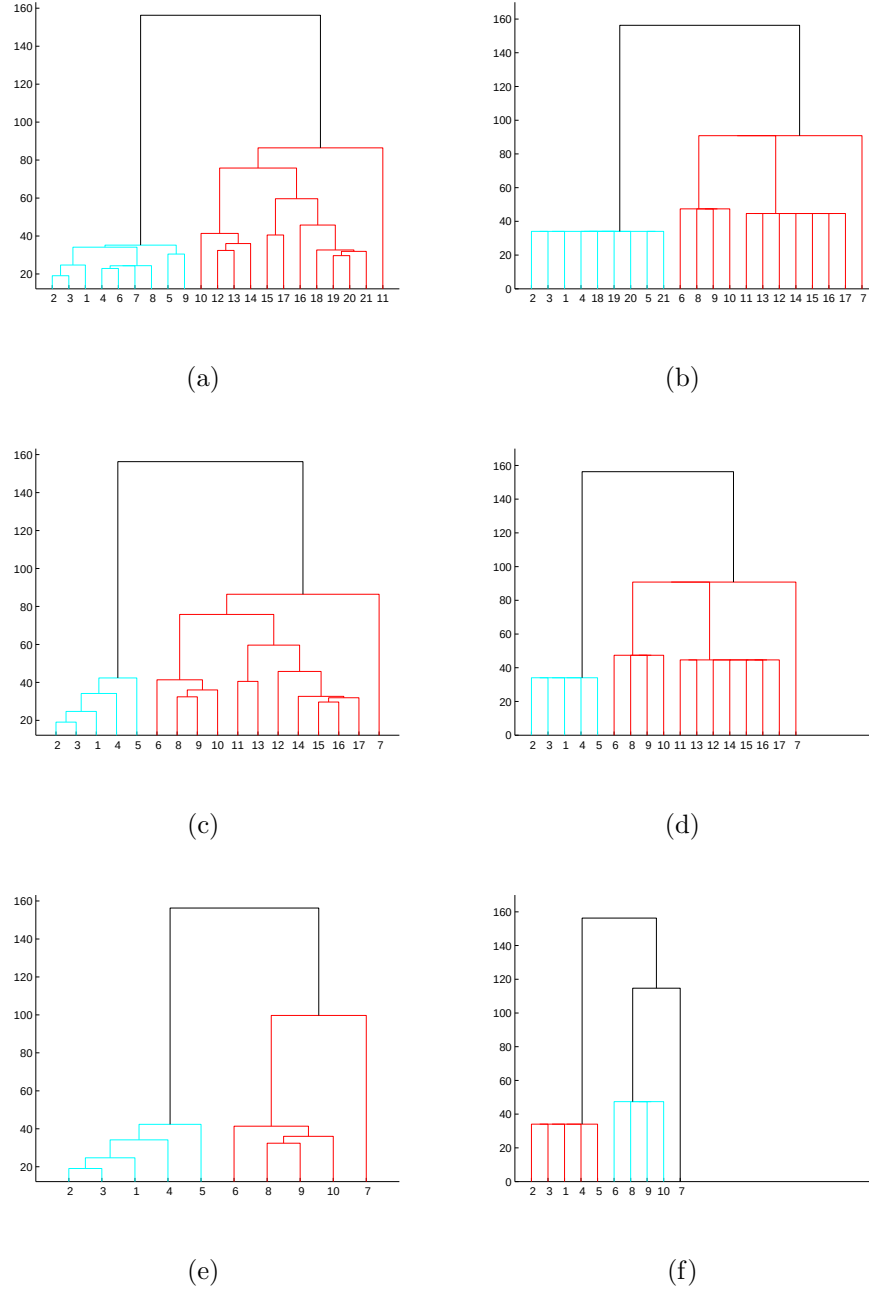


Figure 5.4. Incremental trees constructed with supervised labels. (a), (c) and (e) Trees before inner nodes are merged. (b), (d) and (f) Trees after inner nodes are merged. (a) and (b) Trees after all labels are merged, (c) and (d) Fr, Lj and NC are merged, (e) and (f) Fr and Lj are merged.

For New College database, the images taken from the park are labeled as close to each other which is expected as the images from the park has similar visual features when compared to the images containing human made buildings. The rooms of Fr, Sa and Lj are also labeled together with some exceptions. For example, the washrooms of the Fr and Sa labs are grouped as more similar to each other. This is due to similar visual features of the two places. This also shows that even though the properties of the cameras are different, the resulting bubble space for the places are comparable with each other. Another exception is the washroom of Lj lab, which is at a distant location to the Lj lab in the tree. When the merged tree is investigated, it is seen that the places from the same lab/environment are grouped together if an appropriate threshold is selected. This does not only ease the human interpretation of the environment, but it also reduces the number of models necessary to classify an environment into a reasonable number. It should be reminded that we use *oaa* SVM for multi-class classification at inner nodes.

To investigate the effect of the input order on the tree, three different snapshots have been given in Figure 5.3. At the bottom, initially only Fr and Lj are combined. As it can be seen, the structure of the Ljubljana is same with its final structure, while the structure of Fr changes when Sa is inserted, due to similar visual properties of Saarbrücken to Freiburg lab. Although the final structure of Fr lab is different from its initial structure in the tree, the final structure does not change with a different input order. Furthermore, in the merged trees, as it can be seen the semantic levels are merged together. For example the places from the two indoor labs are grouped together while the outdoor images from New College is grouped together.

When the recognition rates for the trees are investigated in Table 5.3, it is observed that recognition rates under cloudy illumination conditions are better, since the learning is also made under cloudy illumination. The recognition rates are comparable with the state-of-art methods [19]. It should be noted that the tree structure can be updated in an unsupervised manner every time when a new class is available in optimal time. As a result of the first set of experiments, it can be said that the recognition rate of the current method is sufficient when compared with the results obtained in Chapter

Table 5.3. Recognition Rates.

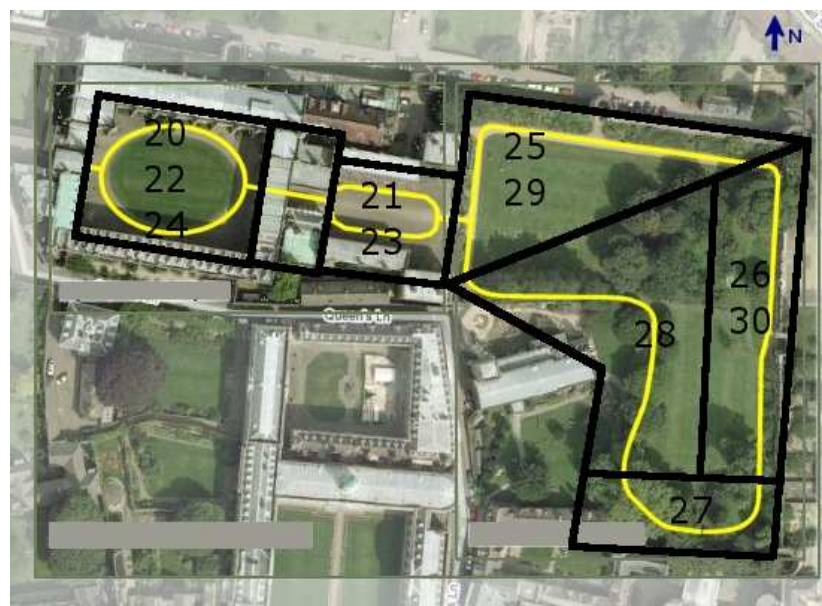
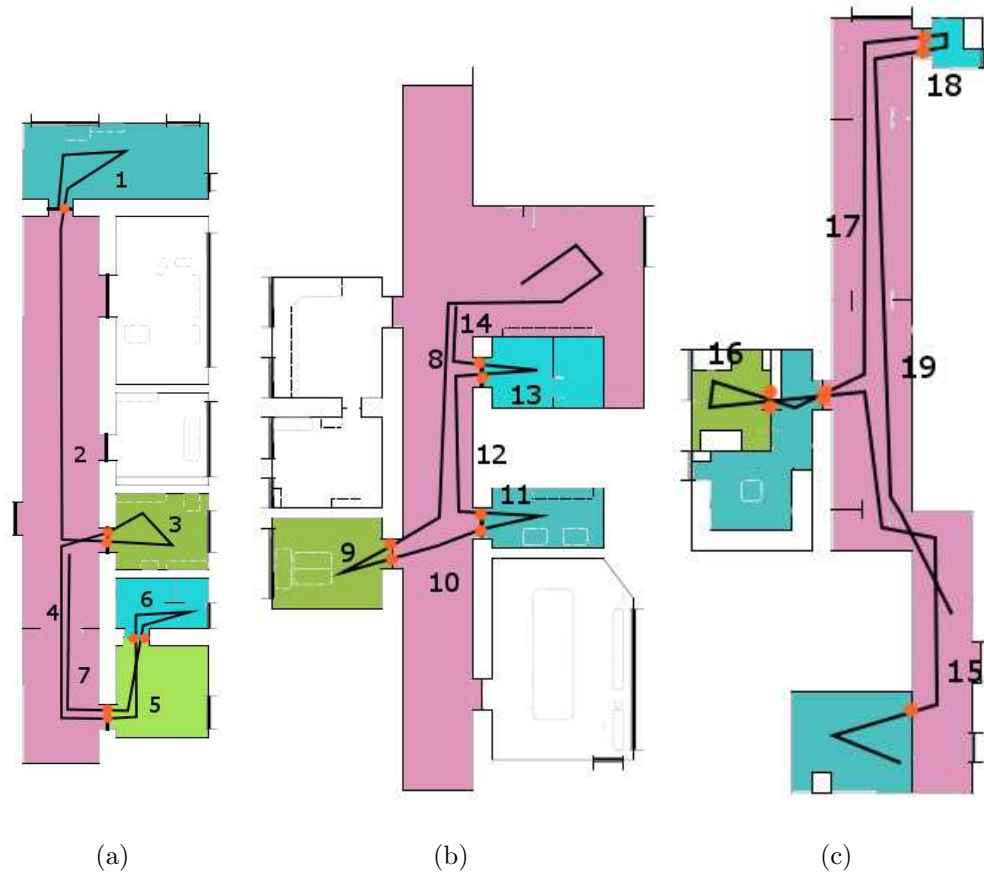
	Fr	Sa	Lj	Nc
Cl	84.3	77.8	73.1	83.5
Su	76.3	-	72.1	-
Ni	46.5	72.2	69	51.5

4. The tree structure is plausible as it combines places from the same environment in the same node and when the input order changes it can output the same final tree.

5.4.2. Incremental & Unsupervised Tree Construction

In the next set of experiments, the scene changes are determined in an unsupervised manner. Again the same parameters with the previous set of experiments are used on the same dataset. The corresponding place labels detected in an unsupervised manner is shown in Figure 5.5 for the input order Fr-Sa-Lj-NC. As it can be seen 30 place labels are detected. The number of labels are higher than the supervised labeling, since some of the places are large and they are divided into smaller labels by the scene change detection system, or some of the places have more than one label as there exists a scene change detection when the robots enters the associated place. Furthermore, some of the places don't have labels, since the number of training samples is not sufficient for constituting a node representation for the corresponding place.

The resulting tree is shown in Figure 5.6. Again, when the tree is investigated, it can be seen that similar places are grouped close to each other. The places from the same or similar locations are in the same branch of the tree. Again, similar to the supervised labeling case, in New College database, the places from the park are labeled close to each other. When the merged tree is investigated, it is observed that the places from the same lab/environment are grouped together if an appropriate threshold is selected. Furthermore, the New College database is divided into two parts. One of the parts contain the images from the urban part of the data set which includes buildings,



(d)

Figure 5.5. (a) Freiburg Lab, (b) Saarbrücken, (c) Ljubljana; (d) New College.

