NTNU – Trondheim
Norwegian University of
Science and Technology

Association of Biological Localization and State-of-the-art Robot Localization

With focus on Local Environment
Representations and Hierarchies

Marianne Holmstrøm

Master of Science in Engineering Cybernetics
Submission date: June 2013
Supervisor: Øyvind Stavdahl, ITK
Co-supervisor: Sigurd Aksnes Fjerdingen, Sintef

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description: Hierarchical Methods for Localization

NTNU supervisor: Øyvind Stavdahl
Academic supervisor: Sigurd Aksnes Fjerdingen, Sintef

Localization is a key aspect for most mobile robots. Without it, intelligent motion is impossible to achieve. As a subfield of robotics, methods for robot localization has been based on traditional sensor fusion concepts from engineering disciplines such as computer science and cybernetics. Prominent examples include the extended Kalman filter. In recent years, biology literature has given revolutionary insights into how animals, and possibly humans, actually achieve localization and navigation. This research has been spearheaded by the Trondheim-based research group lead by Moser. The overarching goal of this assignment is to gain insight into these biological findings from an engineering perspective, and to investigate whether biology can teach us something new in relation to classical engineering approaches for localization. The assignment has three main goals:

1. Study relevant literature in the biology of memory and robotics, and point out similarities and differences in the way these kinds of systems represent and recognize information related to localization. Special emphasis should be put on aspects related to sensor data sequences and hierarchies.

2. Select or suggest at least one method for localization based on local environment observation, and describe its major characteristics with relevance to indoors robot localization.

3. Verify the findings from the previous point by comparison with a state of the art method in robotic localization. The comparison may be based on calculations and/or physical experiments.

Preface

Thanks. First of all, I would like to thank my supervising counsellor Sigurd Aksnes Fjerdingen, at Sintef, for great advice, guidance and reassurance. He has had a great part in making the work on my project and master theses be manageable, and even enjoyable at times. I also thank my NTNU counsellor Øyvind Stavdahl for enthusiastically recommending me to look into biological spatial representation for my master thesis - I couldn't have asked for a more interesting topic.

Espen Henriksen at the Kavli Institute for Systems Neuroscience and Centre for the Biology of Memory deserves a big thank you, for taking time in his busy schedule to show me around in their laboratories. The research they conduct there is without comparison and very inspiring. Also, thanks to Pål Liljebäck at Sintef for making the visit possible.

My father and grandfather again deserve a great thank you for enduring to proofread yet another of my reports.

I would like to thank all my friends and family who have shown me great support this past year. Specifically, I would like to thank my good friend and flatmate Karina for much needed distractions, my good friend Therese for offering to proofread, and the great group of friends I've made through my studies. Hilde, Hedvik, Ylva, Signe, Kristian and Øystein you are the best.

Remark. The software implemented for this thesis is available upon request.

Sammendrag

I mange tekniske systemer er det ultimate målet å oppnå funksjonalitet lik funksjonaliteten til biologiske system. I nyere tid har en stor mengde forskning som avdekker viktige biologiske mekanismer for lokalisering blitt publisert. I denne hovedoppgaven presenteres disse mekanismene på en systematisk måte, og de sammenlignes med mekanismer brukt til lokalisering i robotikk. Sammenligningen gjøres gjennom et omfattende litteraturstudie og visualisering av de ulike mekanismene.

I litteraturstudiet presenteres biologiske mekanismer for romlig representasjon, lokalisering og navigasjon. Funnene i litteraturstudiet inkluderer en hierarkisk organisering av romlig representasjon i likhet med hierarkiske representasjoner brukt i moderne robotikksystemer. Måten biologiske systemer representerer metriske relasjoner inkluderer blant annet celler som fyrer når dyret er på steder i miljøet som til sammen danner et triangulært grid. Denne representasjonen er ulik noen sett i robotikken. Videre presenteres den modulære organiseringen til cellene, som legger til rette for integrering av bevegelse i oppdatering av posisjon. Hvordan dyr bruker denne representasjonen til å navigere er også presentert.

Tre konsepter i den biologiske romlige representasjonen visualiseres. Geometriske egenskaper til de triangulære grid'ene undersøkes gjennom en implementasjon av et biologisk inspirert kart. Simuleringer av dette kartet viser at med en lineær økning i antall grid vil et eksponensielt større område representeres av disse grid'ene. En annen implementasjon viser de biologiske mekanismene for tracking og navigasjon. Visualiseringene av dette systemet understreker at sys-

temet ikke har en måte å takle støy på, siden man ikke vet hvordan støy håndteres i biologiske system. En videre sammenligning med et Kalman filter viser viktigheten av å inkludere et støyhånderingsverktøy dersom systemet skal brukes til å løse praktiske oppgaver.

Totalt sett introduserer denne hovedoppgaven interessante biologiske løsningsmekanismer på problemer som er viktige innenfor robotikk, og oppfordrer til videre undersøkelser og robotikkadapsjoner av disse komplekse mekanismene.

Abstract

For many technical systems the performance of biological systems is the ultimate goal. In recent years a large body of research has been presented which reveals important mechanisms used for localization in biology. In this thesis these mechanisms are systematically presented and compared to the mechanisms used in robotic localization, through a comprehensive literature exposition and visualization of different mechanisms.

The literature exposition discloses biological mechanisms used for spatial representation, localization and navigation. The findings of the exposition include a hierarchically structured spatial representation, similar to hierarchical representations used in state-of-the-art robotics systems. The way biological systems represent metric relations is disclosed, including cells which fire when the animal is at locations in the environment together forming a triangular grid. This metric representation is unlike anything seen in robotics. Furthermore, it is shown how cells are arranged in modules, which facilitates motion integration in updating the animal's location. How animals might use this modular organization for navigation is also elaborated.

Three concepts of the biological spatial representation system are visualized. Geometric properties of the triangular grids are investigated by implementing a biologically inspired map, with over one hundred grids defining a local area. Simulations of this map show that with a linear increase in grids, an exponentially larger area can be represented by these grids. Another implementation demonstrate biological mechanisms for position tracking and navigation. It is further emphasized that such a system does not include a noise handling

mechanism, because it is not known how motion and measurement noise is handled in biological systems. Simulation comparisons of the performance of the implemented system and a Kalman filter demonstrate the need for a noise handling mechanism in future systems.

All in all, this thesis introduces interesting biological mechanisms of solutions to problems important in mobile robotics, and encourages further exploration and emulation of these complex mechanisms.

Contents

1 Introduction **1**

I Robotics **5**

2 Localization in Robotics **7**
2.1 Classification of Localization Problems in Robotics . 7
2.2 Further Taxonomy 8
2.3 The Bayes Filter 9
2.4 Parametric Localization Approaches 11
2.4.1 The Kalman Filters 11
2.4.2 Parametric Methods for Localization 12
2.5 Non-Parametric Localization Approaches 13
2.5.1 The Histogram Filter 14
2.5.2 Grid Localization 14
2.5.3 The Particle Filter 15
2.5.4 Monte Carlo Localization 16
2.6 Map Representations 17
2.7 Robot Planning 18
2.7.1 Markov Decision Process and Value Iteration . 19
2.7.2 Partially Observable Markov Decision Processes 20
2.8 Chapter Summary 21

3 Hierarchical Localization **23**
3.1 Hybrid Localization Approaches 24
3.1.1 A Bottom-Up Approach 24

3.1.2 Top-Down Approaches 25
3.2 Multiple Abstraction Layers 27
3.2.1 Two Feature-Based Abstraction Layers 27
3.2.2 Metric Hierarchy based on Accuracy Needs . . 28
3.3 Chapter Summary 29

II Biology 31

4 The Brain's Spatial Rep. System 33
4.1 Place- and Grid Cells 33
4.1.1 Place Cells in the Hippocampus 34
4.1.2 Grid Cells in the Entorhinal Cortex 35
4.1.3 Local Area Localization 41
4.2 Higher Level Organization 43
4.3 Memory Formation and Navigation 43
4.4 Attractor Dynamics - Localization 46
4.4.1 Attractor Networks - Discrete or Continuous . 46
4.4.2 Attractor Dynamics Organize the Activity of Place- and Grid Cells 47
4.5 Chapter Summary 49

5 Biologically Inspired Map 51
5.1 Spatial Shift, Orientation and Spacing 52
5.2 Implementation 54
5.3 Unique Representation of Place Cells 57
5.3.1 Orientation 57
5.3.2 Spatial Shift 59
5.3.3 Room Size and Best Representation 61
5.3.4 Calculation of one Place Cell's Grid Firings . 64
5.4 Chapter Summary 66

6 Position Tracking and Navigation 69
6.1 Path Integration 69
6.1.1 Conjunctive Cells 71
6.1.2 Error Handling Mechanism 72
6.2 Sensor Fusion 74
6.3 Biological Navigation 74

6.3.1 Forward Linear Look-ahead Probing, [Erdem and Hasselmo, 2012] 75
6.3.2 Linear Look-ahead using Conjunctive Cells, [Kubie and Fenton, 2012] 79
6.4 Simple System Implementation - BioEmul 83
6.4.1 Path Integration in Grid Modules 84
6.4.2 Associating the Grid Code with Place Cells . 87
6.4.3 Correction of Grid Firings by 'Allothetic' Input 88
6.4.4 Navigation to Goal Place Cell by Probing . . 89
6.4.5 System Performance 91
6.4.6 System Discussion 95
6.5 Chapter Summary 96

III Comparisons and Implications 97

7 Comparison, Robotics and Biology 99
7.1 Spatial Representation 99
7.1.1 Maps 99
7.1.2 Hierarchies 100
7.2 Localization 101
7.2.1 Uncertainty 101
7.2.2 Global Localization and the Kidnapped Robot Problem 103
7.2.3 Position Tracking 104
7.2.4 BioEmul Versus the Kalman Filter 104
7.3 Navigation 108
7.4 Summary 108

8 Discussion 111
8.1 Part I, Robotics 111
8.2 Part II, Biology 112
8.3 Part III, Comparison 113

9 Conclusion 115

10 Future Work 117

Bibliography **119**

A The Kalman Filter Equations **125**

B Motion- and Measurement Models **127**
B.1 Motion Models 127
B.2 Measurement Models 128

Nomenclature

Acronyms

1D One Dimensional

EC Enthorinal Cortex

EKF Extended Kalman Filter

GVG generalized Voronoi graph

MCL Monte Carlo Localization

MDP Markov decision process for robot planning

MHT Multi-Hypothesis Tracking

PFC PreFrontal Cortex.

POMDP Partially Observable Markov Decision Process for robot planning

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localization And Mapping

UKF Unscented Kalman Filter

VM Visual Memory

Greek Letter Symbols

χ_t The set of particles representing the posterior at time t in a particle filter

λ Scale of a grid cell's grid

ϕ Orientation of a grid cell's grid

Latin Letter Symbols

$(\Delta x, \Delta y)$ Spatial shift of a grid cell's grid

A Total activity of a module

m map of the robot's environment, usually metric or feature based

u_t control data, change of state in the interval $(t-1;t]$

v Measurement noise

w Process noise

x_t current state at time t

z_t Measurement data sequence obtained at time t

List of Figures

2.1 Simple block diagram illustrating the generalized localization process. 9
2.2 Derivations of the Bayes filter is divided into two families; parametric and non-parametric filters. 11
2.3 Illustration of the three most popular map representations of the robot's environment 18

3.1 Graph-cut method in bottom-up hierarchical method 25
3.2 Illustration of the hierarchical atlas 26
3.3 In [Menegatti et al., 2004] the hierarchy is determined by frequency-component comparisons 29

4.1 [Moser et al., 2008]Firing field of place cell in the environment. 34
4.2 Schematic of firing fields of three grid cells. 36
4.3 [Moser et al., 2008]Firing field of grid cell in the environment. 36
4.4 Grid cells are classified by their firing fields orientation, spatial shift and scale. 37
4.5 Hierarchical organization of grid cells in the EC. . . . 38
4.6 Modular organization of grid cells in the EC. 38
4.7 The activity of the active grid cell modules drive activity of the place cells. 39
4.8 [Buzsáki and Moser, 2013] Global remapping of grid-cell modules 40
4.9 Schematic of the local-space communicational system of the brain 41

4.10 [Buzsáki and Moser, 2013] Relations between spatial memory formation and navigation 45
4.11 Discrete Attractor 46
4.12 Continuous Attractor 46
4.13 Activity of a grid module shifts with velocity. 48

5.1 First triangle of basic grid 53
5.2 Even numbers of spatial shifts cause duplicate grids . 54
5.3 Algorithm: vertex centers of one module 55
5.4 Test results of the effect of orientation discretization . 58
5.5 Test results of the effect of spatial shift discretization 60
5.6 Test results of the effect of room size 62
5.7 Size of uniquely defined room as a function of necessary grid cells. 63
5.8 From distance to reference point to grid firings 65

6.1 Grid cells with common orientation and spacing but with spatial shifts (phases) distributed in the domain, tile the environment. 70
6.2 [Giocomo et al., 2011] One-dimensional attractor dynamics. 71
6.3 [McNaughton et al., 2006] A model for path integration. 72
6.4 1D representation of three grid modules from three different layers. 73
6.5 [Erdem and Hasselmo, 2012] The prefrontal cortex layer connected to the place cells 76
6.6 The forward linear look-ahead probing allows shortcuts to be made across unfamiliar space 78
6.7 [Kubie and Fenton, 2012] Circular scatter plots of connection strength as a function of difference in heading preference of the connected cells. 80
6.8 [Kubie and Fenton, 2012] The activity centroid in the firing fields of the conjunctive cells move in the preferred heading direction. 81
6.9 Grid pattern used in the system implementation. . . 84
6.10 MATLAB Simulink implementation of Simple 1D Navigation System 85
6.11 Activity frequencies of the different grid modules. . . 91

6.12 Simulation test of the probing mechanism of the system. 92
6.13 Simulation test of the system with multiple goals. . . 93
6.14 Simulation test of the system with multiple goals. . . 93
6.15 Simulation test of the system with corrections. 94

7.1 The plant's input and measurement is sent to both the Kalman filter and the BioEmul system. 105
7.2 Comparison of simulated performance - BioEmul and the Kalman filter, with measurements arriving every 50^{th} time-step. 106
7.3 Comparison of simulated performance - BioEmul and the Kalman filter, with measurements arriving every time-step. 107

List of Tables

5.1 Test results from necessary grids to define a room of size A uniquely. 61

A.1 The discrete Kalman filter equations [Mendel, 1995]. 126

Chapter 1

Introduction

Motivation for understanding how the biological brain works is vast; simply interest in how animals operate, how certain illnesses, such as Alzheimer's and Parkinson's diseases, deteriorate the mind, and the wish to find cures for such illnesses. Another motivational factor comes from engineering; the wish to understand the functionality of the mind, which has capabilities unseen in the technological world, and use this information to improve technical solutions.

The technological motivation comes from many different research areas. One obvious motivator is the field of artificial intelligence, which aims at creating technical systems with the capacity to decide and learn on their own. An effort made in recent years, as part of the DARPA (Defense Advanced Research Projects Agency) research project *SyNAPSE* [DARPA, 2012], is to develop computers operating in a fundamentally different way than the ones seen to date. The goal is to create electronic systems inspired by the human brain, that can understand, adapt and respond to information in a way fundamentally different from traditional computers.

The field of robotics also shows interest for biological mechanisms. Focus areas of robotic research include that of spatial representation, localization and navigation. These areas are crucial in allowing robots to operate autonomously, which is the main objective of many projects. Within these areas tremendous progress has been made the

last decades, with more and more sophisticated methods emerging. However, as in all fields, performance of the systems are sought optimized. In many localization and navigation solutions, human abilities are viewed as the ultimate goal. Thus, understanding how the human brain works allows for emulation of its mechanisms, which might help the robotic field progress towards its ultimate goal in these areas.

Great advances in research of animal and human brains have been made in the last century. In particular, notions of how space is represented have come to light after the discovery of cells which fire when the animal is in a particular location. This discovery was made by O'Keefe and Dostrovsky back in 1971. Since then, more cells have been discovered which contribute in animals' spatial representations [Moser et al., 2008]. Hence, with the advances made in the understanding of the brain's spatial representation system, investigations of the application of biological mechanisms in robotics are possible.

Numerous robotic systems striving to mirror the visual processing capabilities of humans have been presented [Siagian and Itti, 2009, Ivan, 1994]. However, only a handful research projects have focused on emulating the brain's internal spatial representation system [Milford et al., 2010, Weitzenfeld et al., 2012, Jauffret et al., 2012]. Most of the research presented is confined to one or two mechanisms, and does not provide a holistic emulation. One exception is the project led by Michael Milford, ratSLAM, which has presented systematic research over the past decade [Milford et al., 2004]. However, the focus of the ratSLAM project has mostly been on replicating the biological mechanisms, without general comparisons to the state-of-the-art methods in robotics.

This study aims at showing how mechanisms of biological spatial representation, localization and navigation relate to the corresponding state-of-the-art methods in robotics. In particular, the thesis make the following contributions:

- A systematic presentation of the basic elements and mechanisms for biological spatial representation, targeting engineers

rather than biologists and medical professionals.

- Display of how the biological spatial representation is potentially used for localization and navigation, through a unified literature exposition and visualizations of the different mechanisms.

- A comparison of biological and robot mechanisms, emphasizing important aspects still undiscovered in the studies of biological localization and navigation.

The investigation is conducted by comparing the biological methods to renowned robotic methods, and inspecting the basic biological elements which manifests the advanced and to-date unreplicable biological systems. Furthermore, a simple 1D implementation emulating biological navigation processes is implemented, to enhance the understanding of the collaboration between the individual elements used for biological navigation and tracking.

This paper is organized in three parts:

Part I starts with introducing state-of-the-art methods for robot spatial representation, localization and navigation, together with a unifying taxonomy. Subsequently, hierarchical spatial representations are presented, which combine standard representations in a hierarchical manner to optimize localization performance.

Part II, chapter 4, discloses the basic elements and the organization of the brain's spatial representation system, with the organization structure showing clear parallels to the hierarchical robot localization systems. In chapter 5, the geometric and mathematical properties of the brain's spatial map are explored. In chapter 6, popular theories are given of how animals use their spatial representation system for tracking and navigation, and a simple system, BioEmul, is implemented to visualize these processes.

Part III relates the findings from the previous part to the corresponding methods and structures in robotics. It demonstrates that a

direct comparison is hard to make at this stage, because it remains unknown how biological systems represent uncertainty. A comparison of the implemented system from part II, BioEmul, and the Kalman filter visualizes the importance of representing uncertainty in localization systems. At the end, a discussion is given which underline important limitations of the conducted study.

Part I

Robotics

Chapter 2

Localization in Robotics

The most basic perceptual problem in robotics is the *generalized localization problem*. *Mobile robot localization* is an instance of this problem and can be defined as:

> *"Mobile Robot Localization is the problem of determining the pose of a robot relative to a given map of the environment."* [Thrun et al., 2005]

In this chapter an overview of the different components constituting the mobile localization problem and some renowned solutions will be presented. The different spatial representations commonly used in robotics will be introduced, and a brief description of navigation approaches will be given at the end. The majority of the material presented is collected from the book 'Probabilistic Robotics' [Thrun et al., 2005].