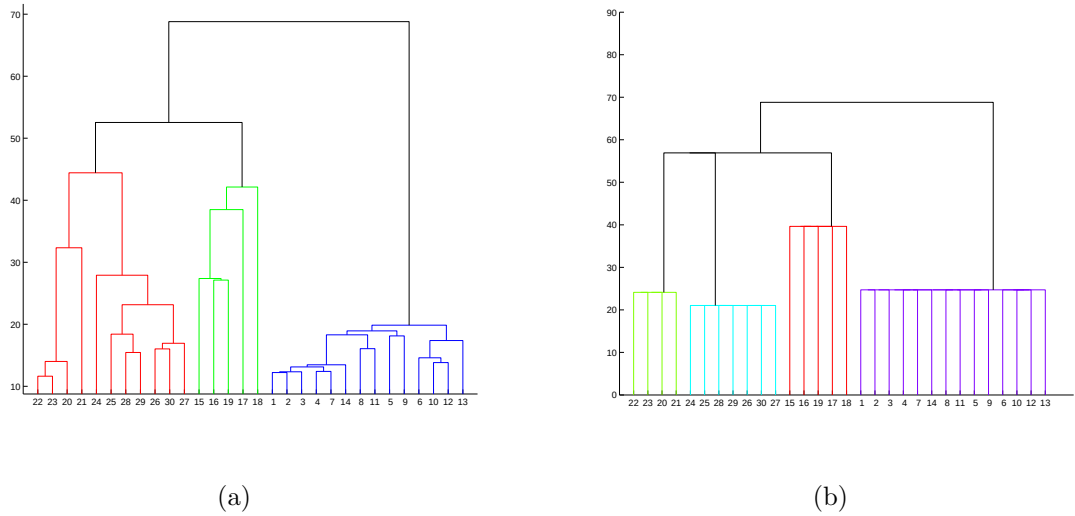


Figure 5.6. (a) Incremental tree constructed with unsupervised samples. (b) Incremental tree constructed with unsupervised samples after inner nodes are merged.

Table 5.4. Recognition Rates for Unsupervised Labels.

	Fr	Sa	Lj	Nc
Cl	81.1	77.0	70.1	81.1

and the second one includes images from the park side of the data set.

The recognition rate for unsupervised images can be seen in Table 5.4. To compare the recognition rates with the incremental tree constructed in a supervised manner, only one illumination condition is considered. The recognition rates are slightly worse than the results obtained when an unsupervised labeling is applied. It should be noted that as a new scene is detected, it can be inserted into the tree automatically. As the number of images increase, it would be a difficult task to label all of the images. The performance of the incremental tree with labels determined in an unsupervised manner is sufficient to be used in a place recognition system. If the scene change detection system performs different, i.e. closer to the behavior of a human, the results from the first set of experiments show that the performance can be improved.

5.4.3. Comparison of Supervised vs Unsupervised Tree Construction

In the last part of experiments, the dissimilarity of the trees that were constructed by using both supervised and unsupervised labeling is compared on the same dataset with the previous two experiments with same parameters. In Table 5.5, the labels given in a supervised manner can be seen on the left-most column. In the middle column, the labels given in an unsupervised manner by the scene change detection system is shown. In the right-most column, the description of the corresponding labels are given.

To compare the trees constructed by two different methods, the dissimilarity matrices are used. If a supervised label is represented by more than one label in the unsupervised manner, the unsupervised labels are merged as one super-node. The distance of this new super-node to the other nodes is taken as the minimum distance between the members of the super-node and the corresponding other nodes. Therefore, the size of the dissimilarity matrix of the incremental tree constructed in an unsupervised manner is reduced to the size of the supervised tree which is $N_p \times N_p$. Furthermore, some of the supervised labels have no associated nodes in the unsupervised labels. The distances between these nodes and others are extracted from supervised dissimilarity matrix. The size of the new dissimilarity matrix reduces to $N_{p'} \times N_{p'}$ where $N_{p'}$ is the number of rows and columns of the modified dissimilarity matrix after the rows and columns corresponding to nodes that exist in supervised tree, but not in unsupervised tree are removed. The similarity matrix S between the two dissimilarity matrices can be found as follows:

$$s_{ij} = \frac{\|d_{(ij)}^u - d_{(ij)}^s\|}{\|d_{(ij)}^u + d_{(ij)}^s\|} \quad (5.2)$$

where d_{ij}^u and d_{ij}^s are the i th and j th elements of the dissimilarity matrices for the unsupervised and supervised trees respectively. Then, a similarity score from the similarity

Table 5.5. Label Descriptions.

Supervised Label	UnSupervised Labels	Description
1	2 4 7	Fr Corridor
2	1	Fr Printer Area
3	3	Fr Office
4	5	Fr Stairs
5	6	Fr Bathroom
6	8 10 12 14	Sa Corridor
7	11	Sa Printer Area
8	9	Sa Office
9	13	Sa Bathroom
10	-	Lj Office
11	18	Lj Bathroom
12	-	Lj LAB
13	15 17 19	Lj Corridor
14	16	Lj Office
15	20 22 24	NC Buildings
16	-	NC Tunnel
17	21 23	NC Park Entrance
18	25 29	NC Park
19	26 30	NC Park and Wall
20	27	NC Trees
21	28	NC Park Exit

matrix can be obtained by:

$$sc = \frac{2}{N_p^2} \sum_i^{N'_p} \sum_j^{N'_p} s_{ij} \quad (5.3)$$

The similarity score is expected to be 1 if the two matrices are completely different, and 0 if they are same. When the dissimilarity matrices are compared, the similarity score is computed to be 0,25, which is smaller than 0,5. Therefore, it can be concluded that these two structure of the trees are similar to each other. It should also be noted that, the trees before merging the inner nodes are used.

5.4.4. Summary

Experimental results made with the datasets show that the proposed approach can construct the same tree structure regardless of the order of input data. The recognition rate of the corresponding tree is similar to the results obtained by previous methods for the same dataset. The incremental nature of the tree makes it possible to insert new nodes into the tree. When the tree structure is investigated, it can be seen that the similar places are grouped together, which makes it easier for humans to interpret. When the labels of the places are not given to the system, and the labels are determined in an unsupervised manner, the results show that the scene change detection can construct a tree similar to the one obtained in a supervised manner. Again, visually similar places are grouped together. The decrease in the recognition rates suggest that a scene change detection method which results in labels to the one in the first part of the experiments can increase the performance. As a last step, the similarities of the two constructed trees have been investigated by using a similarity score. The result suggests that the trees are similar to each other quantitatively. In summary, the advantages of incremental and unsupervised bubble descriptors tree construction can be explained as follows: i) Labeling of large collection

of images from different places in an unsupervised manner with high performance rates,
 ii) Construction of trees incrementally making it possible to insert new nodes into the tree in the optimal time.

5.5. Conclusion

This chapter introduces incremental and unsupervised bubble descriptor tree construction. There are two important aspects in tree construction for place recognition, one of them is incremental update of the tree and the second one is the construction of the tree with unlabeled images from the environment. Each place is represented by a set of bubble descriptors, which corresponds to a leaf node in the tree. The tree is constructed by using a single-linkage algorithm. There are two novelties in this study: First, an incremental method for tree construction is used with bubble descriptors for place recognition, as such, any node can be inserted into the tree in optimal time and the final structure is independent of the input data order. Secondly, the trees are constructed by using unlabeled bubble descriptors which are detected with a scene change algorithm. Reconstructing the classification structure of the environment and labeling of huge data sets in a supervised manner and requires higher computation time and labor. On the other hand, one of the main concerns of incremental and unsupervised tree is the dependence on the order of input data and recognition rate performance. These two concerns are tested by using a combined data set including both indoors and outdoors images. The results demonstrate that the recognition rates are high and as the scene change detection algorithm approaches the performance of a human in unsupervised labeling of places, the recognition rates increase. Furthermore, it has also been shown experimentally that the incremental tree structure achieves the same structure independent of the input data order. Thus, incremental and unsupervised bubble descriptor tree can be used for place recognition where the task is to label and construct a structure for place recognition.

6. CONCLUSION

This thesis is concerned with the robotic place recognition problem. From the robot's perspective, the goal of place recognition is to determine its whereabouts - given its knowledge of places it has visited previously. This problem has been broken into four stages: First, the problem of looking around in a place and collecting sensory data is considered. For this, a model of attentive visual behaviours based on a family of artificial potential functions (APF) is developed so that the robot is able to consider the interesting points in the scene. Hence, instead of processing the whole scene, only salient features are gathered. In this manner, The contribution of this approach is that a variety of different features including corners, color and motion can be selected in a pre-attentive stage in mobile robots by using a closed loop approach based on APF. The method is tested under varying conditions and the results are compared both with an other attentive vision model and with the information obtained from human. The advantage of this approach is that a rich set of primitive features can be used for detecting salient point during pre-attentive stage on a real-time robot. The attention model currently considers only visual data feature set, this can be extended to include 2D or 3D features.

Next, place representation is considered. For this, a topological model referred to as bubble space is introduced. Sensory features associated with all the interesting points are encoded as bubble surfaces. Bubble surfaces enable the representation of all visual features that is implicitly dependent on robot pose while preserving their local S^2 -geometry. For compact representation, bubble surfaces are transformed into bubble descriptors that are rotationally invariant with respect to heading changes while being computable in an incremental manner as each new set of visual observations is made. Bubble descriptors are used for supervised place learning and recognition with support vector machines. BuS representation enables high recognition rates with low standard deviation under strict testing in stable and varying illumination conditions. Another advantage is its codebook extraction time per frame which is significantly lower. In this thesis, the representation is only used for place recognition, however the results

suggest that it can be used in various real-time robotic tasks such as loop closure and map building.

Following, the problem of integration of sensory and semantic cues is considered and a representation referred to as cue descriptor semantic forests is proposed. The integration of different feature and semantic cues becomes very simple regardless of their type or their number while allowing relative and possibly partial semantic expressiveness via cue weights. Comparative experiments on a benchmark dataset demonstrate that while learning times are decreased considerably with the introduction of new sensory cues, recognition rates are comparable to state-of-the-art approaches in place recognition. Besides the advantage of reduced learning times, the integration method can be investigated with other techniques for obtaining the cue weights to see the effect of the cue weights determination technique on recognition results.

Finally, the problem of unsupervised place learning is considered. A novel approach for the incremental and autonomous construction of bubble descriptor semantic trees is proposed. An incremental method for tree construction is used with bubble descriptors for place recognition, as such, any node can be inserted into the tree in optimal time and the final structure is independent of the input data order. Secondly, the trees are constructed by using unlabeled bubble descriptors which are detected with a scene change algorithm. Comparative experiments on benchmark datasets as well as real-time robotic experiments indicate high recognition rates. The distance metric used in tree construction is currently an Euclidean metric between the bubble descriptors of the places, the effects of using other metrics on both tree structure and recognition rates can be considered.

In conclusion, place recognition problem has been considered in this thesis and the following advantages have been observed: i) a rich set of primitive features can be used for detecting salient point during pre-attentive stage on a real-time robot, ii) high recognition rates with low standard deviation under strict testing in stable and varying illumination conditions with BuS representation and significant low codebook extraction time per frame, iii) simple integration of different feature and semantic cues

regardless of their type or their number and high recognition rates after integration, iv) incremental and autonomous construction of bubble descriptor semantic trees with high recognition results. As part of future work, the following points can be considered: i) attention model modification to include 2D or 3D features, ii) Using Bubble Space representation in various real-time robotic tasks such as loop closure and map building, iii) investigated of other techniques for obtaining the optimal cue weights to see the effect of the cue weights determination technique on recognition results, iv) using another metric in tree construction between the bubble descriptors of different places.

APPENDIX A: APES G3 - ATTENTIVELY PERCEIVING SYSTEM

APES is a differential wheel autonomous robot with an attentive vision that has been developed in ISL at Bogazici University. It can work with two cameras with pan-tilt motion capability. The first generation was developed by Ç. Soyer [14]. The last generation of this robot is APES G3 as shown in Figure A.1 with improved locomotion and processing capabilities. The robot can be used both manually by connecting remotely or in automated mode.

APES is designed to be able to make different visual behaviors by using its active vision system and its locomotion. The visual behaviors that can be achieved by the robot includes free viewing, search, tracking by using its body and remain fixated. Additionally, it can switch between these behaviors automatically [100]. In the following sections, its technical, mechanical, electrical and software details will be given.

A.1. Hardware

The technical details can be seen in Table A.1. It should be noted that the power consumption for main board is the maximum possible consumption and the batteries can be changed with LiPo 22.2 V 6S batteries as well.

A.1.1. Mechanical Details

The modifications that has been carried on the mechanical aspects of APES can be listed as follows:

- Active vision system re-designed and manufactured with a worm gear system (Figure A.2).

Table A.1. Technical Details of APES.

Dimension (cm)	Height	77
	Diameter	40
Max Linear Velocity (m/s)		1,40
Mass (kg)		26,90
Number of motors		4
Motor Type	DC	
Degree of freedom		2
Drive		Differential
Power	Lead-Acid	12V 7Ah x2
Camera Module	Vertical Velocity (rad/s)	28,89
	Horizontal Velocity (rad/s)	38,44
	DoF	2 (Pan/Tilt)
Computer	Processor	1.6 GHz DualCore
	Ram	1024 MB
	FrameGrabber	USB
	HDD	8 GB
	USB	x4
	Serial	x1
	Parallel	-
	LAN	x1
	Power Consumption	26 W
Camera		Colorful (PAL)
		x2 Analog

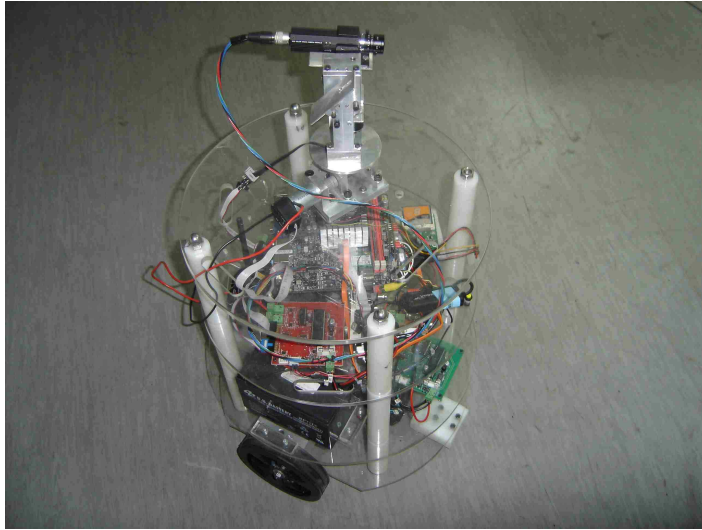


Figure A.1. A general view of the APES G3.

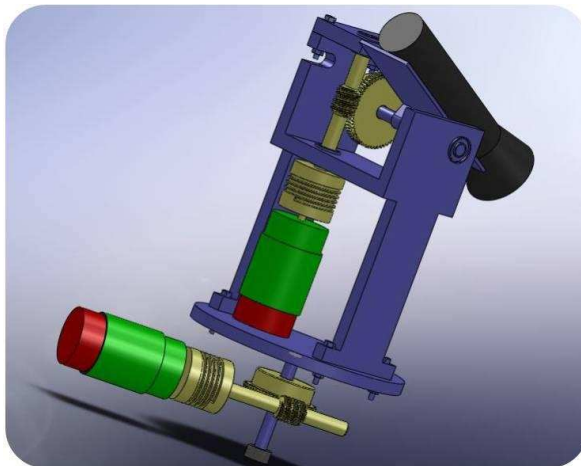


Figure A.2. A 3D model of the active vision system with two worm gears on it.

- A new layer has been added to its body for carrying two batteries, one of them for actuating motors via motor cards, and the other one for camera and computer systems.
- Relocating the computer layer for the Mini-ITX form main board and new motor control cards.

A.1.2. Electrical Details

The wiring diagram of APES can be seen in Figure A.3.

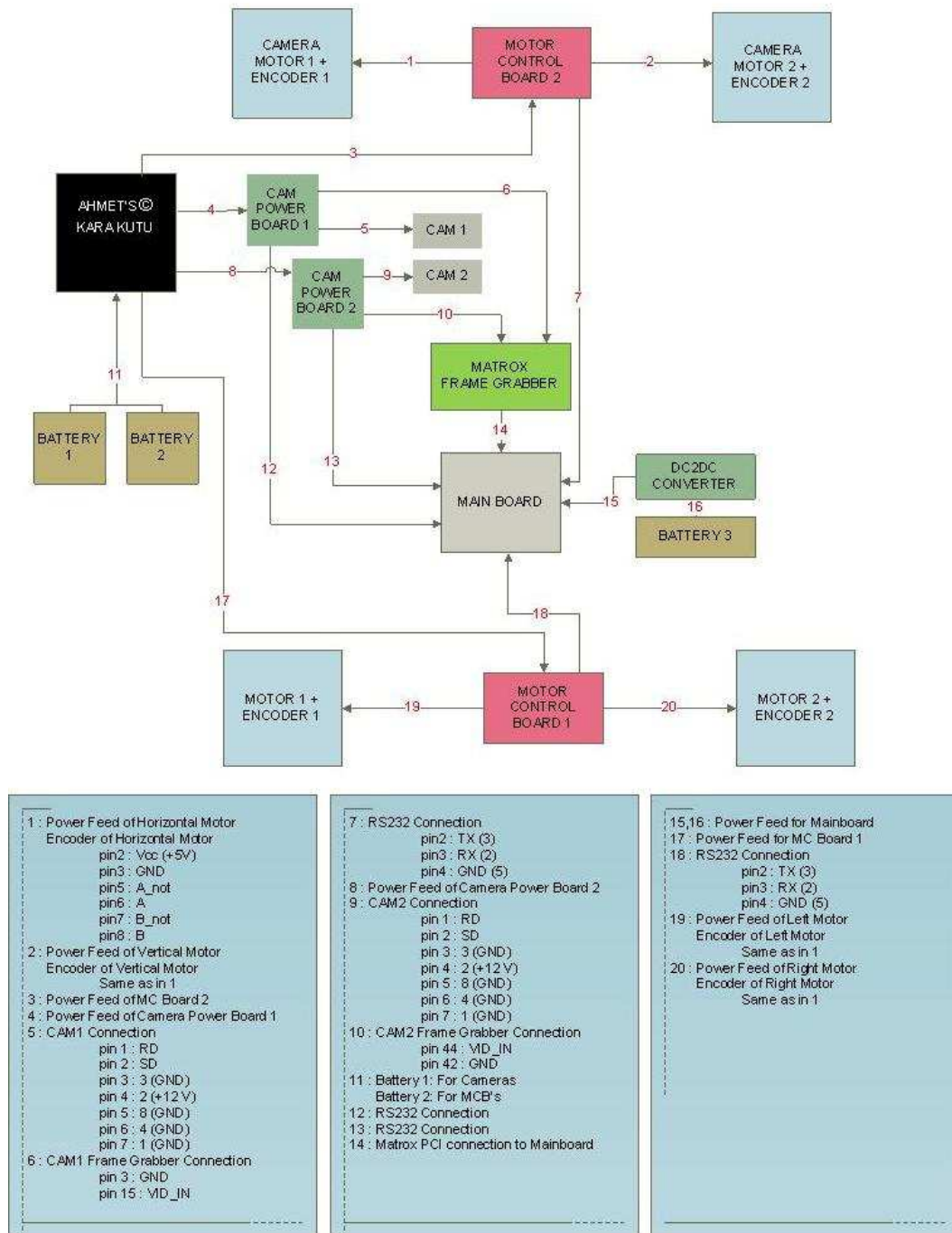


Figure A.3. Wiring diagram of APES.

A.1.3. Electronic Details

The electronic devices on APES can be listed as:

- two batteries (can be extended to four batteries),
- two cameras (SONY XC555) to be used in stereo,
- two camera power boards to regulate the voltages of the cameras,
- two USB based frame grabbers (previous frame grabber Matrox is replaced with HomeVideo 4 EyeGrab),
- one DC2DC converter as a power supply for the main board
- one mini-ITX form main board (Zotac)
- one Compact Flash Card as hard disk (8 GB)
- two motor control boards developed at ISL.

A.2. Software

The robot has Windows XP operating system on it. The following details are given for the Windows XP Operating System.

A.2.1. Required Software

Visual Studio 2005 should be installed on the system. OpenCV 1.x and IPP 6.x libraries should also be installed by using the executable setup files. For image capturing, the driver for 4eyeGrab, which is an analog to digital video converter, can be installed by using executable setup files.

A.2.2. How To Use Software

The software for APES is a Visual Studio 2005 project named 'apesExp' under the 'Documents' directory. The program on the robot enables it to store images and the corresponding motor encoder and angle values into a file while making a visual search based on a pre-attentive approach. The images are stored in the folder 'apes0'.

By changing the `defs.txt` file in the project folder, the robot can perform different behaviors. Some of the parameters that can be changed in this file can be listed as follows:

- The size and visual field angle of the captured image (must be given),
- Number of previous fixated points to be kept in memory,
- The parameters to change the visual behavior of the robot as explained in [100],
- Maximum and minimum speeds that the pan-tilt cameras can reach,
- PID parameters for locomotion motors and pan-tilt motors, both for speed and for position,
- Options for storing images, saliency map surfaces at every given number of fixations,
- If necessary, limits for the motion of the camera head (for cases where the robot is required to fixate on a certain region),
- A variable called as `APPROACH` for changing the robot's behavior according to the selected value. For example, a value of 10 will make the robot move according to APF camera dynamics, if it is set to 4, the robot can be controlled manually by using keys `z,x,c,s` and arrow keys to control the pan and tilt movements. of the camera. `'t'` will take one shot image while `'g'` will cause the system to grab images continuously. To stop grabbing process, `'b'` key should be used. You command APES through a Terminal window, however you can see the image grabbed by the camera in real-time as long as the camera has power and is connected to the robot. A value of 3 will make it to store a mosaic image set of its environment.

Finally, the robot can be used to recognize its environment by using the set of images acquired with an attentive based approach, provided that the models of the places are computed and given to the system prior to recognition process.

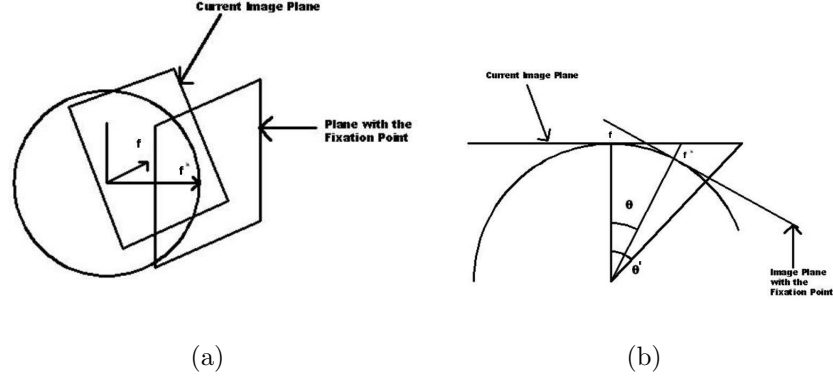


Figure A.4. (a) Image plane orientations; (b) Image transformation.

A.3. Camera Head Dynamics

The robot moves its camera head with velocity control. The dynamical model of the robot is defined as follows. Once the camera starts moving away from the current fixation point f to f' , the image plane changes as shown in Figure A.4a. Hence, the saliency map gradients need to be computed at the corresponding image plane as shown in Figure A.4b. First, recalling that $f = \begin{bmatrix} f_1 & f_2 \end{bmatrix}^T$, let $f_1 = \theta$ and $f_2 = \phi$ be the pan and tilt angles of the camera associated with the current image plane as shown in Figure A.5a. These values are obtained from its motor encoders. Define f' similarly. Let $y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$ denote the intersection of f with the current image plane. Define y' similarly – using f' . Using trigonometric identities, it is easy to show that $|y_1 - y'_1| \propto \frac{W_I \tan(\theta)}{2 \tan(\theta')}$ where W_I denotes the image plane width. The value $|y_2 - y'_2|$ can be computed in analogous manner. Hence, the point corresponding to y' on the current image plane can be computed.

In order to move the camera, the gradient vector which is of Cartesian nature needs to be transformed to the angular velocities. The arc we want to move along ϕ direction is shown in Figure A.5b. A similar case holds for θ . Let δy be equal to the gradient of the saliency map as given by Equation 2.2

$$\delta y \triangleq \begin{bmatrix} \delta y_1 \\ \delta y_2 \end{bmatrix} = u_k(t, x)$$

As δy is in Cartesian coordinates, it needs to be converted to angular coordinates being input to the motors. This is easily achieved via the following transformations as the angular velocity components proportional to the Cartesian velocity components as:

$$\delta\phi = \frac{1}{r}\delta y_1 \quad \delta\theta = \frac{1}{r}\delta y_2$$

where r is the focal length.

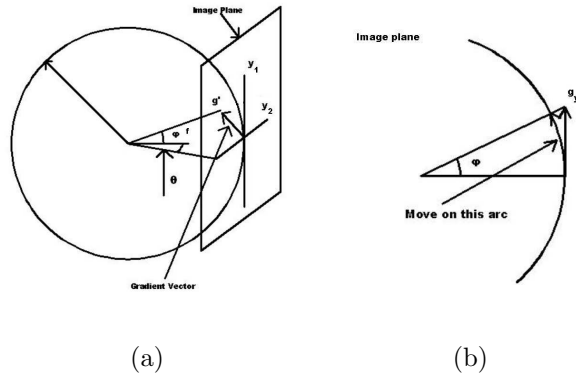


Figure A.5. (a) Computation of gradient vector; (b) Dependence of ϕ on the gradient vector.