2.1 Classification of Localization Problems in Robotics

There are several localization problems defined in the robotics area, ranging in nature and difficulty. The problems listed below are presented in ascending order of difficulty [Thrun et al., 2005].

Position Tracking Problem Assumes that the *initial* robot pose is known. The location of the robot can be acquired by accommodating the motion noise.

Global Localization Problem The initial pose of the robot is unknown. The robot is placed somewhere in a known environment, but the exact location is unknown.

Kidnapped Robot Problem A variant of the global localization problem, only during operation the robot can get kidnapped and teleported to another location.

2.2 Further Taxonomy

Important terminology used in describing the localization solutions is now to be introduced. The taxonomy coincides with the taxonomy given in [Thrun et al., 2005].

State All aspects of the robot and its environment that can impact the future are comprised in the state vector x_t. If all variables of the state are time-invariant the state is called a *static state*, otherwise *dynamic state*.

Pose Comprises the robot's location and orientation relative to a global coordinate frame. A rigid mobile robot possesses six such state variables: three Cartesian coordinates ($x-, y-, z-$ dimensions), and three for angular orientation (roll ϕ, pitch θ, yaw ψ). In planar environments, only three variables are needed to describe the robot's pose: x, y, ψ.

The Markov Assumption Predicates that past and future data are independent if one knows the current state x_t.

Landmarks Distinct, stationary features of the environment that can be recognized reliably.

State Transition Probability Specifies how environmental state evolves over time as a function of robot controls u_t . It is given by the probability distribution $p(x_t|x_{t-1}, u_t)$.

Measurement Probability Specifies the probabilistic law according to which measurements z are generated from the environment state x. It is given by the probability distribution $p(z_t|x_t, m)$, where m is a map of the robot's environment.

Belief The robot's internal knowledge about the state of its environment. It is presented through conditional probability distributions, more specifically given by $bel(x_t) = p(x_t|z_{1:t}, u_{1:t})$.

Prediction The posterior calculations of the belief, that is prior to incorporating the current measurement z_t, is referred to as *prediction*: $\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t})$. Calculating $bel(x_t)$ from $\overline{bel}(x_t)$ is called *correction*.

Multimodal Posterior A belief distribution that has multiple local maxima, allowing more than one hypothesis to be represented by the posterior/belief.

2.3 The Bayes Filter

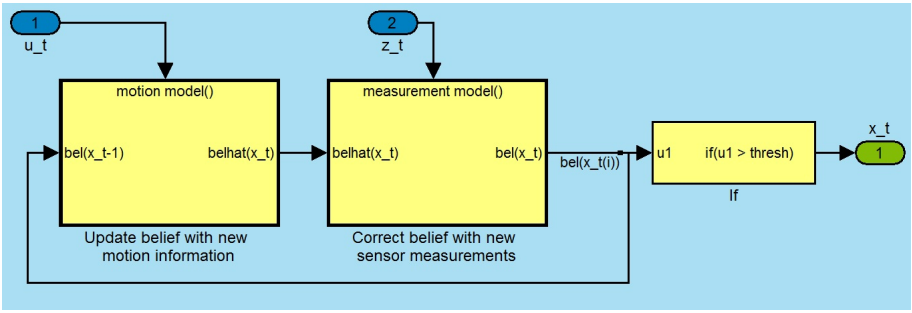

Figure 2.1: *Simple block diagram illustrating the generalized localization process. All filters commonly used in localization processes are extensions of the Bayes filter. The location of the robot is considered found if one state belief is over some design-specific threshold.*

Bayes filter is the most general algorithm for calculating beliefs, and all subsequent filters introduced are extensions of this filter (figure 2.1). In short, it calculates the belief distribution *bel* from measurement and control data. Pseudo-code for the update rule of the Bayes

algorithm is given in algorithm 2.1 [Thrun et al., 2005].

Algorithm 2.1 Update rule of the Bayes filter [Thrun et al., 2005]

procedure UPDATEBAYES($bel(x_{t-1})$, u_t, z_t)

  for all x_t **do**
    $\bar{bel}(x_t) = \int p(x_t|u_t, x_{t_1})bel(x_{t-1})dx_{t-1})$
    $bel(x_t) = \eta p(z_t|x_t)\bar{bel}(x_t)$ ▷ η is a normalization factor
  end for
  return $bel(x_t)$
end procedure

As can be seen from algorithm 2.1, the Bayes filter algorithm solves for the posterior belief recursively, and hence requires an initial belief $bel(x_0)$ at time $t = 0$ to be set. If one knows x_0 exactly, $bel(x_0)$ should be initialized as a point mass that centers all probability at state x_0. The global localization problem entails having no knowledge about the initial state x_0, in which situation $bel(x_0)$ should be initialized as a uniform distribution over the domain of x_0 [Thrun et al., 2005].

An assumption set for the Bayes filter to work is the Markov assumption, defined in the taxonomy. The main contributor to the failure of the Markov assumption is environmental dynamics not modelled in the state x_t. Hence, care should be exercised when defining the state x_t [Thrun et al., 2005].

The Bayes filter algorithm shown in algorithm 2.1 can only be implemented for very simple localization problems; the state space needs to be finite or the integration must be executed in closed form [Thrun et al., 2005]. Practical derivations of the Bayes filter are usually divided into two main families, parametric and non-parametric methods (figure 2.2). Members of the two families will now be introduced, together with the localization methods which utilize the individual derived filters.

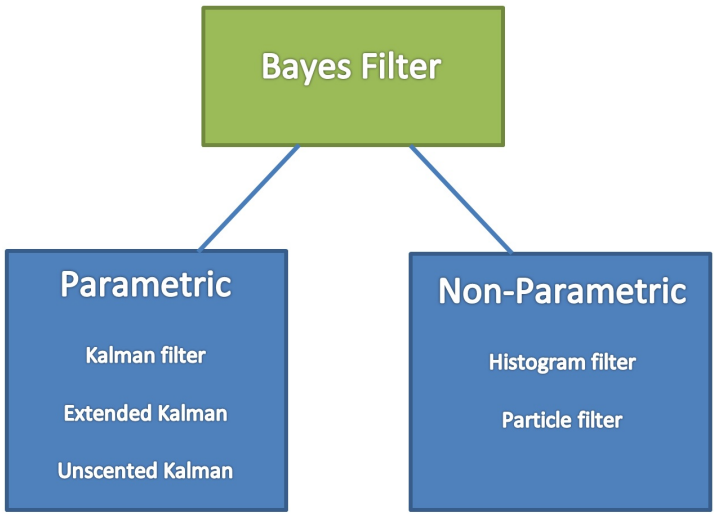

Figure 2.2: *Derivations of the Bayes filter is divided into two families; parametric and non-parametric filters. The filters given in the boxes are the ones which are presented here.*

2.4 Parametric Localization Approaches

2.4.1 The Kalman Filters

A Kalman filter is a member of the parametric family of recursive state estimators known as *Gaussian filters*. The basic idea behind all Gaussian filters is that beliefs (posteriors) are represented by multivariate normal distributions. One consequence is that only the most probable state is distinguished, because the Gaussian is a unimodal distribution. Hence, when using Gaussian filters, only single-hypothesis (intermediate-) solutions are directly possible [Thrun et al., 2005].

Expressed succinctly, the Kalman filters represent the belief $bel(x_t)$ at time t by the mean μ_t and the covariance Σ_t. To update these parameters, the Kalman filter requires the measurement z_t and the control u_t in its measurement update step and prediction step, respectively. The update step decreases and the prediction step increases

uncertainty in the robot's belief [Thrun et al., 2005].

The general Kalman filter makes certain requirements to the system and its variables in order to ensure a Gaussian distributed posterior, $bel(x_t)$. In addition to the validity of the Markov assumption, three conditions must hold: the state transition probability $p(x_t|u_t, x_{t-1})$ and the measurement probability $p(z_t|x_t)$ must be *linear* functions in their arguments, with added Gaussian noise. Furthermore the initial belief $bel(x_0)$ must be normally distributed [Thrun et al., 2005]. Equations of the discrete Kalman filter are given in appendix A.

Several extensions of the Kalman filter have been designed in order to overcome some of the imposed system requirements of the general linear Kalman filter. The *Extended Kalman filter* and the *Unscented Kalman filter* are both extensions which linearize the system equations at each step in order to apply the Kalman filter to nonlinear systems. The linearization techniques used are *Taylor expansion* and *unscented transform*, respectively. The *multi-hypothesis Kalman filter* represents the posterior by a mixture of Gaussians (weighted sum), in order to achieve multimodal posteriors with a Kalman filter [Thrun et al., 2005].

2.4.2 Parametric Methods for Localization

The process of applying Bayes filter to a localization problem is called *Markov localization*. As will be discussed in the following, the parametric methods offer enhanced computational performance compared to non-parametric methods. However, several of the parametric methods are not applicable to the global localization and kidnapped-robot problems.

EKF localization. EKF localization applies the extended Kalman filter to the localization problem, and is primarily applied to feature based maps. The choice of features to represent the environment is crucial in avoiding ambiguities while still ensuring that the features

are densely enough represented in the environment. The EKF localization algorithm is poorly applicable to global localization because it only works well if the position uncertainty is small (as in position tracking) [Thrun et al., 2005].

UKF Localization. UKF localization uses the unscented transform to linearize both motion and measurement models in the localization problem. Just like the EKF, it is less applicable to global localization problems, for similar reasons [Thrun et al., 2005].

Multi-Hypothesis Tracking. The multi-hypothesis tracking algorithm (MHT) investigates multiple correspondences, by using a Gaussian mixture to represent the posterior. Mixture components are created dynamically. Because MHT can represent multiple initial beliefs, it is applicable to the global localization problem. It may apply either EKF or UKF linearization [Thrun et al., 2005].

2.5 Non-Parametric Localization Approaches

All of the Gaussian filters rely in some way on a fixed functional form of the posterior. The filters introduced in the following do not rely on a fixed functional form, they rather approximate the posterior by a finite number of values. Because these filters do not make strong parametric assumptions on the posterior density, they are classified as *nonparametric filters*. Furthermore, they are often the filters of choice when trying to solve global localization problems because they can represent complex multimodal posteriors [Thrun et al., 2005].

Non-parametric localization methods exhibit certain characteristics which make them preferable to parametric methods in many applications;

- They can process raw sensor measurements, including negative information (free-space).

- They are not bounded to a unimodal distribution such as EKF and UKF localizers.
- They can solve global localization problems, and kidnapped robot problems in some instances.

The methods presented in the following have gained significant popularity and exhibited excellent performance in a number of problems. The first approach is called *grid localization* and uses a histogram filter to represent posterior beliefs. The second approach is the popular *Monte Carlo Localization* (MCL) algorithm , which uses particle filters to estimate posteriors [Thrun et al., 2005].

2.5.1 The Histogram Filter

Histogram filters decompose the state space into a set number of regions and represent the cumulative posterior probability of each region by a single value. When applied to finite spaces, histogram filters are known as *discrete Bayes filters* (in signal processing: the forward pass of a *hidden Markov model, HMM*). The techniques used in decomposition are either *static* or *dynamic*, that is, they either rely on a fixed decomposition irrespective of the shape of the posterior or they adapt the decomposition to the specific shape of the posterior (respectively). *Density trees* decompose the state space recursively, so as to adapt the resolution to the posterior probability mass. Another dynamic technique is known as *selective updating*, which updates only regions whose posterior probability exceeds some user-determined threshold [Thrun et al., 2005].

2.5.2 Grid Localization

The *grid localization* algorithm approximates the posterior using a histogram filter over a grid-decomposition of the state space. The set of all grid regions forms a partition of the space of all valid poses:

$$domain(\mathbf{X}_t) = \mathbf{x}_{1,t} \cup \mathbf{x}_{2,t} \cup ...\mathbf{x}_{K,t}$$

where $\mathbf{x}_k$ is a grid region. Algorithm 2.2 shows the pseudo-code for the grid localization algorithm [Thrun et al., 2005]. η is the final normalizer before obtaining the posterior collection $bel(x_t) = \{p_{k,t}\}$. For the global localization problem with no prior knowledge of the location, $\{p_{k,0}\}$ is initialized as a uniform distribution over the K grid regions. As motion and sensor data is incorporated with time, the corresponding grid regions' probability values are raised. Motion and measurement models are presented in appendix B for the interested reader.

Algorithm 2.2 Grid Localization [Thrun et al., 2005]

1: **procedure** GRIDLOCALIZATION($\{p_{k,t-1}\}$, u_t, z_t, m)
2:
3: **for all** k **do**
4: $\bar{p}_{k,t} = \sum_i p_{i,t-1}$**motionModel**$(mean(\mathbf{x}_k), u_t, mean(\mathbf{x}_i))$
5: $p_{k,t} = \eta$**measurementModel**$(z_t, mean(\mathbf{x}_k), m)$
6: **end for**
7: **return** $\{p_{k,t}\}$
8: **end procedure**

2.5.3 The Particle Filter

The *particle filter* is a version of the Bayes filter which also approximate the posterior by a finite number of parameters. Contrarily to the histogram filter, the particle filter does not sample the state space in a certain pattern: the posterior is represented by a set of *random* state samples drawn from the posterior. The representation of the posterior becomes an approximation, but it can represent a broad space of distributions and model nonlinear transformations in random variables, due to its nonparametric nature.

The samples of a posterior distribution are called *particles* and are denoted:

$$\chi_t := x_t^{[1]}, x_t^{[2]}, ..., x_t^{[M]} \tag{2.1}$$

Each particle represents a hypothesis as to what the true world state may be at time t. Each particle is given a weight or *importance factor* ($w_t^{[m]}$), to correctly represent the Bayes filter posterior $bel(x_t)$. The most important attribute of the particle filter is the resampling procedure. The resampling is done based on the new weight distribution among the particles [Thrun et al., 2005].

2.5.4 Monte Carlo Localization

Monte Carlo Localization is a renowned localization method, which represents the belief $bel(x_t)$ by particles. The method is easy to implement and tends to work well across a variety of localization problems.

Algorithm 2.3 shows the pseudo-code for the MCL algorithm. When the initial position of the robot is unknown, particles are drawn at random from the entire pose set and form the initial particle set χ_0. Each particle's state, together with odometry data, is used to generate a hypothetical state $x_t^{[m]}$ in the *sampleMotionModel* algorithm. As sensing data z_t is incorporated, *importance weights* $w_t^{[m]}$ are given to the new states depending on how the states coincide with the sensed information, using the *measurementModel* (line 6). The set $\bar{\chi}_t$ is then updated with the locations and weights of each new particle. In line 10 of algorithm 2.3 the real trick to the method begins - the resampling procedure. A new set of particles, χ_t are drawn from the temporary set $\bar{\chi}_t$, *with* replacement. This way, important (probable) particles are strongly represented in the new set (duplicates occur as the probability increases) and less important particles are not represented in the set [Thrun et al., 2005].

The Monte Carlo localization algorithm can approximate almost any distribution of practical importance. Increasing the total number of particles increases the accuracy of the approximation, while increasing computational complexity.

Algorithm 2.3 Monte Carlo Localization [Thrun et al., 2005]

1: **procedure** MCL(χ_{t-1}, u_t, z_t, m)
2: $\bar{\chi}_t = \chi_t = \emptyset$
3:
4: **for** $m = 1$ **to** M **do**
5: $x_t^{[m]} =$ **sampleMotionModel**(u_t, x_{t-1}^m)
6: $w_t^{[m]} =$ **measurementModel**($z_t, x_t^{[m]}, m$)
7: $\bar{\chi}_t = \bar{\chi}_t + (x_t^{[m]}, w_t^{[m]})$
8: **end for**
9:
10: **for** $m = 1$ **to** M **do**
11: draw i with probability $\propto w_t^{[i]}$
12: add $x_t^{[i]}$ to χ_t
13: **end for**
14: **return** χ_t
15: **end procedure**

2.6 Map Representations

A very important factor in the localization scheme is the choice of space representation. The type of map used has a great effect on computational cost, memory requirements and the accuracy (success) of the localization procedure. Localization methods are often categorized by the type of map used, as will become clear later.

The robotics literature usually distinguishes three representations of space: *topological*, *feature-based* and *grid-based* [Lisien et al., 2005]. Illustrations of these three representations are given in figure 2.3. Topological representations are usually thought of as coarse graph-like representations, where the nodes correspond to significant places in the environment (*landmarks*, e.g. T-junctions, doors, etc.). Because the number of landmarks present depends on the nature of the environment, the resolution of the map is heavily dependent on the structure of the environment. A feature-based map, which represents the environment by some user-determined features, encounter the same structural dependencies, only the feature-map is usually much

denser than the topological node-map. A metric grid-based map contrarily decompose the state space using regularly-spaced grids, allowing the resolution to be set by the designer. Typically the spatial resolution of grid-representations tends to be higher than of topological and feature-based representations. Obviously the increased accuracy of the metric grid-map comes at the expense of increased computational complexity [Thrun et al., 2005, Rady et al., 2010].

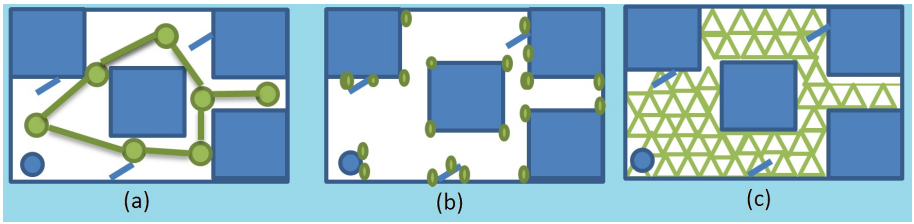

Figure 2.3: *The three main map types. (a) Topological representations, which only mark specific landmarks as locations. (b) feature-based maps represent the location based on the distance to salient features in the environment. (c) Grid-based representations decompose the space into regularly-spaced grids of desired size.*

The grid-based or *location-based* maps are *volumetric* and hence contain information not only about the location of objects, but also about the absence of objects (e.g. free-space). Contrarily, topological maps only specify the shape of the environment at different locations.

The *occupancy grid map* is a popular version of the metric grid-based map; for each x-y coordinate it specifies a binary occupancy value indicating whether the location is occupied or not. Such occupancy grid maps are well suited for robot navigation in that they make it easy to find the shortest path through the unoccupied space [Thrun et al., 2005].

2.7 Robot Planning

In the previous sections, methods for robots' perception of their location in the environment were presented. However, the ultimate goal

of many robotics systems is to enable the robot to choose the right action. In this section robot planning paradigms will be presented, which encompass the uncertainty in robot motion and perception.

2.7.1 Markov Decision Process and Value Iteration

Markov decision processes (MDPs) is a robot planning paradigm which encompasses uncertainty in robot motion. MDPs assume that the state of the environment can be fully sensed at all times ($p(z|x)$ is deterministic), while action models $p(x'|u, x)$ can be stochastic. When the action model is stochastic, it is insufficient to plan a single sequence of actions; the planner has to generate actions for a range of situations the robot might encounter because of unpredictable dynamics. Actions are defined for all states through a *control policy*.