APPENDIX B: APES G3 - USER GUIDE

In this Section, a user guide to APES for controlling APES is given. Before starting, make sure that the wiring is made appropriately according to the wiring diagram.

B.1. Power Connection

There are two main power connections for APES. One of them is the connection of the main board. This can be connected to a DC battery between 9-26 V. It should be noted that since the power consumption of the main board is 26W, an appropriate battery should be chosen. A LiPo or a lead-acid battery at 12V can suffice. The output of the battery should be connected to the wires as shown in Figure B.1 where the red one is for positive while black one is for ground connection.

The second connection is used to feed the two motor control boards for locomotion and pan-tilt camera movement. Camera also uses this connection, therefore since the camera is fed with 12 ± 1 V, a 12 V 7AH lead-acid battery is appropriate for this connection. A sample connection is as shown in Figure B.2.

B.2. Main Board Power Up

To power up the main board, the button on the second layer of APES is used (Figure B.3). When this button is pressed the computer starts up and the fan on the board starts to work. If there is a problem with the main board, make sure that the power to the power supply is sufficient by measuring the voltage on the line to the power supply from the batteries. Since a current of 2A passes through this wire, a small amount of voltage drop may result in a significant effect.

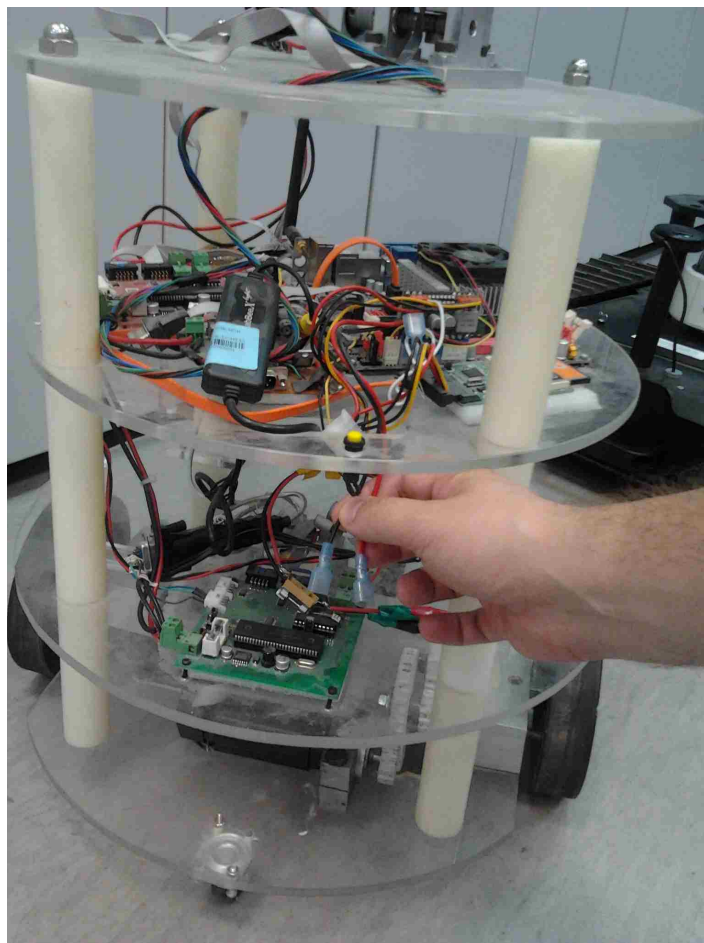


Figure B.1. The power connection for main board.

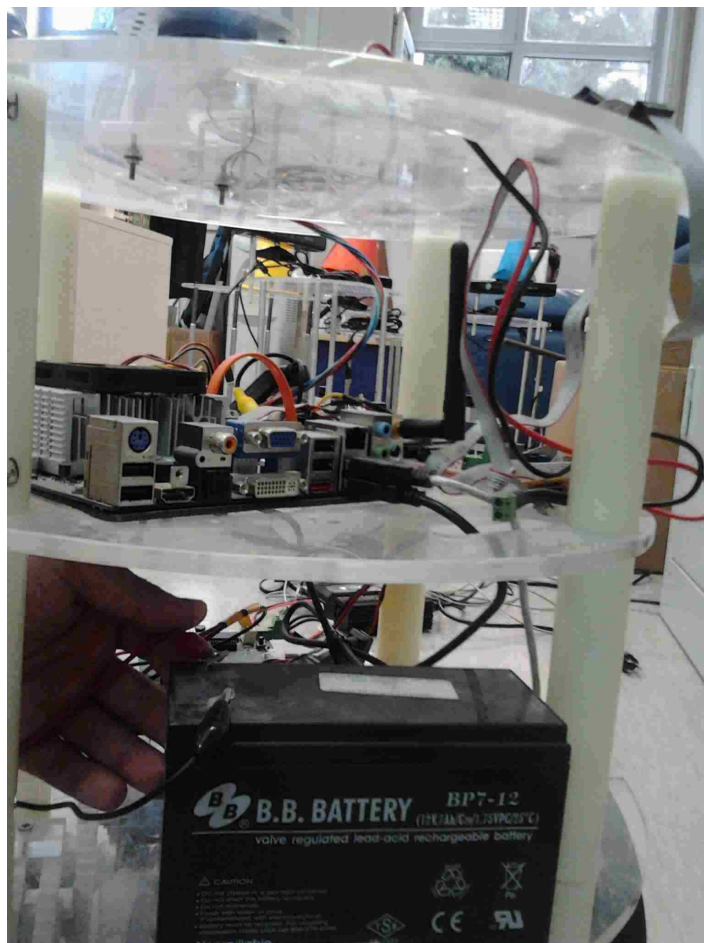


Figure B.2. The power connection for motor control boards and the camera.

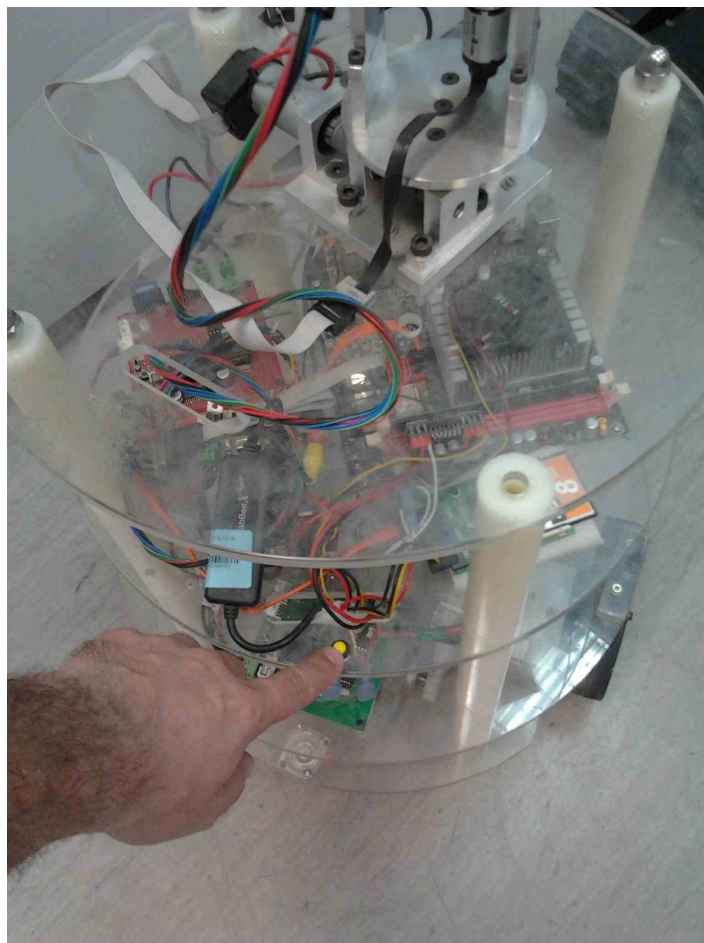


Figure B.3. The power button for the main board of APES.

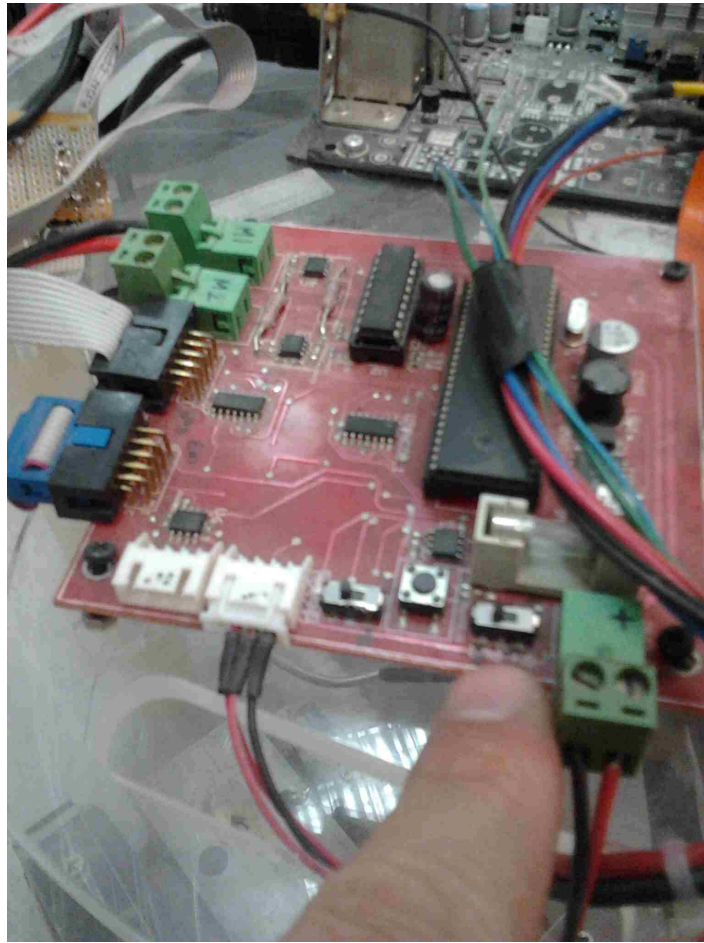


Figure B.4. On/off switch for MCB is pointed out. The board is ON, and in operation mode as observed from the state of the switch on the left.

B.3. Motor Control Board Power Up

To control the motors, the motor control boards (MCB) must be switched on. Make sure that they are turned on by checking their on/off switch shown in Figure B.4. Also, you can check the leds on the bottom surface of the MCBs. You should also check that the program switch is at operation mode, not programming mode. The programming switch is just besides the on/off switch. Before a program starts to execute, you can reset the Encoder and parameter information of the MCB by using the reset button which is in between the on/off switch and the program switch.

B.4. Connection to APES

There are two options for connecting to APES. We will cover them here.

B.4.1. Direct Connection

You can directly connect to APES by connecting a monitor, a keyboard and a mouse to it. This is the suggested way of connection if there is a problem with network connection. After the network problem is solved by connecting in this way, you can switch to remote connection which will be explained in the next Section. The robot can be controlled by opening the “apesExp” Visual Studio project. In this connection type, make sure that the motion of the robot will not affect the connections of the keyboard or the mouse.

B.4.2. Remote Connection

To be able to use Remote Connection, you must be connected to APES from a remote computer with a Microsoft Remote Desktop Connection program which is on the same Network with APES. It should also be noted that APES should be connected to the network (either via WiFi or LAN connection). This can be assured in two ways: One of them is that APES should be Shut down properly last time with a connection to a known network. The second way is to open APES with a direct connection, check and maintain the Network connection and then proceed with Remote Desktop Connection.

For Remote Desktop connection, the name of the APES on Network is “APES” with user name “vision”. By using the user name and the password, you can connect to and control APES remotely.

APPENDIX C: VISUAL FEATURES

The choice of $\mathcal{V}$ will vary depending on the task at hand and the representation mode. Here, visual features are Cartesian and Non-Cartesian filters⁷ of varying frequencies ω_k , $k = 1, \dots, K$ determined a priori and color. However, other visual features could easily be used without loss of generality.