Value iteration is an algorithm for finding control policies when the state is fully observable. It incorporates the goal configuration and the cost variables in a single function called the *payoff function*, r. An example payoff function is:

$$r(x,u) = \begin{cases} +100 & \text{if } u \text{ leads to a goal configuration or state} \\ -1 & \text{otherwise} \end{cases} \tag{2.2}$$

This payoff function rewards the robot with $+100$ if a goal configuration is reached, while it penalizes the robot by -1 whenever it has not reached that configuration. This way, the payoff function incorporates the trade-off between goal achievement and cost along the way. The control policy, denoted π, is a program which optimize future payoff in expectation. Every policy has an associated *value function*, which measures the expected value of the specific policy. For the greedy case (when only the immediate next payoff is seeked minimized) , the value function is:

$$V(x) = \gamma \max_u r(x,u) \tag{2.3}$$

where γ is a discount factor.

Value iteration is the algorithm which calculates the value function of the optimal policy. For the general case, the approximated value function $\hat{V}$ is set to r_{min}, the minimum possible immediate payoff. Value iteration then successfully updates the approximation via the following recursive rule:

$$\hat{V}(x) \leftarrow \gamma \max_u \left[r(x,u) + \int \hat{V}(x') p(x'|u,x) dx' \right] \tag{2.4}$$

For a finite state space the integral can be implemented as a finite sum over all states, and computed efficiently since $p(x'|u,x)$ is usually non-zero for relatively few states x and x' [Thrun, 2008].

2.7.2 Partially Observable Markov Decision Processes

Partially observable Markov decision processes (POMDPs) are extensions of MDP to problems where the state is not fully observable from the measurement z, it can only be estimated up to a certain degree. Thus, the robot has to estimate a posterior distribution over possible world states, called the *belief* state, denoted b. POMDPs compute a value function (equation 2.4) over belief space:

$$V_T(b) = \gamma \max_u \left[r(b,u) + \int \hat{V}_{T-1}(b') p(b'|u,b) db' \right] \tag{2.5}$$

with $V_1(b) = \arg\max_u E_x[(x,u)]$. Direct computation of this value function is complicated, due to the complex nature of the belief space. Luckily, for finite worlds, in which the state space, the action space, the space of observations and the planning horizon are all finite, an exact solution exists. The solution represent value functions by *piecewise linear functions* over the belief space. The linearity is a consequence of the fact that expectation is a linear operator. The piecewise nature is a result of the fact that the robot has the ability to select controls, and in different parts of the belief space it will select different controls [Thrun, 2008]. Examples and extensions of POMDP are found in [Thrun, 2008].

2.8 Chapter Summary

This chapter serves as the foundation for later comparison between current robotics systems and biological systems for localization and planning. In the beginning of the chapter the different localization problems in robotics were defined, together with a listing of robot localization taxonomy. Further, the Bayes filter together with its parametric and non-parametric extensions were introduced, followed by their application in localization. The different map representations commonly used in robotics were then presented. At the end, a brief introduction to robot planning was given. In the next chapter the focus will be turned to hierarchical solutions to robot localization; how representing an environment in several abstraction layers facilitates enhanced localization performance.

Chapter 3

Hierarchical Localization

One of the localization procedures presented thus far together with a suitable map representation constitute a localization method. The methods are usually classified into three categories; feature-based, grid-based and topological, each of which has its advantages and disadvantages. Metric grid-maps usually have a higher spatial resolution, which provides accurate metric pose estimation, shadowed by extra overhead in storage, computation and maintenance requirements [Rady et al., 2010]. Further, grid-based approaches are very dependent on a good *motion model*, to account for drift in the odometry. Topological approaches handle this problem in that only global topological consistency is needed, not metric. Topological approaches usually provide a coarser decomposition of the environment, resulting in the inability to localize arbitrarily in the environment (only nodes in the topological graph), and to disambiguate similar topological regions. For topological approaches to achieve the same precision as metric approaches, would require computational efforts not currently applicable for fully autonomous robots. Feature-based approaches extract distinct landmark features from the environment, but do not address obstacles unless they have structured and observable characteristics [Lisien et al., 2005, Tomatis et al., 2003].

Hybrid approaches to localization problems have shown that positive characteristics of all of the above discussed approaches can be integrated to compensate for each of their weaknesses [Tomatis et al., 2003].

The typical way to represent a hybrid-hierarchical approach is with a high level topological map, representing landmarks as nodes, and lower-level feature- or grid-based maps associated with each node for exact location tuning. The main motivation for such hierarchical approaches is to reduce the set of location candidates toward a solution, and hence, the costly computations of the grid- or feature-based map are evaluated only on a minimal subset of the environment [Rady et al., 2010]. Some examples of different hierarchical approaches found in the literature, will now be presented.

3.1 Combinations of Representations - Hybrid Localization Approaches

Combinations of topological and either feature- or grid-based representations are generated in a top-down or a bottom-up manner. In the top-down case, the global topological map is initially constructed, and further linked to a set of detailed local grid- or feature-based maps for each node, as introduced in the preceding paragraph. Contrarily, in the bottom-up case, a high resolution map is initially constructed, from which a topological map is generated by the use of some splitting measure [Rady et al., 2010].

3.1.1 A Bottom-Up Approach

A bottom-up approach was introduced by Blanco et. al., based on spectral graph theory for grouping together robot observations. The bottom layer is some kind of metric map structure. The space sensed in each observation is considered a node of a higher-level graph whose edges represent the *sensed-space overlap* (see figure 3.1 [Blanco et al., 2006]) between two observations. The partitioning of this *observation graph* is done through recursive-, minimum-, normalized graph-cuts. One important motivation for this approach was that the popular top-down approaches do not provide a mathematically founded solution [Blanco et al., 2006].

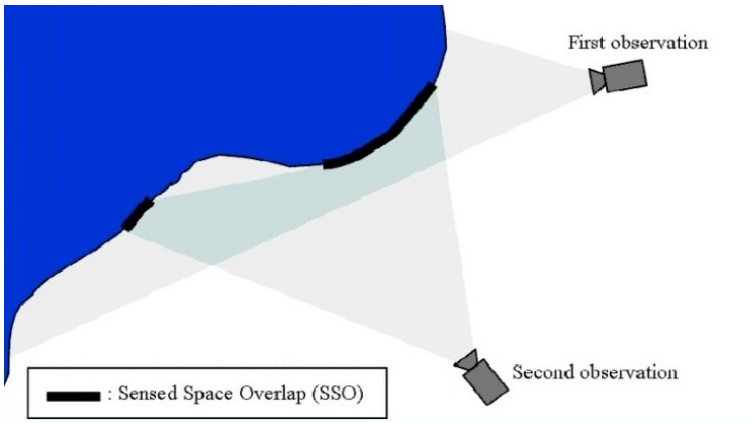

Figure 3.1: *[Blanco et al., 2006] The sensed-space overlap is the common part of the environment captured by two observations.*

3.1.2 Top-Down Approaches

The top-down approach is as mentioned the most popular approach in hierarchical localization, and several sub-approaches exist. One popular approach is using the *hierarchical atlas* [Lisien et al., 2005], which combines a topological high-level map with a lower-level feature-based map. The topological map decomposes the space into regions, in which a feature-based of *moderate* complexity is built. In the hierarchical atlas, the topological graph is based on the *Generalized Voronoi Graph* (GVG) , which offers the ability to plan paths and safely navigate in the presence of obstacles. Nodes of the GVG are either *meet points*, the set of points equidistant to three or more obstacles or *boundary points*, where the distance between two obstacles equals zero. In the hierarchical atlas, a reduced version of the GVG is used, accordingly named *reduced* GVG (RGVG). In the RGVG, the *boundary edges*, the edges that terminate at an obstacle in a boundary point (corresponding to door jambs and corners), are eliminated, motivated by their potentially unstable nature [Lisien et al., 2005].

Each submap of the hierarchical atlas is an *edge map*, a local map

of one edge, referenced from one meet point (node) toward another. Consequently, for each edge in the RGVG, there are two edge maps. Since the nodes of the RGVG serve as the origins of the edge maps, the local maps are tied both to the topological map and to the free space, see figure 3.2 [Lisien et al., 2005].

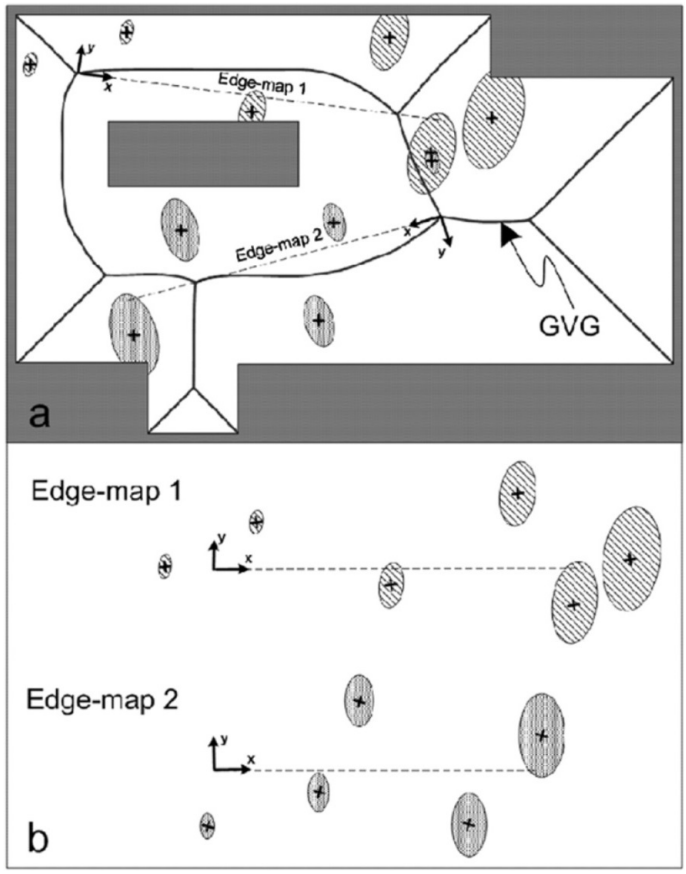

Figure 3.2: *[Lisien et al., 2005] (a) Placement and orientation of edge-map frames are determined by meet points in the GVG. (b) Edge maps are stored as individual, abstract structures.*

In [Tully et al., 2007], the hierarchical atlas has been used in a hybrid localization scheme. Their framework is set up to accommodate both global localization and kidnapped robot localization. In the

localization process, a discrete Bayes filter (histogram filter applied to discrete map) is used for localization in the topological map, and Kalman filters are used to locate the robot within each feature-based metric edge map. The single-hypothesis nature of the Kalman filter suffices, because within each edge-map, only one location is plausible; due to initialization to the edge map origin after each discrete topological transformation between edge maps, the metric localization is reduced to a tracking problem [Tully et al., 2007].

3.2 Multiple Abstraction Layers in Feature-Based Representations

The hybrid approaches presented above, are by far the most popular hierarchical approaches for localization in mobile robotics. Nevertheless, another group of hierarchical methods has evolved, with some desirable characteristics. Belonging to this group are feature- or grid-based representations with multiple abstraction layers constituting the hierarchy [Rady et al., 2010].

3.2.1 Two Feature-Based Abstraction Layers

In [Courbon et al., 2008], two abstraction layers of the feature-based method are employed. The prerecorded map of the environment is represented by a *visual memory (VM)*, a database of omnidirectional images. During the localization process, when a test image is recorded, it is first compared to the images in the database by means of *global descriptors*. Global descriptors represent each image by a single descriptor, which allows a coarse but fast matching process. A set of images from the database are extracted as possible candidates. The images in the set are next compared to the test image on a more local level. *Local descriptors* concentrated around visual features in the images are compared, distinguishing the most likely candidate from the set. The process of comparing local descriptors is more accurate than that of global descriptors, at the cost of increased

computational complexity. Because this complex comparison is only done on a small subset of the visual memory of the robot, the high accuracy can be obtained without substantial computational costs [Courbon et al., 2008].

Wang et al. presented a similar two-layer hierarchy structure, with one coarse- and one fine-localization layer. In their research, *Scale invariant feature transform (SIFT)* descriptors are used for both levels, only in the coarser level the features are matched with a *visual vocabulary* built from the SIFT descriptors of the reference images, using the k-means algorithm [Wang et al., 2005].

3.2.2 Metric Hierarchy based on Accuracy Needs

In [Menegatti et al., 2004] the hierarchical set-up is altered to meet a whole other set of requirements. Here, it is not the benefits of the hierarchical structure in finding the robot's exact location that serves as motivation. It is rather the question of whether the exact location is necessary knowledge. In certain situations the answer surely is yes, in others, such as when the robot is traversing a hallway on its way somewhere, the need for exact location is less evident.

The way [Menegatti et al., 2004] achieves hierarchy in location accuracy, is by representing the environment by the Fourier-coefficients of omnidirectional images, taken in a discrete metric grid pattern across the environment. The Fourier-coefficients of reference images in the memory are compared to the Fourier-coefficients of the test image. The number of coefficients compared is the basis of the hierarchical solution: starting with the lowest-frequency coefficients the comparison is made up to some designer-determined frequency. The increase in frequency is proportional to the level of detail the comparison is made at. Hence, comparisons at a low frequency distinguishes an area of the environment in which the robot is likely to be located; comparing the next pair of frequency components decreases the size of this location area, see figure 3.3. In other words, by assigning different values to the designer-determined frequency threshold in different situations, the level of locational accuracy needed in that situation is

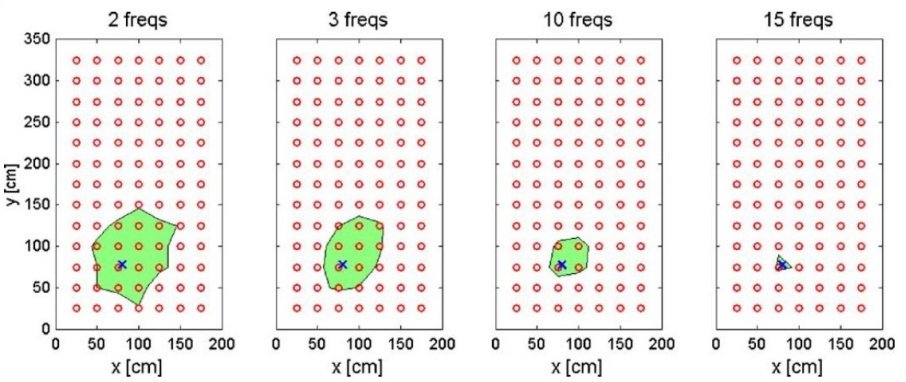

Figure 3.3: *[Menegatti et al., 2004]. Illustration of the hierarchical localization scheme in [Menegatti et al., 2004]. The red circles represent the reference images, the blue X the exact location of the robot, and the green area the set of possible locations of the robot. The number of Fourier coefficients compared increases from left to right.*

achieved [Menegatti et al., 2004].

3.3 Chapter Summary

In this chapter the different hierarchical methods for representing space in robotics have been presented. The motivation for representing the environment hierarchically is that shortcomings of the different map-types are compensated for. Representing an environment by a higher-level topological map and lower-level feature- or grid-based maps was presented as the most popular representation. The hierarchical atlas is a celebrated version of hybrid approaches. Such hybrid hierarchical maps are created in either a top-down or a bottom-up manner. Lastly, hierarchical methods using different abstraction levels of the same map-type were presented.

The attention will now be turned to spatial representation in biology, which shows similarities to the hierarchical approaches presented.

Part II

Biology

Chapter 4

The Brain's Spatial Representation System

In this chapter, the building blocks of animals' spatial perception system will be presented. First, place- and grid cells will be introduced, which together represent a map of the local environment. The manifestation of these local maps in a global frame will be briefly discussed, as the focus of this thesis will be on localization within a planar local map. The connection between spatial representation and memory is next discussed. Towards the end of the chapter, a popular method for explaining cell activity and activity pattern formation, *attractor dynamics*, will be presented.

4.1 Place- and Grid Cells

About forty years ago, neurons in the *hippocampus* were found to react on visual stimuli. This was the initial experimental breakthrough in understanding how our brain represents space internally and how it approaches the tasks of localization and navigation [Moser et al., 2008]. In 2004, Fyhn et al. (a research team at the Centre for The Biology of Memory, NTNU, led by Edvard and May-britt Moser) showed that neurons in the *Enthorinal Cortex (EC)* (of a rat) are also involved in the localization and navigation processes [Fyhn et al., 2004]. Further investigations on the role of the entorhinal cortex in representation of space have been made in the subsequent years. The research team

discovered that the cells in the EC represent the environment through a triangular-grid map, which is not abstracted from visual input as the place cells in the HC [Hafting et al., 2005]. The latest discovery made by the Moser team is the discrete nature of the grid-maps; space is represented by a map of multiple layers, each with a grid of unique spatial resolution [Stensola et al., 2012].

The essence of the preceding paragraph is that both place cells and grid cells are part of mammals' computation of self-localization. More cell types are likely to contribute: known contributors are the head-direction cells, located among other places in the EC. These cell types are well understood to fire when the mammal is facing a certain direction [Solstad et al., 2008], and will not be discussed further in the following. Note that place cells and grid cells are mechanisms used for *local*-environment localization, that is localization in the vicinity of the animal, such as within a room. Details on place cells, grid cells and their apparent cooperation are presented next.

4.1.1 Place Cells in the Hippocampus

Already in the early 1970's O'Keefe and Dostrovsky discovered neurons in the hippocampus, which exhibit high firing rates when the mammal is located at a particular location in the environment [Moser et al., 2008, Jauffret et al., 2012]. The place in the environment where a cell exhibit high firing rates is called the *firing field* of that cell. These location dependent neurons are known as *place cells* (see figure 4.1).

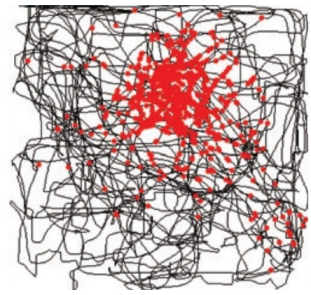

Figure 4.1: *[Moser et al., 2008] Black: A rat's trajectory in an environment. Red: One place cell's firing field.*

The entire local environment is represented by a local place cell pop-

ulation in the hippocampus. When a new local environment is encountered, a different set of place cells is used to represent the new environment. The process of going from one set of place cells to another is known as *remapping* [Moser et al., 2008].

A rat's firing field is anchored to the environment, that is, a place cell will fire at a specific location in the environment independent of the direction of which it entered the place field (certain exceptions exist: when a hallway/narrow pathway is traversed, firing fields are dependent on which direction the rat is moving). Furthermore, extrinsic cues (such as geometric boundaries) play an important role in defining the firing location of a place cell. If the cues are changed, the firing location is moved, or not even present. However, when certain salient landmarks have been removed from a familiar environment, place cells in rats continue to fire in the original locations [Moser et al., 2008]. The attractor dynamics presented in a later section explains this phenomenon.

4.1.2 Grid Cells in the Entorhinal Cortex

The more recent discovery of grid cells in the entorhinal cortex (2005), showed that the spatial navigation system of mammals is not confined to the hippocampus. Contrarily to place cells in the HC, *one* neuron (grid cell) in the EC was found to respond to several locations in the environment, together forming a triangular grid spanning the traversable area (see figures 4.3 and 4.2). Grid cells located in the same part of the EC were found to have similar spacing (grid resolution) and orientation (tilt relative to an external reference axis), but different scale (distance between grid vertices) [Moser et al., 2008], see figure 4.4.

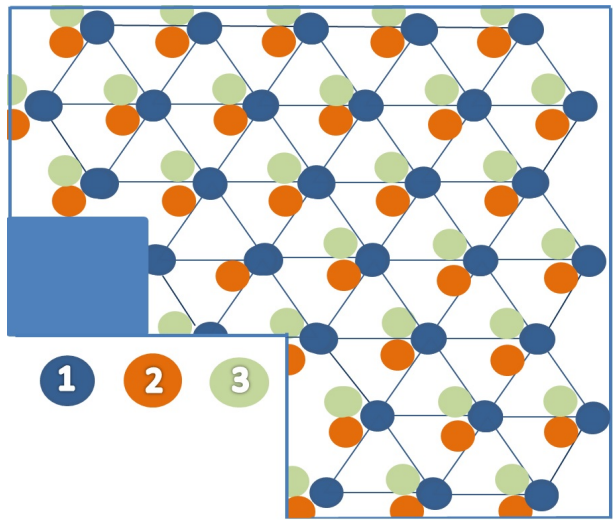

Figure 4.2: *Schematic of firing fields of three grid cells. Cell 1, 2 and 3 all have firing fields in the environment which form triangular grids. Notice that objects which block the environment surface (blue box) does not have firing fields covering it, because it is un-traversable. Objects and boundaries are represented by boundary cells, which are co-localized with the grid cells in the EC [Moser et al., 2008].*

The main contributors to the update of grid cell firing are intrinsic variables such as translational and angular velocity. Geometric boundaries in the environment anchor the grid map, while self motion maintains and updates the grid representation. The anchoring of the grid map is thought to be done through communication between grid cells and cells known as *border cells*, which are co-localized in the EC. The theory is that border cells assist grid cells in assessing allocentric distances by triangulation. Integrating self motion in order to keep track of the change in position is known as

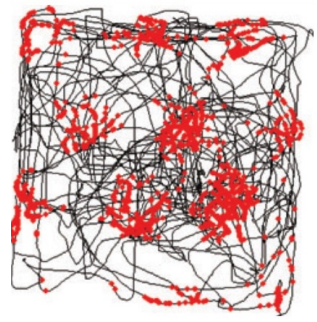

Figure 4.3:
[Moser et al., 2008] Black: a rat's trajectory in an environment. Red: One grid cell's firing fields, which form a periodic triangular grid tiling the entire environment available to the rat.

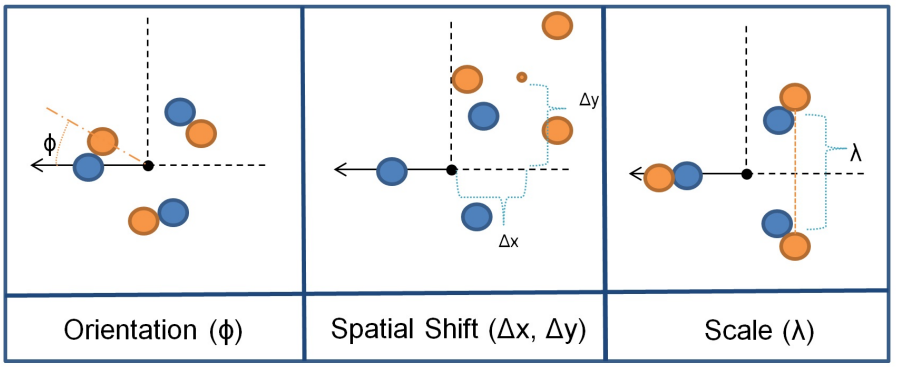

Figure 4.4: *Grid cells are classified by their firing fields orientation, spatial shift and scale, which are defined as displayed here (the blue firing fields representing the basic triangle, corresponding to zero spatial shift* $(\Delta x, \Delta y) = (0, 0)$ *and zero angle* $\phi = 0$*). All grid cell firing fields are defined with reference to a specific point and a direction.*

path integration, a process important to both place cell and grid cell firing. The area in the brain where this process is executed remains unknown, although recent discoveries suggests that the path integration process is deeply connected to the grid cell network [Moser et al., 2008, Buzsáki and Moser, 2013].

The latest discovery unveils the hierarchical organization of grid cells in the EC. In [Stensola et al., 2012], they show to recordings from up to 186 grid cells in individual rats that grid cells are organized in a network of anatomically overlapping layers, each with its distinct scale (see figure 4.5). The maximum number of layers found in a rat was 5, while more layers are thought to exist (< 10 in total), considering only 50% of the range of the medial EC was investigated. Possibly the most intriguing discovery is the ratio between successive layer averages, which fluctuated around a constant value of 1.42 ($\approx \sqrt{2}, s.d. = 0.02$). The ratio translates to a near-perfect doubling of the area of the grid triangles between layers of successive grid scale [Stensola et al., 2012]. The literature investigated does not disclose possible reasons why this ratio-specific organization has emerged.

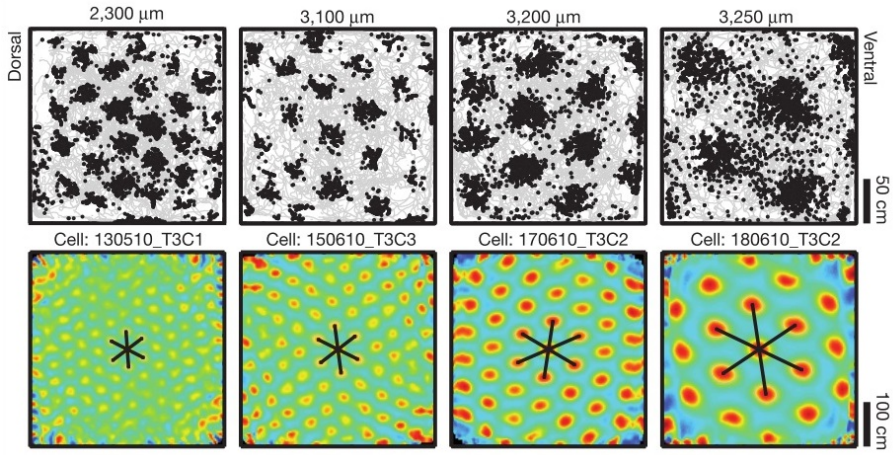

Figure 4.5: *[Stensola et al., 2012] Recordings of the firing patterns of four grid cells belonging to four different grid-modules (the numbers at the top is the position of the grid cell in the medial-EC). The ratio between grid-edge lengths of any two successive modules is approximately equal to* $\sqrt{2}$ *[Stensola et al., 2012].*

In addition to the discovery of the layered structure of the grid cell network, Stensola et al. showed that each layer is further divided into sub-modules, based on grid orientations. That is, grid cells in one module share common orientation and scale, see figure 4.6. This allows each module to tile an environment with grid cell firings, and facilitates the notion that path integration is done independently for each module [Stensola et al., 2012]. An elaborated presentation of path integration and the tiling property of such modules is given in

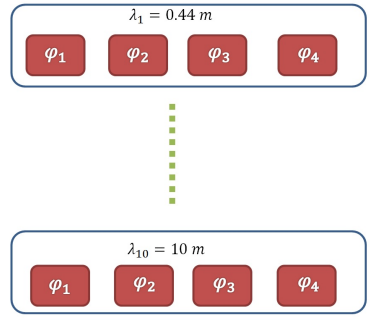

Figure 4.6: *Modular organization of grid cells in the EC. Within each layer, given by the spatial scale* λ*, the grid cells are further divided into modules based on their orientation preference* ϕ *[Stensola et al., 2012].*

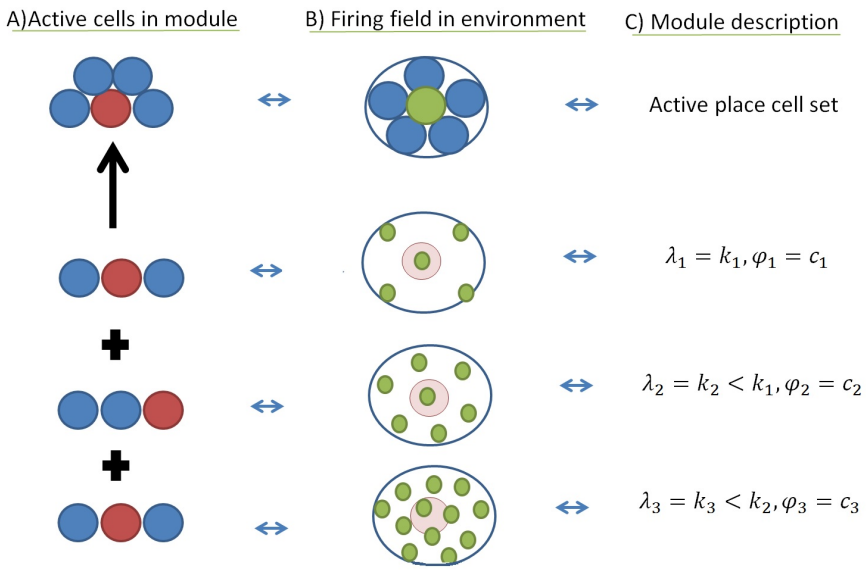

Figure 4.7: *The activity of the active grid cell modules drive activity of the place cells. The figure illustrates a situation where three grid cell modules are active, each with three cells in the module. These modules drive the activity of the active place cell set, which includes five place cells. A) the cells in the different modules, the red cell indicates highest activity, i.e. where the activity bump is. The activity bump of the place cell set is generated from the summation of the activity in the grid modules. B) The green circles illustrates the firing field of the most active cell in the circular environment. Observe that the most active cells in the grid modules have firing fields within the place cell firing field. C) Description of the modules. The grid cell modules comes from three different layers, and each module has a constant orientation [Moser et al., 2008].*

chapter 6.

A popular theory proposes that place cell firing fields are extracted from overlapping grid firing fields, that is, the activity bump in each of the active grid modules are summed to form the place field. With respect to neuronal connections between place cells and grid cells, this entails that the different constellation of activ-

ity bumps across the ensemble of modules drive activity of different place cells [Moser et al., 2008], see figure 4.7. The place cells also receive exothetic (external) information such as environmental features extracted from visual input, which is used together with the grid firings to determine which place cell should fire. The weighing of the different inputs are discussed in chapter 6. Additionally, the place cells are thought to send correction information to the grid cell network, which corrects the accumulated error in self-motion measurements [Samu et al., 2009].

Remapping of grid cells does not change the active set. For place cells, when remapping from one local environment to another is performed, a new set of place cells represent the new environment. Contrarily, for grid cells the same set is active in both environments, only the position of the reference point and axis are changed, so the orientation of the firing field of a grid cell may be different in the new environment (relative to the old environment), see figure 4.8.

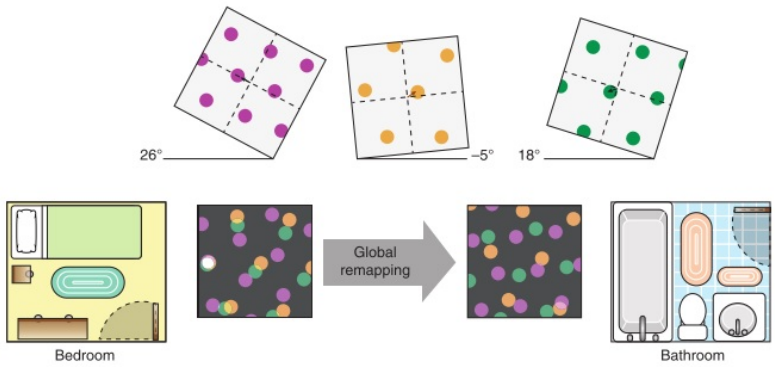

Figure 4.8: *[Buzsáki and Moser, 2013] Global remapping is done through independent changes in grid-cell firings between modules. (Top) The translational and rotational remapping of three grid cells belonging to different modules are independent. (Bottom) The white circle in the left grid-firing illustration indicates the current place cell firing in a bedroom, extracted from overlapping grid module firings. After global remapping to the bathroom environment the relative firing of the grid cells is perturbed, and another set of place cells span the local environment.*

4.1.3 Local Area Localization

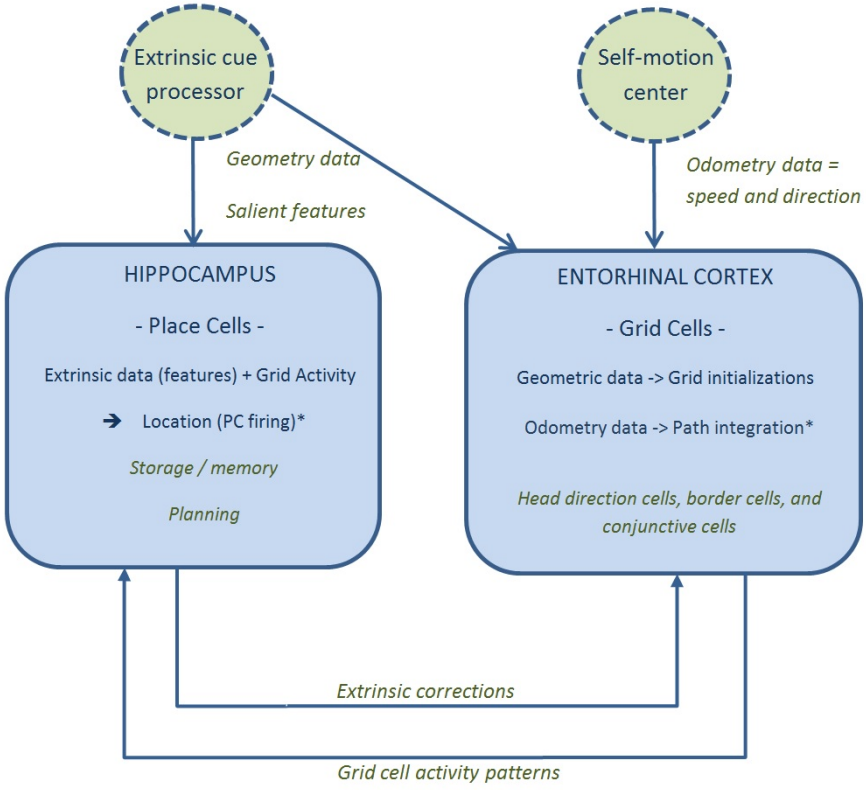

Figure 4.9: *Schematic of the local-space communication system of the mammalian brain. The solid boxes are the ones discussed in this section, hence they are the ones specified with tasks. Head-direction cells can be thought of as placed in the 'self-motion center' box, while border cells decode the geometric data arriving in the EC. Similarly, visual cells decode the information arriving to the hippocampus from the extrinsic module. The asterisks indicate which functionality is based only on a theory, not verified by experiments. Furthermore, more connections between boxes are likely to exist, and more functionalities are known to exist, than those present in the figure. Conjunctive cells, cells in the EC important for path integration, are presented in detail in chapter 6.*

Several dependencies between the grid cells in the EC and the place cells in the hippocampus have been discussed previously, although most are only assumptions. Most intra-module communicational properties of the local-space representation system of the brain remains undisclosed to us, even with the discovery of the modular organization of grid cells. In figure 4.9 a crude schematic of the brains local-space representation is displayed. The properties marked with an asterisk ($*$), are drawn from popular theories given in [Stensola et al., 2012, Moser et al., 2008].

In summary, the local-area localization process of the brain is thought to include cells with the following properties;

Grid cells The grid cells are modularly organized, and the place in the environment where the cells with highest level of activity (activity bump) in each active module fire simultaneously indicates which place cell should fire, i.e. where the animal is located in the local environment. Anchoring the grid-map is done through geometric boundaries processed by boundary cells, while updating of grid cell firing is done by interpreting self motion [Buzsáki and Moser, 2013].

Place cells The hippocampus receives grid-cell information from the EC indicating which place cell should fire. Simultaneously it receives visual information about landmarks in the environment. The availability of external landmarks is thought to influence which of these inputs dominate in the place cell firing [Buzsáki and Moser, 2013]. The place cells send information back to the grid cells with correction data extracted from more reliable visual input, which causes the activity constellation to shift to the corrected location [Samu et al., 2009].

Border cells Border cells in the EC anchor the grid map in the environment by assessing geometric information extracted from visual input [Moser et al., 2008].

Head-direction cells Cells co-located with grid- and border-cells in the EC, which fire when the animal is facing a certain direction [Solstad et al., 2008].

Another type of cells located in the EC, *conjunctive cells*, are thought to play an important role in the path integration process [McNaughton et al., 2006]:

Conjunctive cells Show firing fields in a similar pattern as grid cells, but the firing fields are only present when the animal is facing a certain direction [McNaughton et al., 2006]. Conjunctive cells will be presented in more detail in chapter 6.

4.2 Distributed Higher-level Organization of Space Representation.

Global remapping of local-space representation in new environments (see figure 4.8), together with chunking of travel paths suggests a distributed higher level organization of space. Grid representations in the EC and place firings in the hippocampus are globally remapped when the animal enters a new local environment. Furthermore, representations of travel paths do not consist of long uninterrupted neuronal chains, but are often broken down into repeating chunks by prominent landmarks, state changes or reinforcers. Similar segments of a travel path are indicated by the same sequential firing of hippocampus and EC neurons, although distinct set of neurons fire during the different path segments. These findings suggest a higher-order distributed organization of the brain's space representation, where nodes are categorized by landmarks, state changes or reinforcers, and nodes with similar external categorization (corners, hallways, etc.) show similar neuronal firing sequences. The vicinities of the nodes are represented by unique local area place- and grid-cell firings [Buzsáki and Moser, 2013].

4.3 Memory Formations' connections to activity in the HC and EC.

As previously mentioned, the modularity of the grid cells allows for an abundance of combinations, even if there only were a small set of

cells representing each module. This ability of the local-space representation together with the higher-level distributed organization draws clear parallels to the organization of spatial memories, indicating exploitation of the same mechanisms in both navigation and spatial memory [Buzsáki and Moser, 2013].

Two forms of declarative memory can be distinguished. *Semantic memory* explicitly defines living things, object facts and events of the world without reference to time. *Episodic memory*, on the other hand, empowers individuals with the ability to learn and recall first-person experiences in the context of both space and time, which is useful for planning actions [Buzsáki and Moser, 2013].