The Cartesian filters are sinusoids with varying orientation $n \in \{1, \dots, N\}$. For each n and frequency ω_k , the corresponding filter $v_{nk} : A_b \rightarrow R^{\geq 0}$ is mathematically defined as

$$v_{nk}(p_1, p_2) = \cos(\omega_k(\mu_{1n}p_1 + \mu_{2n}p_2))$$

where $[p_1, p_2]^T \in A_b$ are the image plane coordinates, $\mu_{1n} = \sin(\frac{\pi}{N}(n-1))$, and $\mu_{2n} = \cos(\frac{\pi}{N}(n-1))$. The response of Cartesian filters is defined as the standard deviation of the response in the foveal region. The observation q_k is constructed accordingly.

The Non-Cartesian filters can be grouped as concentric, polar and hyperbolic filters. By initializing i appropriately and then incrementing it for each different ω , we obtain a filter set for each group. The concentric filters $v_{(N+1)k}$ are modeled as:

$$v_{(N+1)k}(p_1, p_2) = \cos(\omega_k(p_1^2 + p_2^2))$$

The polar filters are modeled as:

$$v_{(N+2)k}(p_1, p_2) = \cos(\omega_k \arctan(p_2/p_1))$$

⁷These filters have been determined to mimic the response characteristics of area V4 cells of macaque monkeys [169] and are simple to implement [90]

The hyperbolic filters are modeled as:

$$v_{(N+3)k}(p_1, p_2) = \cos(\omega_k \arctan(\frac{-\sin(\frac{\pi}{4})p_1 + \cos(\frac{\pi}{4})p_2}{\cos(\frac{\pi}{4})p_1 + \sin(\frac{\pi}{4})p_2}))$$

For all the non-Cartesian filters, the response is determined as the maximum response in the foveal region and again the observation q_k is constructed accordingly.

All the filters are placed into kernels of size $d_\phi \times d_\phi$ where d_ϕ is determined considering properties of central and peripheral vision. While peripheral vision which is beyond 5° eccentricity is known to be useful for gist recognition, central vision is more efficient when the visual field angle is greater than 7.8° [170]. Based on this, the filter size is set as $d_\phi = 9^\circ$. The filter with the lowest frequency ω_1 is mapped directly to this kernel. However, the remaining filters at each frequency k , $k = 2, \dots, K$ will have smaller sizes as designated by their frequency ω_k as $\frac{\omega_k-1}{\omega_k}d_\phi \times \frac{\omega_k-1}{\omega_k}d_\phi$. These filters are also placed in the same kernel in a co-centered manner – with zero padding outside. For $N = 6$ orientations and $K = 6$ frequencies, the generated set consists of 54 filters as shown in Figure C.1. Note that low spatial frequencies are chosen since in scene representation (specifically structure encoded in the phase spectrum), the most important information is contained in lower frequencies.

The last two visual features are based on color. Here, we use a and b channels from CIELAB color space [47] due to its stability in illumination changes [171]. Hence, the number of bubble surfaces is $N_v = 56$.

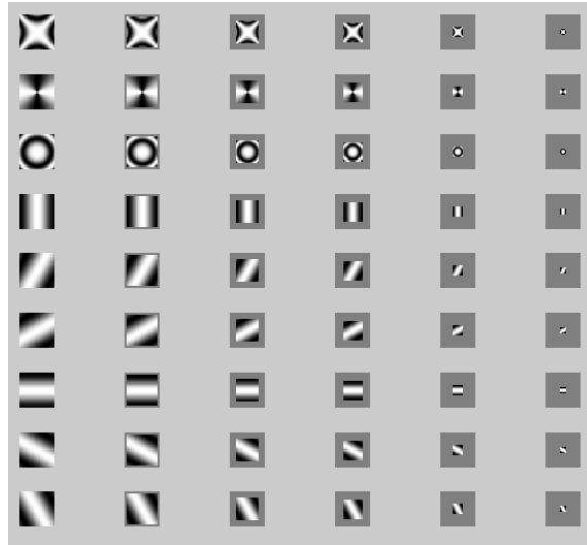


Figure C.1. Visual features – Cartesian and non-Cartesian filters with 6 frequency values.

APPENDIX D: COMPUTATION OF BUBBLE SURFACE COEFFICIENTS

Consider a bubble surface formed at a base $x \in \mathcal{X}$. Suppose that at time t , its camera axis is directed at $f \in \mathcal{F}$. Let $f = [f_1 \ f_2]^T$. In the double Fourier series, for each (m, n) , $m = 0, \dots, H_1$ and $n = 0, \dots, H_2$, the parameters λ_{mn} are defined as:

$$\lambda_{mn} = \begin{cases} \frac{1}{4} & \text{if } m = 0, n = 0 \\ \frac{1}{2} & \text{if } m > 0, n = 0 \text{ or } m = 0, n > 0 \\ 1 & \text{if } m > 0, n > 0 \end{cases}$$

Recall that the bubble surface coefficients consist of the collection of vector of double Fourier coefficients $z_{xi,mn}(t)$ as:

$$z_{xi,mn}(t) = \begin{bmatrix} \eta_{xi,mn}(t) & \beta_{xi,mn}(t) & \mu_{xi,mn}(t) & \nu_{xi,mn}(t) \end{bmatrix}^T \quad (\text{D.1})$$

With $b = [x^T \ f^T]^T$, by definition [91], rhs of Equation D.1 is equivalent to:

$$z_{xi,mn}(t) = \frac{1}{\pi^2} \begin{bmatrix} \int_0^{2\pi} \int_0^{2\pi} \rho_i(b, t) \cos(mf_1) \cos(nf_2) df_1 df_2 \\ \int_0^{2\pi} \int_0^{2\pi} \rho_i(b, t) \sin(mf_1) \cos(nf_2) df_1 df_2 \\ \int_0^{2\pi} \int_0^{2\pi} \rho_i(b, t) \cos(mf_1) \sin(nf_2) df_1 df_2 \\ \int_0^{2\pi} \int_0^{2\pi} \rho_i(b, t) \sin(mf_1) \sin(nf_2) df_1 df_2 \end{bmatrix} \quad (\text{D.2})$$

For incremental computation, for each f_i , $i = 1, 2$, first divide $[0, 2\pi]$ into N_i subintervals and let $\delta f_i = \frac{2\pi}{N_i}$. The rhs of Eqs. D.2 is then equivalently expressed in

term of summations as

$$\frac{1}{\pi^2} \sum_{l,k=0}^{N_1-1} \begin{bmatrix} \int_{k\delta f_1}^{(k+1)\delta f_1} \int_{l\delta f_2}^{(l+1)\delta f_2} \rho_i(b, t) \cos(m f_1) \cos(n f_2) df_1 df_2 \\ \int_{k\delta f_1}^{(k+1)\delta f_1} \int_{l\delta f_2}^{(l+1)\delta f_2} \rho_i(b, t) \sin(m f_1) \cos(n f_2) df_1 df_2 \\ \int_{k\delta f_1}^{(k+1)\delta f_1} \int_{l\delta f_2}^{(l+1)\delta f_2} \rho_i(b, t) \cos(m f_1) \sin(n f_2) df_1 df_2 \\ \int_{k\delta f_1}^{(k+1)\delta f_1} \int_{l\delta f_2}^{(l+1)\delta f_2} \rho_i(b, t) \sin(m f_1) \sin(n f_2) df_1 df_2 \end{bmatrix}$$

Suppose that the robot looks at a new viewing direction f' at $t + \delta t$ which means that its bubble space point is now $b' = [x f']^T$. At this new point, the robot extracts all the N_v visual features from the incoming visual data. Each bubble surfaces encodes its associated feature value via changing the deformation in $b'' \in N_\epsilon(b')$ by an amount $\rho_i(b'', t + \delta t) - \rho_i(b'', t)$. Consider the subintegral corresponding to each f' and approximate this subintegral as a linear function of δf_1 and δf_2 . With some mathematical manipulation, Equation D.2 simplifies as:

$$z_{xi,mn}(t + \delta t) = z_{xi,mn}(t) + \delta z(b', t, \delta t)$$

where $\delta z(b', t, \delta t)$ is defined as:

$$\frac{4}{N_1 N_2} \sum_{b'' \in N_\epsilon(b')} \begin{bmatrix} (-\rho_i(b'', t) + \rho_i(b'', t + \delta t)) \cos(m f_1'') \cos(n f_2'') \\ (-\rho_i(b'', t) + \rho_i(b'', t + \delta t)) \sin(m f_1'') \cos(n f_2'') \\ (-\rho_i(b'', t) + \rho_i(b'', t + \delta t)) \cos(m f_1'') \sin(n f_2'') \\ (-\rho_i(b'', t) + \rho_i(b'', t + \delta t)) \sin(m f_1'') \sin(n f_2'') \end{bmatrix}$$

This in turn implies that bubble surface coefficients are incrementally updated with each new viewing direction. As each bubble is initialized as $B_i(x, 0)$, the initial

value for $z_{xi,mn}(0)$ is defined as:

$$z_{xi,mn}(0) = \begin{bmatrix} \begin{cases} 4\rho_0 & m = n = 0 \\ 0 & \text{otherwise} \end{cases} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

APPENDIX E: BUBBLE DESCRIPTORS & ROTATIONAL INVARIANCE

This section proves rotational invariance of bubble descriptors. For simplifying the notation, $\cos(\cdot)$ function will be denoted as $c(\cdot)$ and $\sin(\cdot)$ function will be denoted as $s(\cdot)$

Proposition E.1. *Consider a bubble surface $B_i(x, t)$ with coefficients of double Fourier series as:*

$$z_{xi,mn}(t) = \begin{bmatrix} \eta_{xi,mn}(t) & \beta_{xi,mn}(t) & \mu_{xi,mn}(t) & \nu_{xi,mn}(t) \end{bmatrix}^T$$

for $m, n \in \mathbb{Z}^+$. Furthermore, define

$$I_{i,mn}(x, t) = z_{xi,mn}^T(t) z_{xi,mn}(t)$$

Then, $I_{i,mn}(x, t)$ is a rotational invariant.

Proof: First, general rotation of the bubble surface is considered by $\delta f \in \mathcal{F}$ where $\delta f = \begin{bmatrix} \delta f_1 & \delta f_2 \end{bmatrix}^T$. As surface points will move by δf amount on the bubble surface, the rotated bubble $\bar{B}$ will be

$$\bar{B}_i(x, t) = \left\{ \begin{bmatrix} f - \delta f \\ \rho_i(b, t) \end{bmatrix} \mid \forall f \in \mathcal{F} \text{ and } b = [x^T \ f^T]^T \right\}$$

Let the coefficients of the rotated bubble be denoted as:

$$\bar{z}_{xi,mn}(t) = \begin{bmatrix} \bar{\eta}_{xi,mn}(t) & \bar{\beta}_{xi,mn}(t) & \bar{\mu}_{xi,mn}(t) & \bar{\nu}_{xi,mn}(t) \end{bmatrix}^T$$

By definition and applying integration to trigonometric functions, $\bar{z}_{xi,mn}(t)$ can be equivalently expressed as

$$\bar{z}_{xi,mn}(t) = R_{\delta f,mn} z_{xi,mn}(t) \quad (\text{E.1})$$

where the matrix $R_{\delta f,mn}$ is an orthonormal matrix defined as:

$$\begin{bmatrix} c(m\delta f_1)c(n\delta f_2) & -c(n\delta f_2)s(m\delta f_1) & -c(m\delta f_1)s(n\delta f_2) & s(m\delta f_1)s(n\delta f_2) \\ c(n\delta f_2)s(m\delta f_1) & c(m\delta f_1)c(n\delta f_2) & -s(m\delta f_1)s(n\delta f_2) & -c(m\delta f_1)s(n\delta f_2) \\ c(m\delta f_1)s(n\delta f_2) & -s(m\delta f_1)s(n\delta f_2) & c(m\delta f_1)c(n\delta f_2) & -c(n\delta f_2)s(m\delta f_1) \\ s(m\delta f_1)s(n\delta f_2) & c(m\delta f_1)s(n\delta f_2) & c(n\delta f_2)s(m\delta f_1) & c(m\delta f_1)c(n\delta f_2) \end{bmatrix}$$

and $z_{xi,mn}(t)$ is vector of the coefficients of the unrotated bubble surface as defined by Equation D.2.