In [Buzsáki and Moser, 2013] it is argued that the evolutionary roots of episodic and semantic spatial memory systems are path integration and landmark-based forms of navigation, respectively (figure 4.10 [Buzsáki and Moser, 2013]). The episodic memory link together events similar to the way location sequences are linked together in a path integrator. Complimentary, the objectivity of the semantic memory resembles the way *allocentric maps* define a location independent of how the individual got there. These co-attributes are supported by the vast number of possible grid-cell combinations, as previously mentioned, and recent work in humans which indicates an extensive overlap in the brain networks supporting navigation, remembering the past, and thinking of the future [Buzsáki and Moser, 2013].

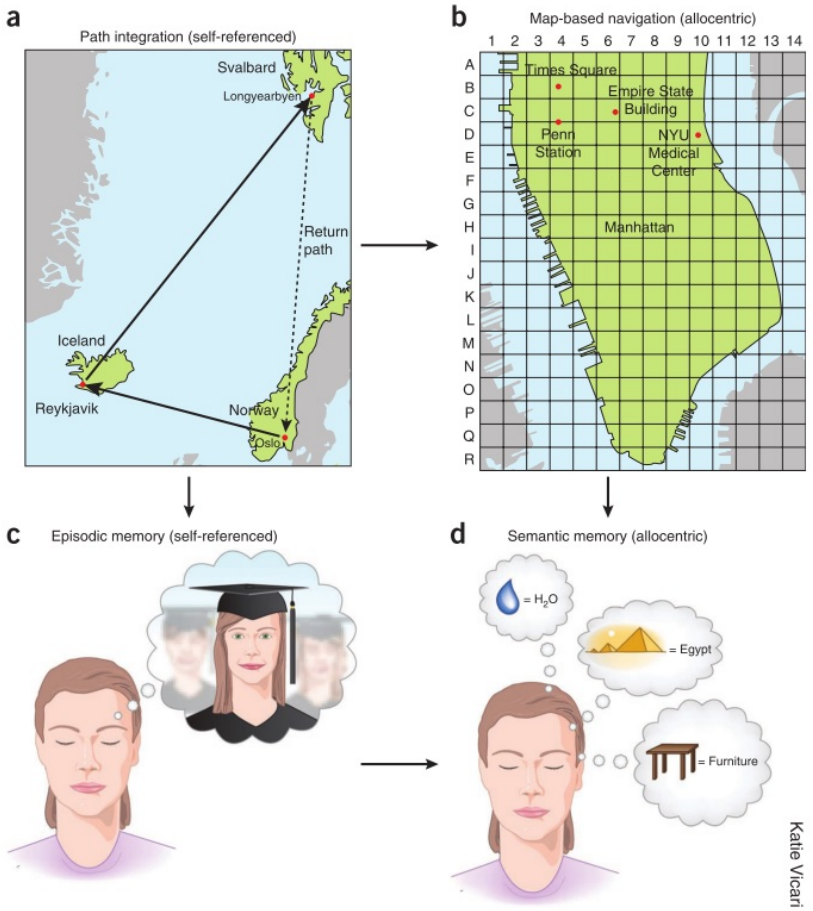

Figure 4.10: *[Buzsáki and Moser, 2013] (a) Path recognition is based on self-referenced information by keeping track of travel distances. (b) Allocentric-map based navigation is supported by the relationships among detectable landmarks. An allocentric map is constructed by exploration (path integration). (c) Episodic memory is 'mental travel' in time and space referenced to oneself. (d) Semantic memory is explicit representations of objects, places and things without temporal and contextual references, but abstracted from episodic memories. These coinciding characteristics indicate an inherent connection between navigation and spatial memory.*

4.4 Attractor Dynamics - Localization

Attractor dynamics is the prevalent model for representing the dynamics of the place- and grid-cell networks, and explains some of the phenomena presented in the previous.

4.4.1 Attractor Networks - Discrete or Continuous

A discrete attractor network is a network with a number of stable states, to witch the system gravitates when it is sufficiently close (such as a dynamical system with limit cycles), see figure 4.11. It represents the energy of the system, where the stable states are low energy states (local minima/basins). Small perturbation will not cause a change in state/basin, a substantial push is needed. When the system goes from one state to another, it does so abruptly. A continuous attractor does not resist perturbations, it moves smoothly from one state to an adjacent one [Jeffery, 2011], see figure 4.12.

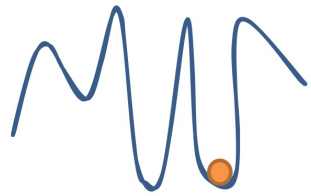

Figure 4.11: *1D attractor landscape for a discrete attractor. The stable states of the system are the basins in which the system (represented by the ball) settles and from which it resists small perturbations [Jeffery, 2011].*

Figure 4.12: *1D attractor landscape for a continuous attractor. Continuous dynamics allow the system to move smoothly from one state to the next [Jeffery, 2011].*

4.4.2 Attractor Dynamics Organize the Activity of Place- and Grid Cells

Attractor dynamics is the prevalent model for place cell computation, and has been supported by experimental evidence [Jeffery, 2011]. The attractor-network model has been one of two main classes of grid cells' computational models, the other being the class of oscillatory interference models. With Stensola et al.'s discovery of sub-layer modular organization of grid cells, based on orientation preference, attractor dynamics have become the prevalent model also for grid cells. Certain properties of the oscillatory-interference model (omitted here) do not coincide with the modular organization of grid cells. Oppositely, the new discovery is nicely represented by a continuous attractor network, allowing inter-module state transitions to be done smoothly; because grid cells in one module tile the environment, translation of activity between two inter-module grid-cells is proportional to the animal's velocity [Stensola et al., 2012].

The place cell network shows both discrete and continuous attractor dynamics. The remapping paradigm, in which the place cell activity is switched from one local population to another, is represented by discrete dynamics. If the stored state of the system is in one attractor basin, and sensing information is received, the state either moves to another basin or it stays in the current basin. When the system is presented with an altered state of the environment (e.g. when the furniture is rearranged), the system compares its stored state to the observed state and decides whether the current state is in the same environment (*pattern completion*), or a different environment requiring a new representation (*pattern separation*). Thus, this explains the property of the place cell network described earlier - that remapping need not occur if some features are removed from the environment [Jeffery, 2011].

The smoothness property of the place cell network is thought to be enabled by the continuous-attractor dynamics of the grid-cell network. The active place cells in each basin of the discrete attractor network together represent a local environment. All these place

cells are considered active in the local environment, and transitions between the active place cells are smooth and can be modelled by continuous attractor dynamics. The "attractors" of this continuous-attractor network is no longer the possible states of the network in the whole environment, but rather the possible activity patterns of the active set, which represents one single place in the local environment. However, the smooth transitions between place cells are thought to be driven by continuous attractor dynamics of the grid cell modules, and is not in itself a continuous attractor network. That is, each grid module is a continuous attractor network, and the activity pattern formed by the collection of grid modules drives the activity of the active place cell set. The place cell activity does, as previously stated, represent the animal's location in the environment [Jeffery, 2011].

The activity change of the attractor network in grid cell modules is driven by the movement of the animal. As mentioned earlier, the tiling property of the grid modules allows the shift in activity to be proportional to the animals velocity (in each direction) [Stensola et al., 2012], see figure 4.13. Thus, the change in total ac-

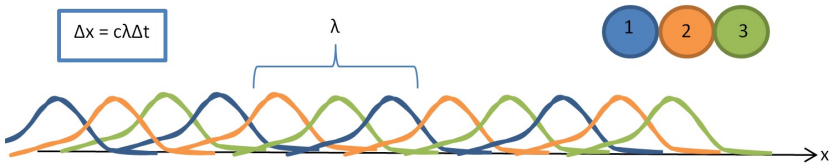

Figure 4.13: *1D environment example, illustrating that the activity of a grid module shifts with velocity. The grid-module consist of three cells, whose activity tile the environment (repeated activity patterns). λ is the scale of the grid-module, and the speed of the animal is proportional to the grid scale.*

tivity, denoted $\dot{A}$ is a function of velocity modulated with the grid scale λ (because lambda gives the period of the activity patterns):

$$\dot{A} = f(\dot{x} \ mod \ \lambda) \tag{4.1}$$

The individual grid cells in a module have connections (synapses) to the other grid cells in the module, and it is these connections

which manifest the attractor network. In short, the connections are trained (through Hebbian learning, which is not discussed further here) so that stable activity patterns are achieved. As an example, inspect figure 4.13. When both the 'orange' and the 'blue' cells' activity levels are small, the 'green' cell's activity level must be large. This cooperative property is created by the weights of the connections between the individual grid cells, and is modelled by continuous attractor dynamics [Jeffery, 2011].

4.5 Chapter Summary

In this chapter, the functionalities of both place and grid cells have been presented. Place cells fire at specific locations in the local environment, while grid cells fire in a regular triangular pattern covering the whole local environment, parametrized by the grid orientation, spatial shift and scale. Grid cells are modularly organized based on grid orientation and scale, which make each module tile the environment with grid cell firings. A place cell firing gives the location of the animal in the environment, and the firing of the place cell is determined by landmark recognition in visual input and the overlapping of firing fields of the most active grid cells in each active grid module. Assumed communicational properties between grid and place cells have been disclosed. Furthermore, the higher level topological layer of biological environment representation has been discussed in short, where nodes in the topological map are represented by salient landmarks, much like hierarchical representations in robotics. Going from one node-map (local environment map) to another requires remapping of both the grid cell modules and the active place cell set.

The implication that the brain's spatial representation system is deeply connected with the formation of memories was briefly discussed. Lastly, the model used for activity shifts in place and grid cell networks, the attractor network model, was introduced.

In the following chapter, a deeper dive will be made into the underlying geometry of the grid cell structures. Specifically, the inves-

tigations focus on how many grid cells are in fact needed to represent an environment of certain size. The motivation for the investigation is to gain insight into the way grid cells represent the environment and why this representation might be useful to emulate.

Chapter 5

Creation of Biologically Inspired Map

In order to get a more thorough understanding of the complex mechanisms of biological navigation systems, the focus here will be on creating a map emulating the grid-based map of the local environment applied in biological systems. The discretization factor of spatial shift in biology has not been disclosed in the literature. In a module, the firing fields of the grid cells tile the environment, but it is unclear how much the firing fields of the grid cells overlap, i.e. how frequent the firing maxima are in the environment. One possible reason this property remains undisclosed is that the activity of all the grid cells belonging to a module has never been measured simultaneously. Thus, the main objective of this chapter is to investigate the influence of discretization factor of orientation and spatial shift, respectively.

In [Solstad et al., 2006], an optimal map of place cells manifested by grid cells is built, much like the task at hand here. The map built in [Solstad et al., 2006] is trying to exactly replicate the mechanisms used in nature, as it is to serve as a mathematical model for the biological grid and place cell structures. In the following design certain abstractions will be made, and the post [Solstad et al., 2006] discovery of the layered structure of grid cells will be incorporated. The main abstraction is to define place- and grid-cell firing fields

as deterministic areas, rather than 2D Gaussian functions used in [Solstad et al., 2006], which replicates the theta-rhythm frequencies in neural firing. The main motivation for the abstraction is that the effect of orientation, spatial shift and room size will for the deterministic-firing map be proportional to the effect in a Gaussian-firing map, while implementation is simplified.

5.1 Spatial Shift, Orientation and Spacing

As discussed in the last chapter, biological grid networks have independent layers of grids, defined by the spacing between grid vertices. The layers are further divided into modules based on grid orientation (angle relative an reference axis). The range of spatial shift of a module will now be given, together with the range of possible orientations, which both are limited by the repetitive nature of grid cell firings.

Range. The range of possible spatial shifts and orientations is found by investigating the grid structure. In the following, the reference point is located at $(x_0, y_0) = (0, 0)$, with the first triangle constellation with orientation $\phi = 0$ as in figure 5.1. The distance from (x_0, y_0) to each of the vertices of the first triangle is $\frac{a}{\sqrt{3}}$, when a is the edge-length of the current grid-layer. From the figure it can be seen that the ranges for shift in the x- and y-directions are $[0, \sqrt{3}a)$ and $[0, a)$, respectively. Furthermore, the orientation range is $\phi \in [0, \frac{4\pi}{3})$.

Note, however, that care must be exercised when defining the discretization factor of spatial shift. The number of discretization steps in each of the x and y directions cannot both be even numbers, if the grid-cell firing fields are distributed over the valid range. Say, if the discretization factor in the x-direction is 2 and the discretization factor in the y-direction is 2 giving a total of four grid cells, as in figure 5.2, then the firing fields of two and two grid cells will overlap.

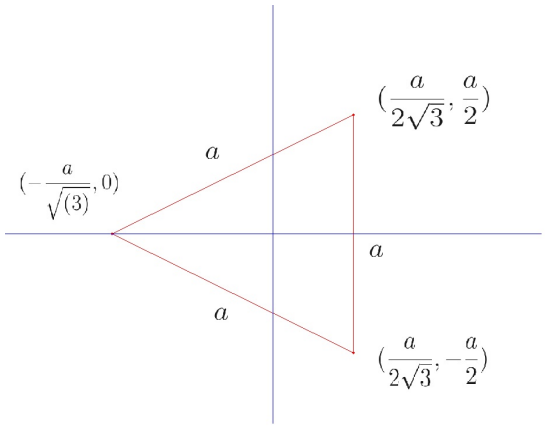

Figure 5.1: *Chosen representation of first triangle of the grid structure without spatial shift or rotation. From the figure one can easily se that a shift in the y-direction of $y_1 = y_0 + a$ with no shift in the x-direction will cause the next triangle of the current grid to be formed. Similarly, with a shift in the x-direction of $x_1 = x_0 + \sqrt{3}a$, another triangle of the same grid is formed. Hence, the ranges for shift in the x- and y-directions are $[0, \sqrt{3}a)$ and $[0, a)$, respectively.*

Thus, in such a situation the inclusion of two of the grid cells is redundant; they do not add to the representation of the environment. The general rule can be stated as:

> *In a module, if the distance between any two grid-origins in the y-direction is $\frac{a}{2}$ and the distance between the same grid-origins in the x-direction is $\frac{\sqrt{3}}{2}$, then the grids defined from the two origins have replicate firing fields in the environment.*

The easiest way to avoid this from happening, is as mentioned to let the discretization factor in at least one of the directions be an odd number.

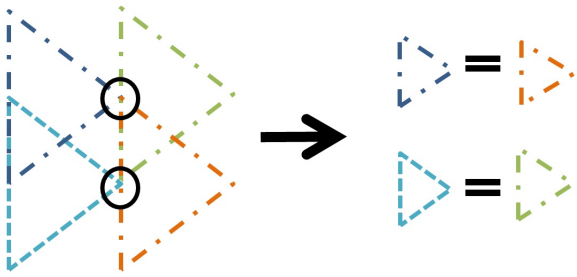

Figure 5.2: *Even numbers of spatial shifts in both the x- and y-direction cause duplicate grids. In the figure four grids are defined, with two possible grid-origin positions in both directions. As can be seen, only two triangular grids are formed in the environment, because two and two of the grid-cell show firing fields which overlap each other.*

5.2 Implementation

The algorithm used to create the map is implemented in MATLAB. The radius of a grid-vertex is defined to be:

$$r_0 = max(\sqrt{2} \cdot a \cdot \frac{1}{2 \cdot N_y}, \sqrt{2} \cdot \frac{\sqrt{3}a}{2} \cdot \frac{1}{2 \cdot N_x}) \tag{5.1}$$

if a is the edge length (scale) of the grid and N_x and N_y are the discretization factors in the x- and y-direction, respectively. This radius ensures that the entire environment is covered by firing fields of a module. In the following, 'place cell' and 'grid cell' will be used to describe the implemented entities with the similar functionality as the cells in biology.

To simplify the calculations, the place cells are represented in a matrix the size of the environment divided by the place field size (here 25 cm). Hence, the firing field of a place cell becomes square-shaped. The grid cells whose firing field overlap the firing field of a place cell, is included in the grid code for that place cell. To check for uniqueness, the grid codes for all place cells are compared and checked for duplicates.

In this implementation a graph-based approach is used to create each

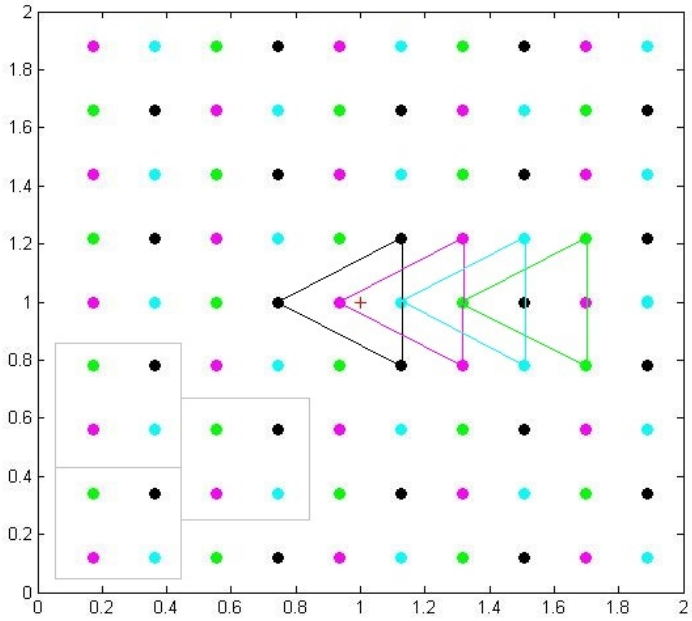

Figure 5.3: *Vertex (firing field) centers of a module, created from the implementation given in 5.1. The module here has the smallest grid scale* $\lambda = 0.44$ *m, and orientation* $\phi = 0$*. Each color represents the center of the firing fields for individual grid cells in the module. The triangles are drawn in to visualize the triangular repetitions of the grid cell firing. The gray squares visualize the tiling property of the grid modules.*

grid, with neighboring vertices being added in cumulative manner. Vertex centers of a grid-module are shown in figure 5.3, which also illustrates the tiling property of the grid-modules. Abstracted pseudocode for the algorithm is shown in algorithm 5.1. Note that (x_0, y_0) gives the location of the reference point in the room, and the negative x-axis gives the reference direction for orientation.

Algorithm 5.1 Place Cell Map Creation

```
1: procedure createPCmap(roomSize, edgeLengths, ΔX, ΔY, ΔΦ, PCsize, (x_0, y_0))
2:     PCmap = distribute(PCsize, roomSize)
3:
4:     for all grids(edgeLengths, ΔX, ΔY, ΔΦ in range ) do
5:         a = currentEdgeLength
6:         BasicTriangle = [(−a/√3, 0)(a/(2√3), a/2)(a/(2√3), −a/2)] + (x_0, y_0)
7:         firstTriangle = rotAndShift(BasicTriangle)
8:         addFiringToMap(firstTriangle, PCmap)
9:         queue = addNeighborsToQueue(firstTriangle)
10:
11:        while queue != empty do
12:            neigh = drawNeighborFromQueue
13:
14:            if neigh in room then
15:                addFiringToMap(neigh, PCmap)
16:                addNeighborsToQueue(neigh)
17:            end if
18:        end while
19:    end for
20:    duplicates = checkForDuplicates(PCmap)
21:    return duplicates
22: end procedure
```

5.3 Unique Representation of Place Cells

As mentioned in the previous section, the collection of grid cell firings of each place cell must be unique for that place cell, in order for the place cells to be defined by grid cell firings. The main investigation in [Solstad et al., 2006] was how many grids were needed to uniquely define each place cell in a simulated room. Each place cell's firing field was about 12 cm, and they argued that diverse orientations were needed to achieve unique place cells, and that diverse spatial shifts were not necessary in the same way. Furthermore, they found that the optimal sampling of spacing was logarithmic, within a set range of $a \in [0.28, 0.73]$ meters. The more recent literature suggest that logarithmic sampling and such limited range of grid spacings are not biologically plausible choices, with the discovery that grid cells are organized in layers based on the grid scale, with a constant relationship ($\sqrt{2}$) between layer scales in the range $[0.44, 10]$ meters [Stensola et al., 2012].