Define the rotational invariant as:

$$I_{i,mn}(x, t) = \bar{z}_{xi,mn}^T(t) \bar{z}_{xi,mn}(t)$$

Using right hand side of Equation E.1,

$$I_{i,mn}(x, t) = z_{xi,mn}^T(t) R_{\delta f,mn}^T R_{\delta f,mn} z_{xi,mn}(t)$$

Since $R_{\delta f,mn}^T R_{\delta f,mn} = 1$,

$$I_{i,mn}(x, t) = z_{xi,mn}^T(t) z_{xi,mn}(t)$$

Finally, recall that rotations are introduced by the changes in the robot's heading. Assuming that the axis of heading rotation is parallel to the axis of rotation of pan angle, a heading change of $\delta\alpha$ corresponds to a rotation with $\delta f_1 = 0$ and $\delta f_2 = \delta\alpha$. Hence the result holds. $\square$

APPENDIX F: PUBLICATIONS RELATED TO THESIS

In the course of the PhD studies, the following papers have been published/ submitted/ presented:

Chapter 2

- O. Erkent and H. I. Bozma, “Artificial Potential Functions Based Camera Movements And Visual Behaviors In Attentive Robots,” *Autonomous Robots*, vol. 32, pp. 15-35, 2012.
- O. Erkent and H. I. Bozma, “Color Based Saccades for Attention Control,” in *European Conference on Computer Vision Workshop: “Vision in Action: Efficient strategies for cognitive agents in complex environments,”* 2008.

Chapter 3

- O. Erkent and I. Bozma, “Bubble Space and Place Representation in Topological Maps,” *International Journal of Robotics Research*, vol. 32, no. 6, pp. 671 - 688, 2013.
- O. Erkent and H. I. Bozma, “Place Representation in Topological Maps Based on Bubble Space,” in *IEEE International Conference on Robotics and Automation*, USA, 2012 pp. 3497-3502
- O. Erkent and H. I. Bozma, “Attentive Vision, Scene Representation And Bubble Space,” in *Signal Processing and Communications Applications Conference (SIU), IEEE* , 2010, pp. 387-390.
- S. Cayci, O. Erkent, and H. I. Bozma, “Attentive Mobile Robot Visual Maps Via Bubble Memory,” in *Perception 38 ECVF Abstract Supplement*, 2009, p. 47.
- H. Karaoguz, O. Erkent, H. Bayram and I. Bozma “Tek Robottan Çoklu Robotlara Ortam Haritalama”, in *EMO Bilimsel Dergi*, 4 (2), 2013.
- O. Erkent and I. Bozma “Robotlar için Görsel Temelli Yönelim ve ortam Açıklık

Bilgilerinin Kestirimi”, in *ASYU (Innovations and Applications in Intelligent Systems Symposium)*, pp 75-79 Turkey, 2012

- H. Karaoguz, O. Erkent and I. Bozma “Topolojik Haritalarda 3B Uzaklık Ölçüm Bilgisine Dayalı Yer Gösterimi”, in *ASYU (Innovations and Applications in Intelligent Systems Symposium)*, pp 153-157 Turkey, 2012

Chapter 4

- O. Erkent and I. Bozma, “Integrating Cue Descriptors in Bubble Space for Place Recognition,” in *International Conference On Computer Vision Systems, Lecture Notes in Computer Science 7963*, 2013, pp. 314-323.
- O. Erkent and I. Bozma, “Cue Descriptor Semantic Forests for Topological Place Recognition,” *IEEE Transaction on Robotics*, (submitted).

REFERENCES

1. Kuipers, B., J. J. Modayil, P. Beeson, M. MacMahon and F. Savelli, “Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy”, *IEEE International Conference on Robotics and Automation*, pp. 4845–4851, 2004.
2. Moravec, H. and A. Elfes, “High Resolution Maps From Wide Angle Sonar”, *Proceedings IEEE International Conference Robotics and Automation*, Vol. 2, pp. 116–121, IEEE, 1985.
3. Brunskill, E., T. Kollar and N. Roy, “Topological Mapping Using Spectral Clustering And Classification”, *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3491–3496, 2007.
4. Kuipers, B. and Y.-T. Byun, “A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations”, *Robotics and Autonomous Systems*, Vol. 8, No. 1, pp. 47–63, 1991.
5. Remolina, E. and B. Kuipers, “Towards A General Theory Of Topological Maps”, *Artificial Intelligence*, Vol. 152, No. 1, pp. 47–104, 2004.
6. Ulrich, I. and I. Nourbakhsh, “Appearance-Based Place Recognition For Topological Localization”, *IEEE International Conference on Robotics*, Vol. 2, pp. 1023–1029, 2000.
7. Fazl-Ersi, E. and J. K. Tsotsos, “Histogram of Oriented Uniform Patterns for robust place recognition and categorization”, *The International Journal of Robotics Research*, Vol. 31, No. 4, pp. 468–483, 2012.
8. Wang, M.-L. and H.-Y. Lin, “An Extended-HCT Semantic Description For Visual Place Recognition”, *The International Journal of Robotics Research*, Vol. 30,

- No. 11, pp. 1403–1420, 2011.
9. Konolige, K., J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit and P. Fua, “View-based Maps”, *The International Journal of Robotics Research*, Vol. 29, No. 8, pp. 941–957, 2010.
 10. Tsotsos, J. K., “On The Relative Complexity Of Active Vs. Passive Visual Search”, *International Journal of Computer Vision*, Vol. 7, No. 2, pp. 127–141, 1992.
 11. Kant, I., *Critique of Pure Reason*, Translated by Werner Pluhar. Hackett Publishing, 1996.
 12. Ilhan, B. D., O. Erkent and H. I. Bozma, “Saccades and Fixating Using Artificial Potential Functions,” , *Proceedings of IEEE/IRJ Conference on Intelligent Robots and Systems*, pp. 5819–5824, 2006.
 13. Soyer, C., H. I. Bozma and Y. Istefanopulos, “A New Memory Model for Selective Perception Systems” , *Proceedings of IEEE/RSJ International Conference on Robots and Systems*, pp. 2304–2309, 2000.
 14. Soyer, C., H. I. Bozma and Y. Istefanopulos, “APES - Biologically Motivated Attentive Robot”, *Autonomous Robots*, Vol. 20, pp. 61–80, 2006.
 15. Pronobis, A. and B. Caputo, “Confidence-Based Cue Integration For Visual Place Recognition”, *Intelligent Robots and Systems. IEEE/RSJ International Conference on*, pp. 2394–2401, 2007.
 16. Pronobis, A. and B. Caputo, “{COLD}: {CO}sy {L}ocalization {D}atabase”, *The International Journal of Robotics Research*, Vol. 28, No. 5, 2009.
 17. Luo, J., A. Pronobis, B. Caputo and P. Jensfelt, “Incremental Learning For Place Recognition In Dynamic Environments”, *IEEE/RSJ International Conference on*

Intelligent Robots and Systems, pp. 721–728, 2007.

18. Orabona, F., C. Castellini, B. Caputo, L. Jie and G. Sandini, “On-Line Independent Support Vector Machines”, *Pattern Recognition*, Vol. 43, No. 4, pp. 1402–1412, 2010.
19. Erkent, O. and I. Bozma, “Bubble Space and Place Representation in Topological Maps”, *The International Journal of Robotics Research*, Vol. 32, No. 6, pp. 671 – 688, 2013.
20. Yeh, T. and T. Darrell, “Dynamic Visual Category Learning”, *Computer Vision and Pattern Recognition. IEEE Conference on*, pp. 1–8, 2008.
21. Posner, I., D. Schroeter and P. Newman, “Using Scene Similarity for Place Labelling”, O. Khatib, V. Kumar and D. Rus (Editors), *Experimental Robotics*, Vol. 39 of *Springer Tracts in Advanced Robotics*, pp. 85–98, Springer Berlin Heidelberg, 2008.
22. Ballard, H. D. and C. M. Brown, “Principles of Animate Vision”, *Computer Vision, Graphics, and Image Processing*, Vol. 56, No. 1, 1992.
23. Tsotsos, J. K., “A Complexity Level Analysis Of Immediate Vision”, *International Journal of Computer Vision*, Vol. 1, pp. 303–320, 1988.
24. Tsotsos, J. K., “Motion Understanding: Task-Directed Attention and Representations, that Link Perception with Action”, *International Journal of Computer Vision*, Vol. 45, pp. 265–280, 2001.
25. Frintrop, S., E. Rome and H. I. Christensen, “Computational Visual Attention Systems and Their Cognitive Foundations:A Survey”, *ACM Transactions on Applied Perception*, Vol. 7, No. 1, 2010.
26. Itti, L., C. Koch and E. Niebur, “A Model of Saliency Based Visual Attention

- for Rapid Scene Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, pp. 1254–1259, 1998.
27. Tsotsos, J. K., S. Culhane, W. Wai, Y. Lai, N. Davis and F. Nufflo, “Modeling Visual Attention Via Selective Tuning”, *Artificial Intelligence*, Vol. 73, pp. 507–545, 1995.
 28. Navalpakkam, V. and L. Itti, “Modeling The Influence Of Task On Attention”, *Vision Research*, Vol. 45, No. 2, pp. 205–231, 2005.
 29. Saul, L. K., Y. Weiss and L. Bottou (Editors), *Discriminant Saliency For Visual Recognition From Cluttered Scenes*, MIT Press, 2005.
 30. Bruce, N. D. and J. K. Tsotsos, “Saliency, Attention And Visual Search: An Information Theoretic Approach”, *Journal of Vision*, Vol. 9, No. 3, pp. 1–24, 2009.
 31. Gao, D. and N. Vasconcelos, “Decision-Theoretic Saliency: Computational Principles, Biological Plausibility, and Implications for Neurophysiology and Psychophysics”, *Neural Computation*, Vol. 21, No. 1, pp. 239–271, 2009.
 32. Butko, N. J., L. Zhang, G. W. Cottrell and J. Movellan, “Visual Saliency Model for Robot Cameras”, *Conference on Robotics*, 2008.
 33. Rimey, R. D. and C. M. Brown, “Controlling Eye Movements With Hidden Markov Models”, *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 47–65, 1991.
 34. Hou, X. and L. Zhang, “Dynamic Visual Attention: Searching for coding length increments”, *Conference on Neural Information Processing Systems*, 2008.
 35. Orabona, F., G. Metta and G. Sandini, “Object-based Visual Attention: a Model for a Behaving Robot”, *International Workshop on Attention and Performance*

in Computational Vision, 2005.

36. Clark, J. J. and N. J. Ferrier, “Modal Control Of An Attentive Vision System”, *Second International Conference on Computer Vision*, pp. 514–523, 1988.
37. Ude, A., V. Wyart, L. Lin and G. Cheng, “Distributed Visual Attention on a Humanoid Robot”, *Proceedings of 5th IEEE-RAS International Conference on Humanoid Robots*, 2005.
38. Meger, D., P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little and D. G. Lowe., “Curious George: An Attentive Semantic Robot”, *IEEE International Conference on Robots and Systems, Workshop: From sensors to human spatial concepts*, 2007.
39. Dankers, A., N. Barnes and A. Zelinsky, “A Reactive Vision System: Active-Dynamic Saliency”, *Proceedings of the 5th International Conference on Computer Vision Systems*, IEEE, Los Alamitos, CA, 2007.
40. Vijakumar, S., J. Conradt, T. Shibata and S. Schaal, “Overt Visual Attention For A Humanoid Robot”, *Proceedings of the International Conference on Intelligence in Robotics and Autonomous Systems*, pp. 2332–2337, 2001.
41. Tsotsos, J. K. and K. Shubina, “Attention and Visual Search : Active Robotic Vision Systems that Search”, *The 5th International Conference on Computer Vision Systems, Bielefeld*, 2007.
42. Rasolzadeh, B., M. Bjorkman and J.-O. Eklundh, “An Attentional System Combining Top-Down And Bottom-Up Influences”, *Attention in cognitive systems. Theories and systems from an interdisciplinary viewpoint.*, pp. 123–140, 2007.
43. Forssen, P., D. Meger, K. Lai, J. J. L. S. Helmer and D.G.Lowe, “Informed Visual Search: Combining Attention and Object Recognition”, *Proceedings of IEEE International Conference on Robotics and Automation.*, pp. 935–942, 2008.