With the added constraints on spacing, the effects of discretization of spatial shift and orientation is tested through simulations, in order to attain a map defined by the least number of grid cells. In addition, several room sizes is simulated to investigate the relationship between number of grids needed and room size. From a practical standpoint a place cell was defined to be 0.25x0.25 m, about the size of a small robot. In the orientation and spatial shift tests, the virtual room is set to be a 10x10 m square, resulting in 1600 place cells in total.

5.3.1 Orientation

Figure 5.4 shows the effect of increase in discretization of orientation ϕ, with no spatial shifts. With only four possible orientations in each layer, and hence a total number of 36 grid cells (only nine layers are used, as the largest layer with 10 meter scale is not included), about two thirds of the place cells in the map are defined uniquely. When 16 orientations are available in each layer (144 grid cells in total), all place cells are uniquely defined.

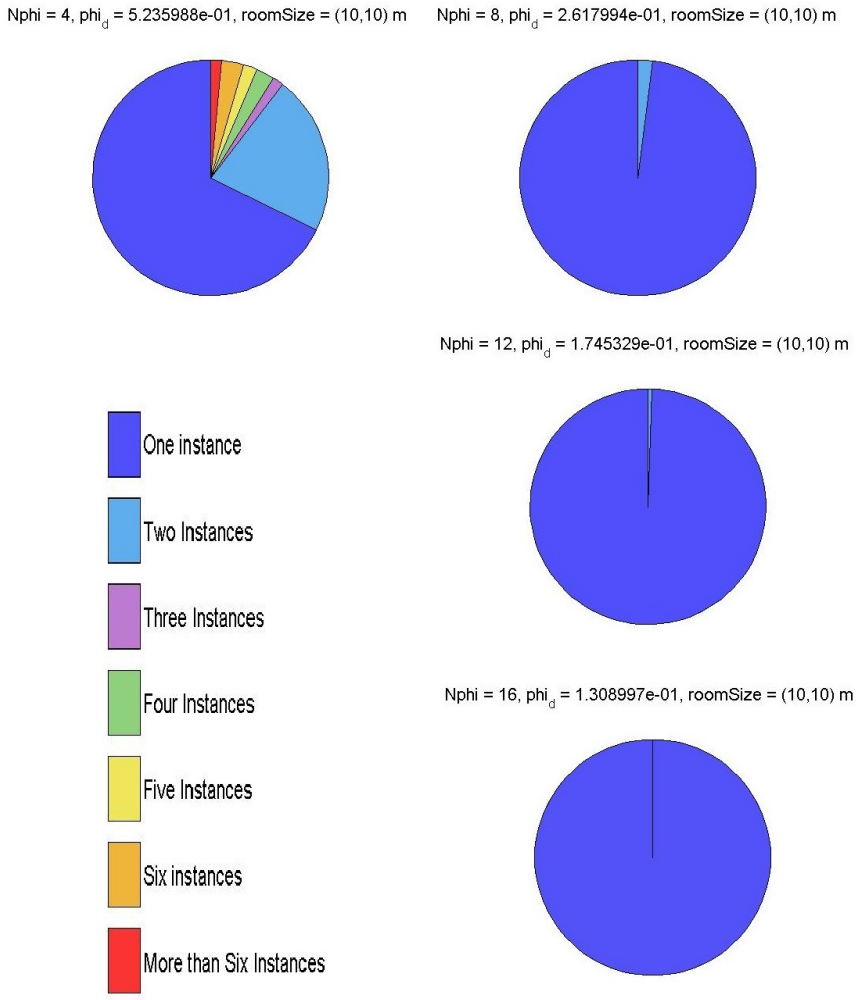

Figure 5.4: *Results from the test of the effect of orientation discretization. The colors indicate the number of repetitions found of the place cells' grid codes. The pie chart in the top left corner, shows that with only four possible orientations allowed, only about two thirds of the place cells in the map are uniquely defined, quite a few have one place cell with the exact same firing pattern. As the discretization step of the orientation is reduced, the number of uniquely defined place cells increases. With 16 possible orientations, that is 144 grid cells in total, the place cells are all uniquely defined.*

The results obtained here are significantly worse than those obtained in [Solstad et al., 2006]. Their results showed that only 50 grids were needed to uniquely define place cells of 12.5x12.5 cm size in a 10x10 m room, when only orientation was varied (no spatial shift). Furthermore, they argued that if the spacings and orientations were set appropriately, then only 20 grids are necessary to define the same place cells uniquely. With 25x25 cm large place cells, the map designed here has uniquely defined place cells when 144 grid cells are used, when only variations in orientation is allowed. There are several possible reasons for the reduced performance of the map created here, compared to the map designed in [Solstad et al., 2006]. As previously mentioned, the constraints on spacing restricts the diversity of grids. Moreover, the choice to define the grid vertices deterministically, is a probable contributor to the increase in grids needed. Addressing each of these issues directly, and hence optimizing the map with respect to the number of grids needed, is a task left for the future. However, work presented in this study may serve as a good basis for this future work.

5.3.2 Spatial Shift

The results of the tests of the effect of the discretization of spatial displacement $(\Delta x, \Delta y)$ are shown in figure 5.5. Only one orientation is used for all the tests in the figure (i.e. only one module per layer). With no spatial shifts, none of the place cells are uniquely defined. With 18 grid cells per module, the entire place cell set is uniquely defined by place cells. It can be seen from the figure that more grid cells are needed to define place cells uniquely when only spatial shifts are allowed, as opposed to when only changes in orientation is allowed. This attribute coincides with the findings made in [Solstad et al., 2006], that the effect of discretization of spatial shift is less than the effect discretization of orientation.

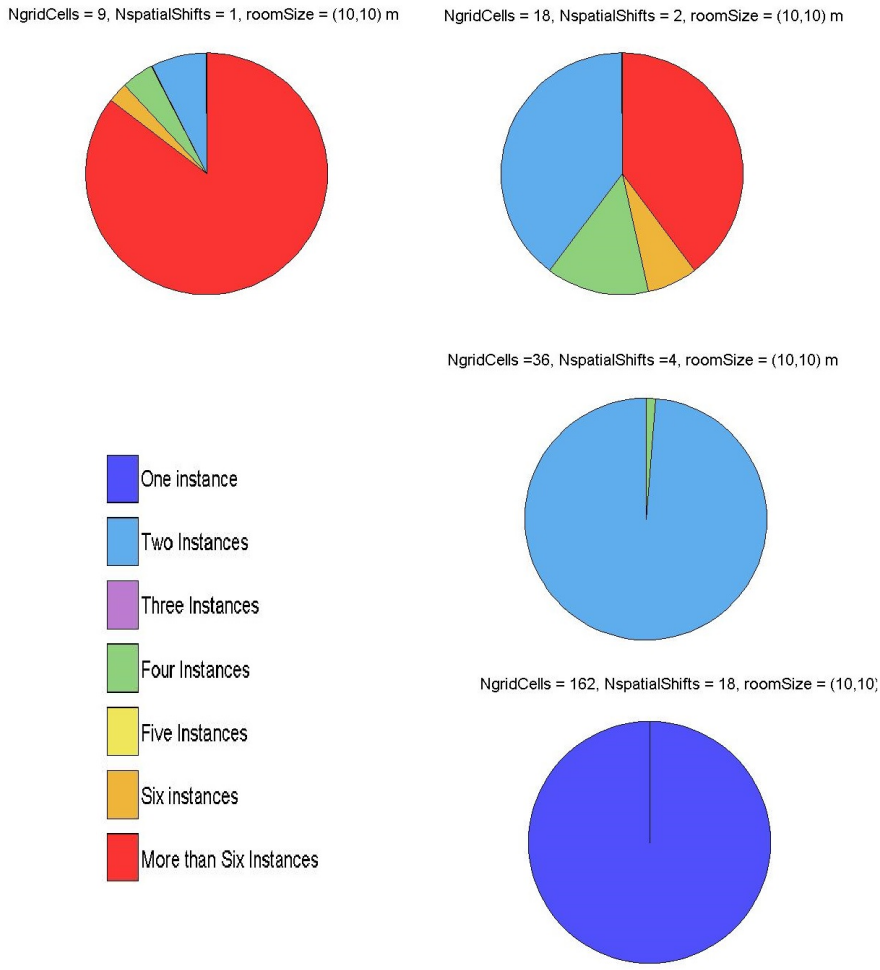

Figure 5.5: *Results from the test of the effect of spatial shift discretization. Only one orientation is used for all of the tests. The pie chart in the top left corner shows that with no spatial shifts, none of the place cells in the map are uniquely defined. As the discretization step of the spatial shift is reduced, the number of uniquely defined place cells increases. With 18 possible spatial shifts the place cells are all uniquely defined. Note the increased number of grid cells needed when only spatial shifts are used, compared to when only changes in orientation is used. This result compares nicely to the claims made in [Solstad et al., 2006].*

5.3.3 Room Size and Best Representation

Figure 5.6 shows the results of the test of the effect of room size, in the range 10x10 to 25x25 meters. The same number of grid cells in each layer is used for all room sizes, but the 10x10 meter room has only nine active layers. The results comply nicely with the results shown in [Solstad et al., 2006] (and intuition): as the room size increases, the number of uniquely defined place cells decreases.

Square Room, side length A [m]	**2**	**4**	**6**	**8**	**10**	**15**	**20**	**25**	**50**
Grid Description									
Number of layers	5	7	8	9	9	10	10	10	10
Number of modules per layer	3	3	6	6	5	5	4	6	5
Number of grid cells per module	2	2	2	2	3	3	4	3	4
Total number of grid cells	30	42	96	108	135	150	160	180	200
Number of place cells uniquely defined (% of total set)	100	100	100	100	100	100	100	100	100

Table 5.1: *Test results from necessary grids to define a square room of side length A uniquely. In the test the place cell size is kept constant at* $0.25x0.25$ *meters. Notice that the increase in room size covered is not proportional to the increase in active grid cells; as the number of active grid cells increases, the area uniquely defined by the grid cells shows a far greater increase.*

In table 5.1 the minimum number of grid cells needed to define

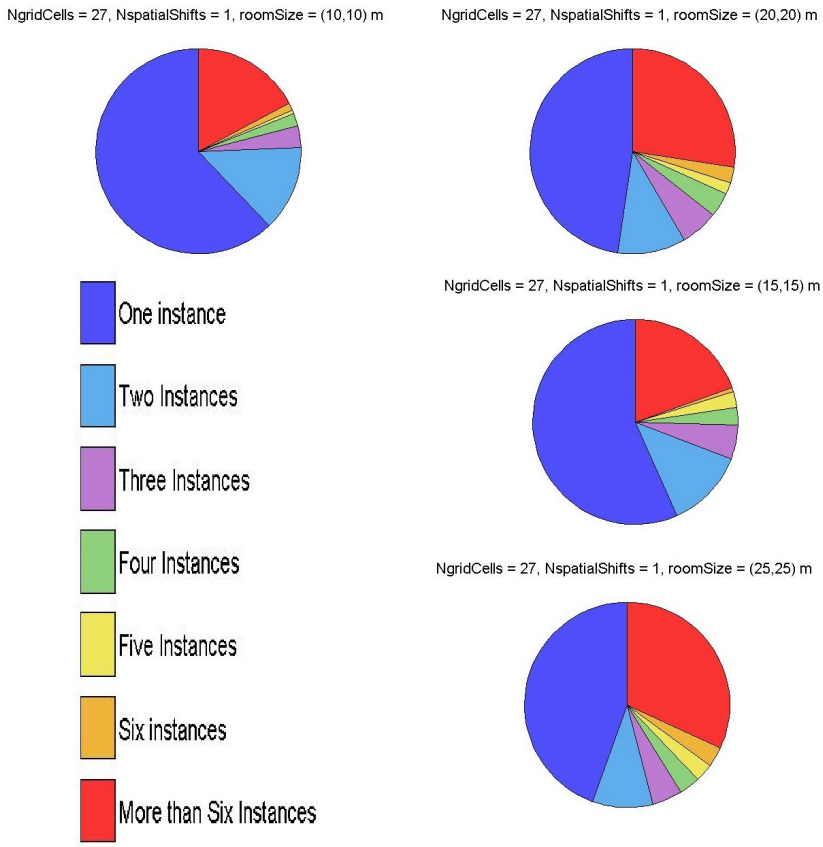

Figure 5.6: *Results from the test of the effect of room size. Three orientation steps and no spatial shifts are used for all of the tests. The pie chart in the top left corner shows the distribution of replication of place cells in a 10x10 meter room. As the room size increases, the bottom one being 25x25 meters, the portion of the place cells which are uniquely defined decreases, in compliance with the results shown in [Solstad et al., 2006].*

0.25x0.25 meter place cells in a range of room sizes is given For each room size, the table lists the number of active layers and the number of discretization steps in both spatial shift and orientation. A plot of the room size as a function of the number of necessary grid cells, data as given in table 5.1, is shown in figure 5.7. The function shows a close to exponential increase in room size with a linear increase in active grid cells. This is consistent with findings made

in [Sreenivasan and Fiete, 2011], where it was shown that a linear increase in grid cells result in an exponentially larger area uniquely defined by these grid cells.

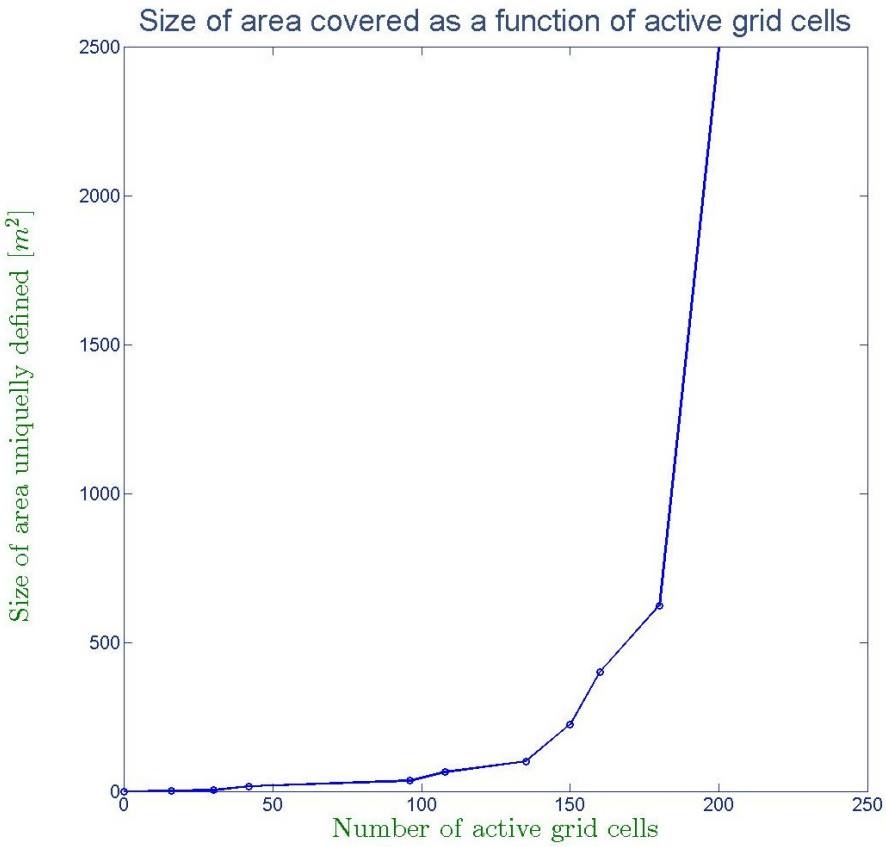

Figure 5.7: *Plot of size of uniquely defined room as a function of necessary grid cells, from values given in table 5.1. Notice the exponential-like form of the function, which is consistent with the findings made in [Sreenivasan and Fiete, 2011].*

5.3.4 Calculation of one Place Cell's Grid Firings

When the local spatial representation is shifted from one local environment to another, the correct place cell in the entered frame must fire, and thus the corresponding grid cells must also fire. In order to calculate the grid cell firings which correspond to a specific position in the new frame, the distance and angle of the position relative the reference point must be known. These parameters might be obtained by knowing the constellation of salient landmarks in the local environment and the distance and angle of the current location relative one of these landmarks. However, none of the literature studied discusses how the initial firings in a local map emerge. In the following, the distance and angle to the reference point is assumed known.

In the system used for the previous tests, indexing the place cell 'matrix' to obtain the grid firings is not a biologically plausible operation, since it is the grid cells which manifests the place cell map in the environment, not the other way around - the place cell network's metric measure is the grid cell network. Thus, place cells adjacent in the environment need not be adjacently represented in the matrix. A method for calculating the correct grid cell firings at a position will now be introduced.