44. Coombs, D. and C. Brown, “Real-Time Smooth Pursuit Tracking For A Moving Binocular Robot”, *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 23–28, 1992.
45. Rivlin, E. and H. Rotstein, “Control of a Camera for Active Vision: Foveal Vision, Smooth Tracking and Saccade”, *International Journal of Computer Vision*, Vol. 39, No. 2, pp. 81–96, 2000.
46. Frintrop, S., A. Konigs, F. Hoeller and D. Schulz, “A Component-Based Approach To Visual Person Tracking From A Mobile Platform”, *International Journal of Social Robotics*, 2010.
47. CIE, *Colorimetry (Second Edition)*, Tech. Rep. 15.2, Central Bureau of the Commission Internationale de Le clairage, Vienna, Austria, 1986.
48. Khatib, O., “Real-time Obstacle Avoidance for Manipulators and Mobile Robots”, *The International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90–99, 1986.
49. Koditschek, D. E., “An Approach to Autonomous Robot Assembly”, *Robotica*, Vol. 12, pp. 137–155, 1994.
50. Tatler, B. W. and N. J. Wade, “On Nystagmus, Saccades, and Fixations”, *Perception*, Vol. 32, pp. 167–184, 2003.
51. Krekelberg, B., “Saccadic Suppression”, *Current Biology*, Vol. 20, No. 5, pp. 228–229, 2010.
52. Guez, J. E., P. Marchal, J. F. L. Gargasson, Y. Grall and J. K. O’Regan, “Eye Fixations Near Corners: Evidence For A Centre Of Gravity Calculation Based On Contrast, Rather Than Luminance Or Curvature”, *Vision Research*, Vol. 34, No. 12, pp. 1625–1635, 1994.
53. Trajkovic, M. and M. Hedley, “Fast Corner Detection”, *Image and Vision Com-*

- puting*, Vol. 16, No. 2, pp. 75–87, 1998.
54. Turatto, M. and G. Galfano, “Color, Form And Luminance Capture Attention In Visual Search”, *Vision Research*, Vol. 40, No. 13, pp. 1639–1643, 2000.
 55. Nothdurft, H. C., “The Role Of Features In Preattentive Vision: Comparison Of Orientation, Motion And Color Cues”, *Vision Research*, Vol. 33, No. 14, pp. 1937–1958, 1993.
 56. Xu, Y., E. C. Higgins, M. Xiao and M. Pomplun, “Mapping the Color Space of Saccadic Selectivity in Visual Search”, *Cognitive Science*, Vol. 31, p. 877:887, 2007.
 57. Sugi, H. and Y. Takeda, “Effect Of Previously Fixated Locations On Saccade Trajectory During Free Visual Search”, *Vision Research*, Vol. 46, pp. 3831–3844, 2006.
 58. Theeuwes, J. and S. V. der Stigchel, “Saccade Trajectory Deviations And Inhibition-Of-Return: Measuring The Amount of Attentional Processing”, *Vision Research*, Vol. 49, No. 10, pp. 1307–1315, 2009.
 59. Walther, D. and C. Koch, “Modeling Attention To Salient Proto-Objects”, *Neural Networks*, Vol. 19, pp. 1395–1407, 2006.
 60. Smith, M., I. Baldwin, W. Churchill, R. Paul and P. Newman, “The New College Vision and Laser Data Set”, *The International Journal of Robotics Research*, Vol. 28, No. 5, pp. 595–599, 2009.
 61. Elfes, A., “Sonar-Based Real World Mapping And Navigation”, *IEEE Transactions on Robotics and Automation*, Vol. 3, No. 3, pp. 249–265, 1987.
 62. Jaulin, L., “Range-Only SLAM With Occupancy Maps: A Set-Membership Approach”, *Robotics, IEEE Transactions on*, Vol. 27, No. 5, pp. 1–6, 2011.

63. Sim, R. and G. Dudek, “Learning And Evaluating Visual Features For Pose Estimation”, *Proceedings 7th International Conference on Computer Vision*, pp. 1217–1222, Greece, 1999.
64. Se, S., D. G. Lowe and J. J. Little, “Vision-Based Global Localization And Mapping For Mobile Robots”, *Robotics, IEEE Transactions on*, Vol. 21, No. 3, pp. 364–375, 2005.
65. Frese, U., “Treemap: An Algorithm For Indoor Simultaneous Localization And Mapping”, *Autonomous Robots*, Vol. 21, No. 2, pp. 103–122, 2006.
66. Bosse, M., P. Newman, J. Leonard and S. Teller, “Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework”, *The International Journal of Robotics Research*, Vol. 23, No. 12, pp. 1113–1139, 2004.
67. Vasudevan, S., S. Guchter and V. Nguyen, “Cognitive Maps For Mobile Robots: An Object Based Approach”, *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 359–371, 2007.
68. Davison, A. J., I. D. Reid, N. D. Molton and O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 1052–1067, 2007.
69. Jogan, M. and A. Leonardis, “Robust Localization Using Panoramic View-Based Recognition”, *Pattern Recognition, 2000. Proceedings of 15th International Conference on*, Vol. 4, pp. 136 –139, 2000.
70. Oliva, A. and A. Torralba, “Modeling the Shape of the Scene: A Holistic Representation of the SpatialEnvelope”, *International Journal of Computer Vision*, Vol. 42, No. 3, pp. 145–175, 2001.
71. Torralba, A., K. P. Murphy, W. T. Freeman and M. A. Rubin, “Context-Based

- Vision System For Place And Object Recognition”, *Computer Vision, IEEE International Conference on*, Vol. 1, p. 273, 2003.
72. Qiu, G., “Indexing Chromatic And Achromatic Patterns For Content-Based Colour Image Retrieval”, *Pattern Recognition*, Vol. 35, No. 8, pp. 1675–1686, 2002.
 73. Murillo, A. C., C. Sagues, J. J. Guerrero, T. Goedemi, T. Tuytelaars and L. V. Gool, “From Omnidirectional Images To Hierarchical Localization”, *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 372–382, 2007.
 74. Harris, C. and M. Stephens, “A Combined Corner And Edge Detector”, *Alvey Vision Conference*, pp. 147–151, 1988.
 75. Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vis.*, Vol. 60, No. 2, pp. 91–110, 2004.
 76. Bay, H., T. Tuytelaars and L. Van Gool, “SURF: Speeded up robust features”, A. Leonardis, H. Bischof and A. Pinz (Editors), *European Conference on Computer Vision*, Vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer Berlin - Heidelberg, Austria, 2006.
 77. Lamon, R., I. Nourbakhsh, B. Jensenl and R. Siegwart, “Deriving And Matching Image Fingerprint Sequences For Mobile Robot Localization”, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1609–1610, 2001.
 78. Nowak, E., F. Jurie and B. Triggs, “Sampling Strategies for Bag-of-Features Image Classification”, A. Leonardis, H. Bischof and A. Pinz (Editors), *European Conference on Computer Vision*, Vol. 3954 of *Lecture Notes in Computer Science*, pp. 490–503, Springer Berlin / Heidelberg, 2006.
 79. Lazebnik, S., C. Schmid and J. Ponce, “Beyond Bags Of Features: Spatial Pyra-

- mid Matching For Recognizing Natural Scene Categories”, *Proceedings IEEE International Conference Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.
80. Bosch, A., A. Zisserman and X. Muñoz, “Scene Classification Using a Hybrid Generative/Discriminative Approach”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 4, pp. 712–727, 2008.
 81. Fraundorfer, F., C. Engels and D. Nister, “Topological Mapping, Localization And Navigation Using Image Collections”, *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems*, pp. 3387–3872, 2007.
 82. Cummins, M. and P. Newman, “Appearance-Only Slam At Large Scale With FAB-MAP 2.0”, *The International Journal of Robotics Research*, Vol. 30, No. 9, pp. 1100–1123, 2010.
 83. Wolf, J., W. Burgard and H. Burkhardt, “Robust Vision-Based Localization By Combining An Image-Retrieval System With Monte Carlo Localization”, *Robotics, IEEE Transactions on*, Vol. 21, No. 2, pp. 208–216, 2005.
 84. Albus, J. S., “Outline for a Theory of Intelligence”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, pp. 473–509, 1991.
 85. Peters, R. A., K. E. Hambuchen, K. Kawamura and D. M. Wilkes, “The Sensory Ego-Sphere as a Short-Term Memory for Humanoids”, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2001.
 86. Meilland, M., A. I. Comport and P. Rives, “A Spherical Robot-Centered Representation For Urban Navigation”, *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 5196–5201, 2010.
 87. Makadia, A. and K. Daniilidis, “Rotation Recovery From Spherical Images Without Correspondences”, *IEEE Transactions on Pattern Analysis and Machine In-*

- telligence*, Vol. 28, No. 7, pp. 1170–1175, 2006.
88. Friedrich, H., D. Dederscheck, K. Krajsek and R. Mester, “View-Based Robot Localization Using Illumination-Invariant Spherical Harmonics Descriptors”, *Proceedings of International Joint Conference on Computer Vision and Computer Graphics Theory and Applications*, Vol. 2, pp. 543–550, 2008.
 89. Rossi, F., A. Ranganathan, F. Dellaert and E. Menegatti, “Toward Topological Localization with Spherical Fourier Transform and Uncalibrated Camera”, *Omnidirectional Robot Vision Workshop, International Conference Simulation, Modeling, and Programming for Autonomous Robots*, pp. 319–330, 2008.
 90. Cakiroglu, G., C. Soyer, and H. I. Bozma, “Biologically Inspired Cartesian and Non-Cartesian Filters for Attentional Sequences”, *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1261–1274, 2003.
 91. Tolstov, G. P., *Fourier Series*, Prentice-Hall, New York, 1962.
 92. Choset, H., K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, 2005.
 93. Gutmann, J.-S. and K. Konolige, “Incremental Mapping Of Large Cyclic Environments”, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 318 –325, 1999.
 94. Cummins, M. and P. Newman, “Probabilistic Appearance Based Navigation and Loop Closing”, *Proceedings of IEEE International Conference on Robotics*, pp. 2042–2048, 2007.
 95. Ullah, M. M., A. Pronobis, B. Caputo, J. Luo, R. Jensfelt and H. I. Christensen, “Towards Robust Place Recognition For Robot Localization”, *Robotics and Automation, IEEE International Conference on*, pp. 530–537, 2008.

96. Cristianini, N. and J. Shawe-Taylor, *An Introduction To Support Vector Machines And Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.
97. Cortes, C. and V. Vapnik, "Support-vector Networks", *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.
98. Bartlett, P. B. Schoelkopf, D. Schurmans and A. J. Smola (Editors), *Advances in Large Margin Classifiers*, chap. Probabilis, pp. 61–74, 1999.
99. Lin, H.-T., C.-J. Lin and R. Weng, "A Note On Platt'S Probabilistic Outputs For Support Vector Machines", *Machine Learning*, Vol. 68, No. 3, pp. 267–276, 2007.
100. Erkent, O. and H. I. Bozma, "Artificial Potential Functions Based Camera Movements And Visual Behaviors In Attentive Robots", *Autonomous Robots*, Vol. 32, pp. 15–35, 2012.
101. Chang, C.-C. and C.-J. Lin, "{LIBSVM}: A Library For Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 1–27, 2011.
102. Cummins, M. and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance", *The International Journal of Robotics Research*, Vol. 27, pp. 647–665, 2008.
103. Maddern, W., M. Milford and G. Wyeth, "Towards Persistent Localization and Mapping with a Continuous Appearance-based Topology", R. Nicholas (Editor), *Proceedings of Robotics Science and Systems Conference*, 2012.
104. Kuipers, B., R. Browning, B. Gribble, M. Hewett and E. Remolina, "The Spatial Semantic Hierarchy", *Artificial Intelligence*, Vol. 119, pp. 191–233, 2000.
105. Mozas, O. M., R. Triebel, P. Jensfelt, A. Rottmann and W. Burgard, "Supervised Semantic Labeling Of Places Using Information Extracted From Laser And Vision

- Sensor Data”, *Robotics and Autonomous Systems*, Vol. 55, No. 5, pp. 391–402, 2007.
106. Marinakis, D. and G. Dudek, “Pure Topological Mapping in Mobile Robotics”, *Robotics, IEEE Transactions on*, Vol. 26, No. 6, pp. 1051–1064, 2010.
 107. Clark, J. J. and A. L. Yuille, *Data Fusion for Sensory Information Processing Systems*, Kluwer Academic Publisher, 1990.
 108. Triesch, J., D. H. Ballard and R. A. Jacobs, “Fast Temporal Dynamics Of Visual Cue Integration”, *Perception*, Vol. 31, pp. 421–434, 2002.
 109. Poggio, T., V. Torre and C. Koch, “Computational Vision And Regularization Theory”, *Nature*, Vol. 317, No. 26, pp. 314–319, 1985.
 110. Aloimonos, J. and D. Shulman, *Integration of Visual Modules: an Extension of the Marr Paradigm*, Academic Press, Ottawa, 1989.
 111. Bozma, H. and J. Duncan, “A Game-Theoretic Approach To Integration Of Modules”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 11, pp. 1074 –1086, 1994.
 112. Kragic, D. and H. Christensen, “Cue Integration For Visual Servoing”, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 1, pp. 18 –27, 2001.
 113. Nilsback, M. E. and B. Caputo, “Cue Integration Through Discriminative Accumulation”, *Computer Vision and Pattern Recognition. Proceedings of IEEE Computer Society Conference on*, Vol. 2, pp. 578–585, 2004.
 114. Erkent, O. and I. Bozma, “Integrating Cue Descriptors in Bubble Space for Place Recognition”, M. Chen, B. Leibe and B. Neumann (Editors), *International Conference On Computer Vision Systems, Lecture Notes in Computer Science 7963*, pp. 314–323, Springer, Heidelberg, St. Petersburg, 2013.