In figure 5.8 a grid defined from the reference point (x_0, y_0) is shown. The current position (x_p, y_p) falls inside one vertex' radius of the grid, and hence that grid cell should fire at the given location. The x-distance to the closest grid vertex is found by inspecting the residue of integer division by one triangle's height h (see figure):

$$n_x = floor(\frac{x_p - x_0 + \frac{\sqrt{3}a}{3}}{h}) \tag{5.2}$$

$$x_{residue} = x_p - x_0 + \frac{\sqrt{3}a}{3} - n_x \cdot h \tag{5.3}$$

$$x_{closeNode} = min(x_{residue}, h - x_{residue}) \rightarrow n_x = n_x + 1 \text{ if } h - x_{residue} \text{ is min} \tag{5.4}$$

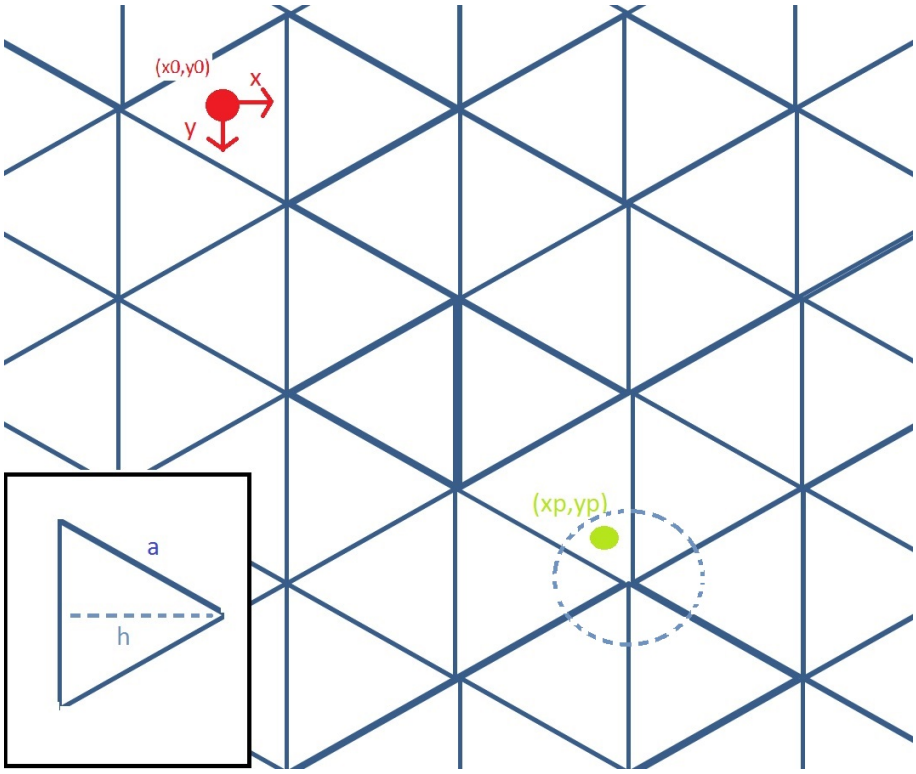

Figure 5.8: *Illustration of one grid cell's firing field. If the location of the current position (green) relative the reference point (red) is known, then investigations of whether or not a particular grid cell fires at that location can be calculated by inspecting the residue of integer division by h and a in the x and y directions, respectively.*

The calculation of the y-distance to the closest node depends on n_x:

$$n_y = floor(\frac{y_p - y_0 - \frac{a}{2}}{a}),\ if\ n_x\ odd \tag{5.5}$$

$$n_y = floor(\frac{y_p - y_0}{a}),\ if\ n_x\ even \tag{5.6}$$

In the same way as was done in the x-direction (equations 5.3 and 5.4), the distance to the closest node in the y-direction is calculated. If the point is within the vertex radius of the grid, then the grid fires at that location. By rotation of the coordinate system with the

rotations of the different grids, the same procedure can be calculated for all grid orientations. Testing with an implementation of the above described calculations showed collections of grid firings identical to the ones found in the place cell map from subsection 5.3.3, when the same orientations and shifts were used (results not shown).

5.4 Chapter Summary

In this chapter a biologically inspired local map was created, based on popular beliefs of cooperation between place cells and grid cells. Certain abstractions were made; the place cell size was set to a constant 25x25 centimeters, while place cells of different sizes have been found in the hippocampus [McNaughton et al., 2006]. The grid cell and place cell firings were set to be deterministic, while in reality they are similar to 2D gaussian functions [Solstad et al., 2006].

The simulations showed that 144 grid cells where needed to define place cells of 25x25 centimeters uniquely in a 10x10 meter room, when only variations in orientation was allowed (and 10 scale layers). With only variations in spatial shift, 162 grid cells were needed to define the same place cells uniquely. Thus, the effect of variations in orientation is larger than the effect of variations in spatial shift, an observation which is consistent with the finding in [Solstad et al., 2006].

Concluding the simulations, the effect of room size was investigated. As expected, the number of place cells uniquely defined decreased with an increase in room size, when a constant number of grid cells were used in each layer. Furthermore, it was shown that a linear increase in grid cells used caused an exponentially larger area to be uniquely defined, which is consistent with the observations made in [Sreenivasan and Fiete, 2011].

Lastly, a method for calculating the initial grid firings in a local environment was introduced. The method was created based on geometric observations, without claiming it is the way initial firings are obtained in biological systems. Nevertheless, it presents a method

applicable in robotics, where it could be a part in solving global-localization and kidnapped-robot problems.

The next chapter discloses proposed theories on how path integration (tracking), sensor fusion and navigation is done in biology, using the hitherto described spatial representation.

Chapter 6

Biological Position Tracking and Navigation

In the previous chapter the geometric structure of the grid based maps were defined, and it was shown that only a sparse number of grids were needed to explicitly define a local map. In this chapter, the investigation evolves to the biological mechanisms for updating the animal's position in terms of the grid structures. First, the mechanism for integrating idiothetic cues, known as *path integration* will be described. Subsequently, the mechanisms involved in sensor fusion of visual (allothetic) and self motion (idiothetic) cues are presented. Lastly, potential navigation strategies used in biological systems will be presented, followed by a simple system visualizing some of the mechanisms presented in the preceding sections.

6.1 Path Integration

Because grid cells fire at the same (x, y) position in the environment irrespective of the animal's speed and direction, and that the firings persist in darkness, self motion is thought to be used actively by grid cells to keep track of the animals position in the environment. This process of integrating self motion is as previously mentioned called *path integration* [Giocomo et al., 2011].

As stated earlier, grid cells which are proximate in the brain, share

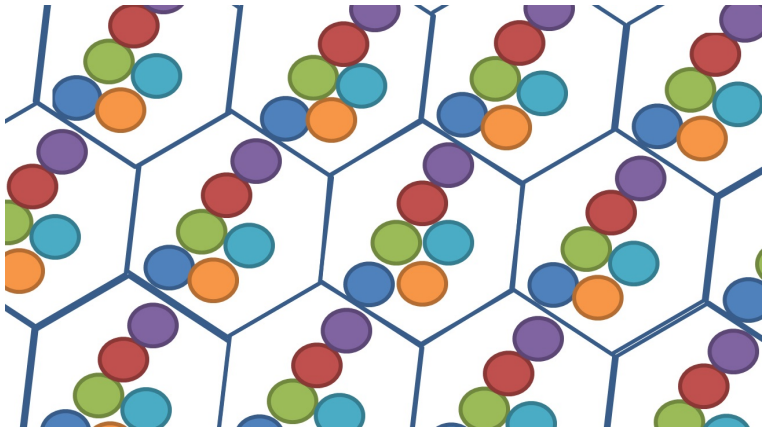

Figure 6.1: *Grid cells with common orientation and spacing but with spatial shifts (phases) distributed in the domain, tile the environment. The circles represent vertices of grids, vertices with common colors correspond to one grid, represented by one grid cell. Here, the tiles are hexagons, but this need not be the case; any 'tileable' shape can be used, as long as one tile only contains one vertex from each grid. Organization of grid cells in modules based on common orientation and spacing, is only an assumption, but it seems reasonable with regards to path integration and navigation [Kubie and Fenton, 2012]. Note that the tiles are not completely filled for graphical purposes, in reality the entire space within each tile is occupied by grid cell firings.*

a common orientation and common scale. However, they have random relative spatial shifts, allowing a small number of co-localized grid cells to tile the environment completely, see figure 6.1. As mentioned in chapter 4, this attribute raises the possibility that within each grid layer, the grid cells are further modularized with respect to orientation, where each module independently tile the environment [Barry and Bush, 2012]. In light of the investigations made in chapter 5, this modular organization is not optimal with respect to the minimum number of grid cells required, which would entail only a small number of spatial shifts and an large number of orientations. However, as will be shown in the following, this modular organization shows some desirable characteristics with respect to path integration. Furthermore, the modular organization provides the brain with a unique ability to detect noise which is uncorrelated across modules

[Sreenivasan and Fiete, 2011].

6.1.1 Conjunctive Cells

Co-localized with grid cells in the entorhinal cortex are, as mentioned, border cells, head direction cells and conjunctive cells. Experiments have shown that conjunctive cells serve as the bridge between head direction cells and grid cells, in that they have a similar organizational structure to that of grid cell modules, only their firings are dependent on which direction the animal is facing. One layer of conjunctive cells will only fire when the animal is facing a particular direction. Furthermore, the activity of these cells are found to be proportional to running speed and when there is no motion, the layers of conjunctive cells are silent. Consequently, the conjunctive cells are thought to be the medium transferring self motion information to the grid modules. The activity of the conjunctive cell modules project asymmetrically to the grid cell modules, causing a change in the location of the activity bump in the continuous attractor network of a grid module (see figures 6.2 and 6.3) [McNaughton et al., 2006, Giocomo et al., 2011].

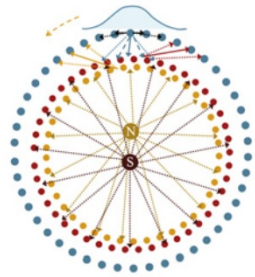

Figure 6.2: *[Giocomo et al., 2011] One-dimensional attractor dynamics. Blue: grid cells form a ring of connectivity, where each cell has strong connections to nearby neighbors, with the strength of connectivity decreasing with distance, causing a 'bump' of activity. Red and orange are so-called 'conjunctive cells', which incorporate the direction of the movement. When an animal moves south (for example), the conjunctive cells with preferred direction south will asymmetrically connect to the grid cell layer, and move the bump in the appropriate direction. The displacement of the bump is proportional to the movement speed [Giocomo et al., 2011].*

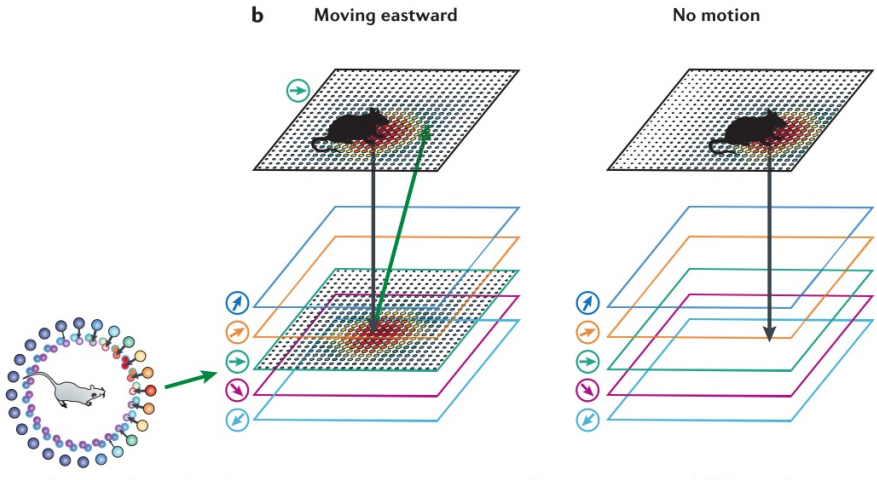

Figure 6.3: *[McNaughton et al., 2006] A model for path integration. The bump of activity in the grid cell layer can be made to move in correspondence with the animal's direction and magnitude of motion using an intermediate layer of cells which are conjunctive for position and head orientation [McNaughton et al., 2006].*

6.1.2 Error Handling Mechanism

In [Sreenivasan and Fiete, 2011], they show that the repetitive nature of grid cell firings makes the system extremely sensitive to noise which is uncorrelated across modules. The noise is present due to integration error from integrating velocity, which is done independently in each module. Integration error is a common problem in neural circuits. The integration error results in firing of a wrong grid cell in the module, which combined with the firings from the other modules represents a location far from the true location, see figure 6.4.

The error is thought to be detected in the hippocampus, where the last active place cell drive cells with similar place preferences. If the grid-code (collection of grid cell firings) arriving from the entorhinal cortex is not in the proximity of the last active cell, then the hip-

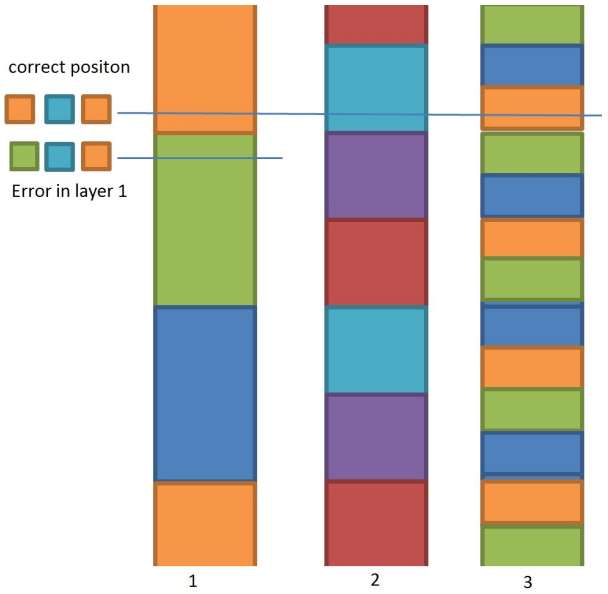

Figure 6.4: *1D representation of three grid modules from three different layers. Each color in one module represent one grid cell's firing field, and combined the cells in a module tile the environment. An internal error in one of the modules (here module in layer 1), resulting in an incorrect position update, will when combined with the grid firings of the other modules result in a position which is unrealistically far from the last location (not displayed in illustration). The large location error caused by errors in individual modules are easily detected and handled by the entorhinal system.*

pocampus sends the grid code of the proximal place cell whose code is most similar to the received code back to the entorhinal cortex for error correction. See [Sreenivasan and Fiete, 2011] for more details.

Because only uncorrelated noise is handled by the error handling mechanism, errors which manifests equally in all modules will not be corrected. Thus, erroneous velocity input will not be handled by this mechanism, instead it is handled by correcting grid firings by incorporation of visual information.

6.2 Sensor Fusion

In biological systems the hippocampus is thought to integrate visual information and grid information when determining the firing of a place cell. Determining which information to weigh heaviest is done through analyzing which source carries the most correct information at a given time instance [Buzsáki and Moser, 2013, Samu et al., 2009].

There is evidence showing that in humans, when the visual and self-motion information are in conflict, either the visual information resets the self-motion system (represented by grid cells), or the information from the two channels are integrated. Specifically, adults, when facing cues in conflict, use a weighted average of the allothetic and idiothetic cues to determine their location. In contrast, young children alternate between which information source to rely on, without combining them. Thus, the weighing of information is somehow learned through experience. In rats, when vision and self-motion are in conflict, the location is generally determined from the visual stimuli [Samu et al., 2009]. A complete model of how integration of idiothetic and allothetic information streams is executed has not yet been realized, because many of the model components remain undisclosed [Weitzenfeld et al., 2012].

6.3 Biological Navigation

Preliminary theoretical results have indicated the involvement of the grid/place cell spatial representation system in navigational tasks. Specifically, entorhinal lesion studies indicate that damages to the grid network does impair an animal's ability to reach a goal. Furthermore, accumulating electro-physiological results from the last 15 years have shown that place cells can fire non-locally, encoding trajectories distant from the animal's current location. Preliminary results indicate that the grid cells also participate in these preplay events [Barry and Bush, 2012]. Only two models have been published to date which direct navigation on the basis of grid firings. Both methods are vector-based, and show how the grid cell structure can be used to traverse unvisited space on a path toward the goal lo-

cation [Kubie and Fenton, 2012, Erdem and Hasselmo, 2012]. These two methods will now be presented.

6.3.1 Forward Linear Look-ahead Probing, [Erdem and Hasselmo, 2012]

In [Erdem and Hasselmo, 2012], a goal directed navigation model is proposed, based on forward linear look-ahead probe of trajectories in a network of head direction cells, grid cells, place cells and prefrontal cortex (PFC) cells. When an environment is partially explored, the virtual rat receives memory of its goal location, and picks its next movement direction by forward linear look-ahead probe of trajectories in several candidate directions.

Prefrontal Cortex (PFC) Cells. Experimental results have reported observations of place cell-like activity in recordings from the medial PFC during navigation planning in rats. Other experiments have shown PFC activity corresponding to preplay of trajectories. In [Erdem and Hasselmo, 2012], the PFC serves as an extension to the place cell map; that is, each place cell has an associated PFC cortical column associated with it.

Each column has three cell layers which are interconnected. The recency layer Q, maintains a recency signal associated with the place cell, and is used to update the lateral connections among the topology layer (U) cells. The topology layer's lateral connections are updated incrementally with the exploration of the environment. The reward layer R maintains the reward signal q_k associated with place cell p_k, see figure 6.5 ([Erdem and Hasselmo, 2012]). As default the reward signal is set to zero. It is this reward signal which is used to represent goal locations.

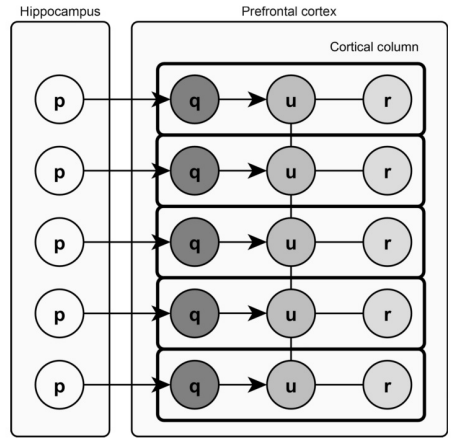

Figure 6.5: *[Erdem and Hasselmo, 2012] The cortical column network in the prefrontal cortex is associated with individual place cells as shown. p represents a place cell, q represents a recency cells and u and r represent a topology and reward cell, respectively. The cortical columns are interconnected laterally through the topology layer, such that the neighborhood of place cells is defined explicitly. In the figure only two neighboring place cells are indicated, while in reality there is many more [Erdem and Hasselmo, 2012].*

Method. Place cell firings are generated as an environment is explored. The place cells are constructed from the collection of grid cell firings, and the grid cells further provide the continuous coding of space. When an environment has been sufficiently explored (at least one path between the location of the virtual rat and the goal location is mapped) the goal location is set:

- First, the goal place cell is chosen. The reward signal associated with that place cell is set to one, which indicates maximum reward.

- The reward signal diffusion process is run next. It involves assigning reward signals in a gradient toward the goal place cell (a similar proposal to goal location representation was made in [Burgess and O'Keefe, 1996]). This way, the length of the linear probe during the navigational process can be shorter, lessening the effect of noise accumulation during the linear probing. In addition, the direct line between the virtual rat and the goal location might be obstructed, and the reward signal diffusion

allows the rat to avoid these obstacles (explained in more detail in the following).

When the goal location is set and the diffusion process is run, the navigation toward the goal can start. It is done through the following steps:

- The virtual rat decides what direction to proceed to reach the goal by probing several forward linear look-ahead trajectory probes with different headings. Each of these probes engages the whole head direction cell → grid cell → place cell circuit as if the rat was moving down the probe trajectory. The decision is made based on which trajectory activates the highest reward signal. If no non-zero reward signal is found along any of the trajectories, the virtual rat proceeds at random, and probes again after some time.

- The virtual rat reprobes after 4 centimeters are traversed, in the implemented system. Erdem and Hasselmo suggests that an interesting alternative could be to reprobe when novel paths between familiar locations are encountered, allowing the discovery of short-cuts.

- The goal location is reached in a finite number of steps.

Obstacle Avoidance and Short cuts. The diffusion process, together with an obstacle detection mechanism, allows the avoidance of obstacles. When a familiar or unfamiliar obstacle is obstructing the rat's movement along its chosen path, the rat probes again, in directions sufficiently far from the direction of the obstacle. Because of the diffusion, the rat chooses the optimal avoidance path.