115. Pronobis, A., O. Martinez Mozos, B. Caputo and P. Jensfelt, “Multi-modal Semantic Place Classification”, *The International Journal of Robotics Research*, Vol. 29, No. 2-3, pp. 298–320, 2010.
116. Zivkovic, Z., B. Bakker and B. Krose, “Hierarchical Map Building Using Visual Landmarks And Geometric Constraints”, *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 2480–2485, 2005.
117. Weiss, C., H. Tamimi, A. Masselli and A. Zell, “A Hybrid Approach For Vision-Based Outdoor Robot Localization Using Global And Local Image Features”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1047–1052, 2007.
118. Valgren, C. and A. Lilienthal, “Incremental Spectral Clustering And Seasons: Appearance-Based Localization In Outdoor Environments”, *IEEE International Conference on Robotics and Automation*, pp. 1856–1861, 2008.
119. Linde, O. and T. Lindeberg, “Object Recognition Using Composed Receptive Field Histograms Of Higher Dimensionality”, *Pattern Recognition, Proceedings of the 17th International Conference on*, Vol. 2, pp. 1–6, 2004.
120. Wu, J. and J. Rehg, “Where Am I: Place Instance And Category Recognition Using Spatial Pact”, *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2008.
121. Ranganathan, A., “PLISS: Detecting and Labeling Places Using Online Change-Point Detection”, *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, 2010.
122. Bulthoff, H. and A. Yuille, “Bayesian Models For Seeing Shapes And Depth”, *Comments on Theoretical Biology*, Vol. 4, No. 2, pp. 283–313, 1991.
123. Morgan, M., G. C. Deangelis and D. E. Angelaki, “Multisensory Integration in

- Macaque Visual Cortex Depends on Cue Reliability”, *Neuron*, Vol. 59, No. 4, pp. 662–673, 2008.
124. Trommershauser, J., K. Kording and M. S. Landy, *Sensory Cue Integration*, Oxford University Press, 2011.
 125. Rottmann, A., O. M. Mozos, C. Stachniss, W. Burgard, O. Martinez, M. Cyrill and S. W. Burgard, “Semantic Place Classification of Indoor Environments With Mobile Robots using Boosting”, in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 1306–1311, 2005.
 126. Steder, B., M. Ruhnke, S. Grzonka and W. Burgard, “Place Recognition in 3D Scans Using a Combination of Bag of Words and Point Feature based Relative Pose Estimation”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1249–1255, 2011.
 127. Pronobis, A., O. Martinez Mozos and B. Caputo, “Svm-Based Discriminative Accumulation Scheme For Place Recognition”, *IEEE International Conference on Robotics and Automation*, pp. 522–529, 2008.
 128. Joachims, T. and C.-N. J. Yu, “Sparse Kernel SVMs via Cutting-plane Training”, *Machine Learning*, Vol. 76, No. 2-3, pp. 179–193, 2009.
 129. Brautigam, C. G., *A Model-Free Voting Approach to Cue Integration*, Ph.D. Thesis, Kungl Tekniska Hogskolan, 1998.
 130. Landy M.S., M. S. B. and D. C. Knill, “Ideal-Observer Models of Cue Integration”, K. K. Trommershauser J. and M. S. Landy (Editors), *Sensory Cue Integration*, pp. 5–29, Oxford University Press, 2011.
 131. DeCarlo, D., “Towards Real-Time Cue Integration by Using Partial Results”, A. Heyden, G. Sparr, M. Nielsen and P. Johansen (Editors), *European Conference on Computer Vision*, Vol. 2353 of *Lecture Notes in Computer Science*, pp. 41–54,

Springer Berlin / Heidelberg, 2002.

132. Ernst, M. O., “A Bayesian View On Multimodal Cue Integration”, Knoblich, Thornton, M. Grosjean and M. Shiffrar (Editors), *Human Body Perception From The Inside Out*, pp. 105–131, Oxford University Press, 2006.
133. Alpert, S., M. Galun, A. Brandt and R. Basri, “Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 2, pp. 315–327, 2012.
134. Galindo, C., A. Saffiotti, S. Coradeschi and P. Buschka, “Multi-Hierarchical Semantic Maps for Mobile Robotics”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3492–3497, 2005.
135. Serrano, N., A. E. Savakis and J. Luo, “Improved Scene Classification Using Efficient Low-Level Features And Semantic Cues”, *Pattern Recognition*, Vol. 37, No. 9, pp. 1773–1784, 2004.
136. Pronobis, A. and P. Jensfelt, “Large-Scale Semantic Mapping And Reasoning With Heterogeneous Modalities”, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3515 –3522, 2012.
137. Zender, H., O. Martínez Mozos, P. Jensfelt, G. J. M. Kruijff and W. Burgard, “Conceptual Spatial Representations For Indoor Mobile Robots”, *Robotics and Autonomous Systems*, Vol. 56, No. 6, pp. 493–502, 2008.
138. Meger, D., P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little and D. G. Lowe, “Curious George: An attentive Semantic Robot”, *Robotics and Autonomous Systems*, Vol. 56, No. 6, pp. 503–511, 2008.
139. Vasudevan, S. and R. Siegwart, “Bayesian Space Conceptualization And Place Classification For Semantic Maps In Mobile Robotics”, *Robotics and Autonomous Systems*, Vol. 56, No. 6, pp. 522–537, 2008.

140. Nüchter, A. and J. Hertzberg, “Towards Semantic Maps For Mobile Robots”, *Robotics and Autonomous Systems*, Vol. 56, No. 11, pp. 915–926, 2008.
141. Pronobis, A. and P. Jensfelt, “Hierarchical Multi-modal Place Categorization”, *Proceedings of the European Conference on Mobile Robotics (ECMR’11)*, 2011.
142. Aydemir, A., A. Pronobis, M. Gobelbecker and P. Jensfelt, “Active Visual Object Search in Unknown Environments Using Uncertain Semantics”, *(to appear) IEEE Transactions on Robotics*, 2013.
143. Tversky, B. and K. Hemenway, “Categories of Environmental Scenes”, *Cognitive Psychology*, Vol. 15, pp. 121–149, 1983.
144. Mojsilović, A., J. Gomes and B. Rogowitz, “Semantic-friendly Indexing And Quering Of Images Based On The Extraction Of The Objective Semantic Cues”, *International Journal of Computer Vision*, Vol. 56, No. 1, pp. 79–107, 2004.
145. Li, X., C. Wu, C. Zach, S. Lazebnik and J.-M. Frahm, “Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs”, *European Conference on Computer Vision*, pp. 427–440, 2008.
146. Quattoni, A. and A. Torralba, “Recognizing indoor scenes”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 36–41, 2009.
147. Horn, R. A. and C. R. Johnson, “Topics in Matrix Analysis (Vol. 2)”, *Cambridge University Press, England*, 1991.
148. Hsu, C. W. and C. J. Lin, “A Comparison On Methods For Multi-Class Support Vector Machines”, *IEEE Transactions on Neural Networks*, Vol. 13, pp. 415–425, 2002.
149. Sun, Z., “Adaptation For Multiple Cue Integration”, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1,

pp. 440–445, 2003.

150. Erkent, O. and H. I. Bozma, “Place Representation in Topological Maps Based on Bubble Space”, *IEEE International Conference on Robotics and Automation*, 2012.
151. Sibson, R., “SLINK: An Optimally Efficient Algorithm For The Single-Link Cluster Method”, *The Computer Journal*, Vol. 16, No. 1, pp. 30–34, 1973.
152. Luo, R., S. Piao and H. Min, “Simultaneous Place And Object Recognition With Mobile Robot Using Pose Encoded Contextual Information”, *Robotics and Automation, IEEE International Conference on*, pp. 2792–2797, 2011.
153. Karasfi, B. and T. Hong, “Speedup Robust Features Based Unsupervised Place Recognition For Assistive Mobile Robot”, *Pattern Analysis and Intelligent Robotics (ICPAIR), International Conference on. Vol. 1. IEEE*, pp. 97–102, 2011.
154. Rogers, J. and H. Christensen, “A Conditional Random Field Model For Place And Object Classification”, *Robotics and Automation, IEEE International Conference on*, pp. 1766–1772, 2012.
155. Ursic, P., M. Kristan, D. Skocaj and A. Leonardis, “Room classification using a hierarchical representation of space”, *Intelligent Robots and Systems IEEE/RSJ International Conference on*, pp. 1371–1378, 2012.
156. Topp, E. A. and H. I. Christensen, “Topological Modelling for Human Augmented Mapping”, *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 2257–2263, 2006.
157. Shi, L., S. Kodagoda and G. Dissanayake, “Application Of Semi-Supervised Learning With Voronoi Graph For Place Classification”, *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 2991–2996, 2012.

158. Martinez-Gomez, J. and B. Caputo, "Towards Semi-Supervised Learning Of Semantic Spatial Concepts", *IEEE International Conference on Robotics and Automation*, pp. 1936–1943, IEEE, 2011.
159. Janet, J. and S. Scoggins, "Self-Organizing Geometric Certainty Maps: A Compact And Multifunctional Approach To Map Building, Place Recognition And Motion Planning", *Robotics and Automation, IEEE International Conference on*, pp. 3421–3426, 1997.
160. Ng, R. T. and J. Han, "CLARANS: A Method For Clustering Objects For Spatial Data Mining", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 14, No. 5, pp. 1003–1016, 2002.
161. Arabie, P., L. J. Hubert and G. De Soete, *Clustering and Classification*, World Scientific Publishing Company, London, 1996.
162. Zhang, T., R. Ramakrishnan and M. Livny, "BIRCH: An Efficient Data Clustering Method For Very Large Databases", *ACM SIGMOD Record*, Vol. 25, pp. 103–114, ACM, 1996.
163. Ribert, A., A. Ennaji and Y. Lecourtier, "An Incremental Hierarchical Clustering", *Proceedings of Vision Interface Conference*, pp. 586–591, 1999.
164. Lai, H. P., M. Visani, A. Boucher and J.-M. Ogier, "An Experimental Comparison Of Clustering Methods For Content-Based Indexing Of Large Image Databases", *Pattern Analysis and Applications*, Vol. 15, No. 4, pp. 345–366, 2012.
165. Utgoff, P. E., N. C. Berkman and J. A. Clouse, "Decision Tree Induction Based On Efficient Tree Restructuring", *Machine Learning*, Vol. 29, No. 1, pp. 5–44, 1997.
166. Skifstad, K. and R. Jain, "Illumination Independent Change Detection For Real World Image Sequences", *Computer Vision, Graphics, and Image Processing*,

- Vol. 46, No. 3, pp. 387 – 399, 1989.
167. Radke, R., S. Andra, O. Al-Kofahi and B. Roysam, “Image Change Detection Algorithms: A Systematic Survey”, *Image Processing, IEEE Transactions on*, Vol. 14, No. 3, pp. 294–307, 2005.
 168. Everitt, B., S. Landau and M. Leese, *Cluster Analysis*, Arnold, London, 2001.
 169. Gallant, J. L., C. E. Connor, S. Rakshit, J. W. Lewis and D. C. Essen, “Neural Responses to Polar, Hyperbolic, and Cartesian Gratings in Area V4 of the Macaque Monkey”, *Journal of Neurophysiology*, Vol. 76, pp. 2718–2739, 1996.
 170. Larson, A. M. and L. C. Loschky, “The Contributions Of Central Versus Peripheral Vision To Scene Gist Recognition”, *Journal of Vision*, Vol. 9, No. 10, pp. 1–16, 2009.
 171. Frintrop, S. and A. B. Cremers, “Visual Landmark Generation and Redetection with a Single Feature Per Frame”, *IEEE International Conference on Robotics and Automation*, pp. 1394–1399, 2010.