In this system, the topological layer of the PFC columns indicate the lateral connections between locations in the environment. A short-cut in this setting entails traversing from one place cell to a proximal one without having previously established a lateral connection between these place cells. Furthermore, the probing mechanism allows the traversal of unvisited space. If the highest reward signal is located on the other side of an untraversed area, the regularity of the

grid cells are used to traverse the unfamiliar space, see figure 6.6. Note that if the length of the probe is shorter than the untraversed area, then no reward signal will be found along the probe, and an alternative route will be taken [Erdem and Hasselmo, 2012].

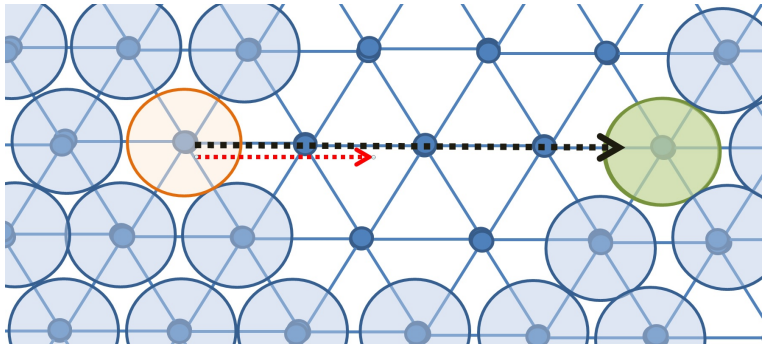

Figure 6.6: *The forward linear look-ahead probing allows short-cuts to be taken across unfamiliar space. (orange) currently located place cell, (green) goal place cell. The black arrow is a linear probe in the direction corresponding to the shortest path to the goal. Across unvisited space the metric provided by the grid cell network is used for navigation. If the probing length is too short (red arrow), the goal place cell will not be reached by the probe, and an alternative route is taken (corresponding to the highest reward signal within the probe length radius of the occupied place cell) [Erdem and Hasselmo, 2012].*

Biological Plausibility. The cortical column representation of PFC cells and their proposed functionality, is not specifically supported by any neuro-scientific findings. Furthermore, even with results supporting replay, the mechanism involving probing in a multitude of directions at a location has not been biologically validated. Despite this, the system corresponds nicely to animals ability to follow the shortest route in homing situations, and does not directly contradict any biological findings [Erdem and Hasselmo, 2012]. In addition, it shows a way the grid-cell system can be used to traverse previously untraversed space.

An important observation is that conjunctive cells are not part

of the above described system. In [Erdem and Hasselmo, 2012], an oscillatory-interference like model was used to update grid cell firings. In the method presented in [Kubie and Fenton, 2012], attractor dynamics and conjunctive cells are used as path integration mechanisms. Furthermore, they propose that linear look-ahead is performed in the conjunctive cell network primarily. More extensive properties of their model will be given next.

6.3.2 Linear Look-ahead using Conjunctive Cells, [Kubie and Fenton, 2012]

In [Kubie and Fenton, 2012], the main focus is on the connection strengths among grid cells and conjunctive cells, and how the connection strengths may serve as the basis for a linear look-ahead mechanism.

Learning Grid- and Conjunctive Cell Connectivity Strengths
Remember that in an attractor network model, each grid cell within a module has connections to all other grid cells in the module. In [Kubie and Fenton, 2012], the strengths of connections were acquired during a learning run. By letting the virtual rat run through an enclosed environment, while counting the number of times one connection was activated (*'hit'*, that is, leaving from one grid cell location and entering the grid cell location on the other side of the connection) and the number of times it was not activated ('miss'), the *hit ratio* gave the connection strength:

$$hit\ ratio = \frac{hits}{hits + misses} \tag{6.1}$$

The same was done for conjunctive cells, where a head direction was included in the update rule of the cells firing, and thus affected the connection strengths. Note that the connections between conjunctive cells are not bidirectional (as they are dependent on head direction). In both learning processes noise was added for a more realistic activity simulation. Figure 6.7 ([Kubie and Fenton, 2012]) show the connection strength between a pair of cells as a function of heading difference. The figure show that strong connections are formed for

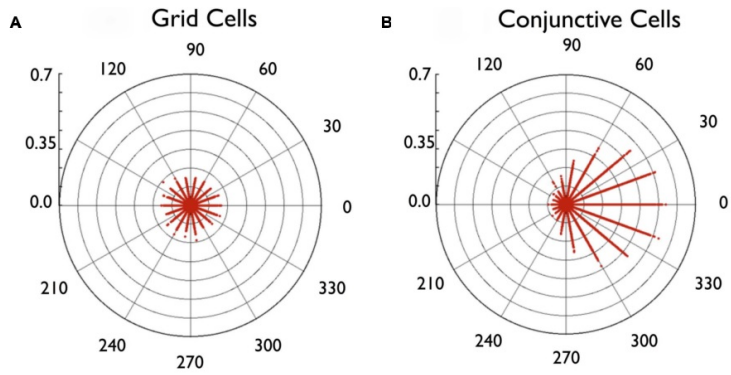

Figure 6.7: *[Kubie and Fenton, 2012] Circular scatter plots of connection strength as a function of difference in heading preference of the connected cells. (A) Connection strengths between pairs of grid cells, which are unmodulated by heading preference. (B) Connection strengths between pairs of conjunctive cells, where high values are found between pairs with similar heading preference (difference value near zero).*

conjunctive cells with similar heading preferences. As expected, the connections between grid cells are unmodulated by heading preferences, and they are lower than the conjunctive cell connections.

Linear Look-ahead The conjunctive cells showed high connectivity, and together with their heading preference property, this makes them good candidates for performing linear look-ahead. The linear look-ahead is executed by activating the outflow connections of the currently activated cell (corresponding to the virtual rat's location). The excitation level of each of the downstream cells of the connections is then calculated from the connection input. If the excitation level exceeds some threshold, the conjunctive cell is activated. The firing cells together form a circular cluster in the environment, with the centroid of the cluster proceeding along the initial head direction, see figure 6.8.

This process is iterated (proportional to the magnitude of virtual ve-

[1]Image collected from supplementary material, Movie s1, found at http://www.frontiersin.org/Neural_Circuits/10.3389/fncir.2012.00020/abstract

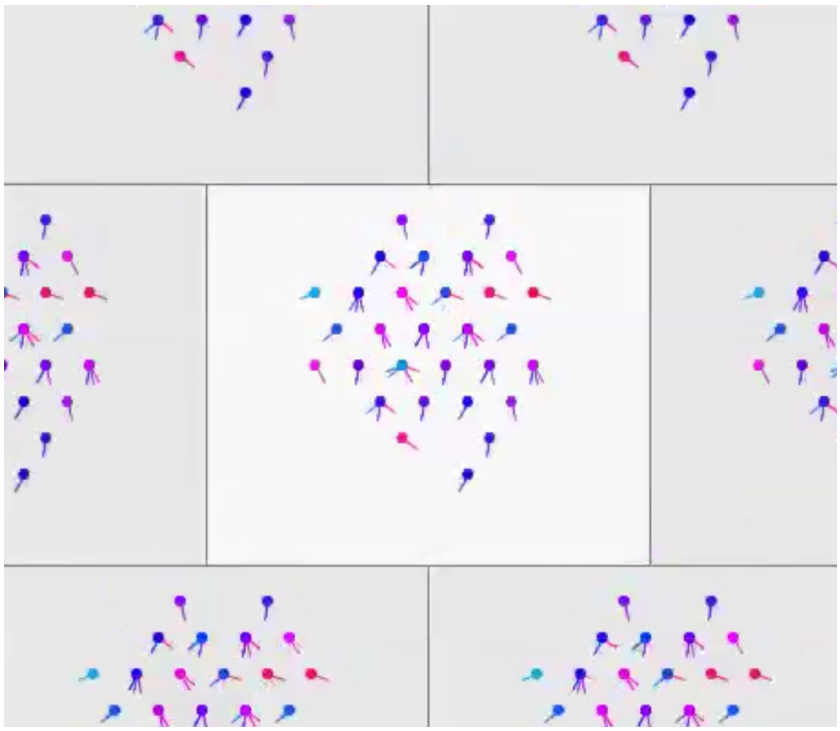

Figure 6.8: *[Kubie and Fenton, 2012]* [1] *The activity centroid in the firing fields of the conjunctive cells move in the preferred heading direction. The brick-like tiling of the environment represents the area in which no repetitions of one conjunctive cells firing field occur (the same tiling as the grid cell modules). Circular dots - conjunctive cells' firing locations within the tile. The 'sticks' emerging from the dots represent their preferred direction (multiple sticks correspond to different conjunctive cells with the same firing location). The centroid of the active cells proceeds in the preferred heading direction (here* 275° *counter-clockwise).*

locity) for the newly activated cells, creating a shift in activity along the direction of virtual motion. Note that, as mentioned, the rat's head direction (the direction of linear look-ahead) is included in the activity measure of conjunctive cells, hence, the conjunctive cells with a heading preference significantly different from the direction of motion are less likely to be activated ('less likely' because of the added noise).

The above described process can be performed with real velocity input or 'imagined' velocity, corresponding to the path integration and linear look-ahead processes, respectively. In their study, Kubie and Fenton does not provide a means for defining a goal location as done in [Erdem and Hasselmo, 2012]. However, the two models only differ in the way linear look-ahead is implemented, hence, a similar goal definition to that of Erdem and Hasselmo can be applied to the model in [Kubie and Fenton, 2012].

Biological Plausibility

The model presented in [Kubie and Fenton, 2012] is more biologically plausible than the one in [Erdem and Hasselmo, 2012], in that the conjunctive cells used for linear look-ahead have been proven to exist. Additionally, the proposed properties of the conjunctive cells are confirmed by experiments. It should be noted, however, that the assumption of path integration being executed in the entorhinal cortex has not been confirmed, despite the experiments supporting this notion (as described in sections 4.1 and 6.1) [Moser et al., 2008].

6.4 Simple System Implementation - BioEmul

In this section a simple localization system is implemented, entitled BioEmul. The implementation is meant to underline the conceptual properties, focused on the above described mechanisms of path integration, sensor fusion and probing. Thus, a number of case-simplifications are made:

1. Movement is restricted to one dimension (1D), making implementation of conjunctive layers excessive. In the 1D system the conjunctive layers, and thus head directions, are represented by the direction of motion (forward/backward).

2. Boundaries are not implemented - the traversable space is a 1D circle.

3. Only five layers of grids, each layer containing one tile-module (making bug discovery easier).

4. The smallest layer has spacing of 0.5 meters, and for each layer the spacing is doubled (making bug-discovery easier).

5. Cell firings are represented by sine functions rather than Gaussians (because of their inherent periodicity).

6. The continuous attractor dynamics is not defined explicitly, it is rather represented by the periodicity of the sinusoids, which renders stable activity patterns across the environment (shifting the activity of one cell's firing in a direction causes a set change in the others, equivalent to attractor dynamics).

The above described grid cell structure will uniquely define 48 place cells, each covering 0.1667 meters of the traversed area. Hence, the total length uniquely defined is 8 meters. Note however, that the system design facilitates expansion to more movement-directions and a larger area uniquely defined by the inclusion of more grid cells in each layer, and addition of grid-layers with larger scale.

In figure 6.10 the *simulink* implementation of the system is shown. It consists of four types of subsystems; five individual grid update modules, a place module, a correction module and a navigation module. The navigation module generates control actions with aim for the goal, the control actions are fed to the grid modules which update their grid firings. Each grid module sends its firing code to the place module, which associates the grid codes with a place cell. If the place cell is different from the one received by the correction module, the place module feeds the correction place cell ID back to the navigation and the corrected grid code to the grid modules. Each of these modules' functionality is now to be described in detail.

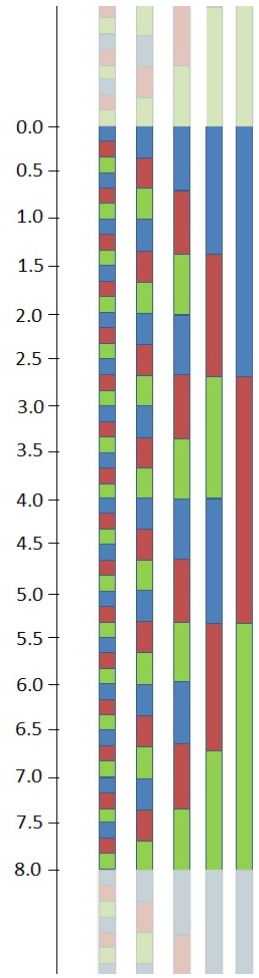

Figure 6.9: *Grid pattern of the 1D area. With five spacing layers (0.5,1,2,4,8 m), each with one module consisting of three grid cells (blue, red, green),* $\frac{0.5}{3} \approx 1.667$ *meter place cells in an 8 meter line is uniquely defined. On both sides of the defined line are duplicate grid codes of the defined area.*

6.4.1 Path Integration in Grid Modules

As mentioned previously, the firings of one module's grid cells are represented by phase-shifted sine functions. The period of

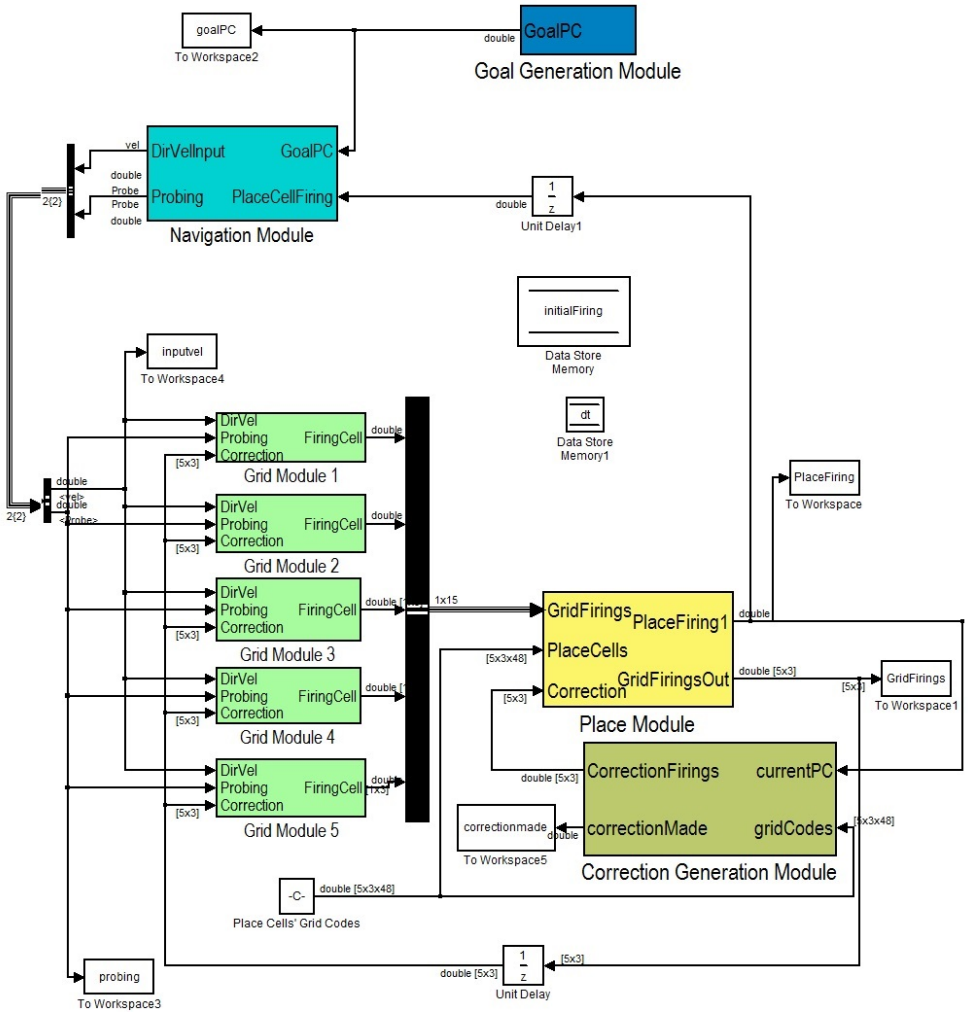

Figure 6.10: *MATLAB Simulink implementation of Simple 1D Navigation System.*

the sine functions are equal to the layer's spacing, and each grid cell's firing function is shifted by $\frac{2\pi}{3}$, so that each grid has the modules maximum firing over a third of the tile/period;

$$f_{\lambda_1}^{GC1} = sin(\frac{2\pi}{\lambda_1}x) \tag{6.2}$$

$$f_{\lambda_1}^{GC2} = sin(\frac{2\pi}{\lambda_1}x - \frac{2\pi}{3}) \tag{6.3}$$

$$f_{\lambda_1}^{GC3} = sin(\frac{2\pi}{\lambda_1}x - \frac{4\pi}{3}) \tag{6.4}$$

where λ_1 is the scale of the layer and x is the position within the tile.

Algorithm 6.1 Grid-Update Module Pseudo-code

1: **procedure** UPDATEGRIDFIRINGS(*direction*, *velocity*, *probOrNot*, *correction*)
2: **if** *probOrNot* = probing and $probOrNot_{t-1}$ = not **then**
3: storedState = $State_{t-1}$ ▷ Store location when probing starts
4: **else if** *probOrNot* = not and $probOrNot_{t-1}$ = probing **then**
5: $State_{t-1}$ = storedState ▷ Retrieve prior-to-probing state
6: **end if**
7: **if** correction != $outFiring_{t-1}$ **then**
8: $State_{t-1} = correction$
9: **end if**
10: pos_{t-1} = **findPositionInTile**($State_{t-1}$) ▷ overlapping arcsin outputs
11: $pos_t = pos_{t-1} + \Delta t \cdot velocity \cdot direction$
12: $pos_t = scale$ **modulo** pos_t ▷ keep within tile
13: $State_t$ = **firingAtPos**(pos_t) ▷ eqs: 6.2, 6.3, 6.4
14: $index$ = **maxAtIndex**($State_t$)
15: $outFiring_t(index) = 1$, $outFiring_t(!index) = 0$
16: **return** $outFiring_t$
17: **end procedure**

In algorithm 6.1 the pseudo-code for the grid-update module is shown. Notice that when probing is signalled to start (line 2), the state, which is the collection of firings within the layer, is stored. When probing is signalled to end (line 4), the stored state is retrieved.

The output from the grid modules to the place module is the grid

cell with the maximum firing. Across the tile, there will always be one grid cell whose firing is larger or equal to 0.5. This value serves as the firing threshold, indicating which grid cell is considered active. If two cells have equally large firing values (0.5), then the direction of motion decides which one should fire: the cell whose firing field is about to be entered fires.

6.4.2 Associating the Grid Code with Place Cells

The task of the place module is rather simple. A matrix containing all place cells' grid-codes is compared to the input grid-code and the index of the element in the matrix with the same grid-code represents the currently occupied place cell. Note that the order of place cells in the matrix need not correspond to the order of firings in the environment, though they are here for visual purposes. It is the grid cells which anchor the relative positions in the environment, not the place cells.

If a correction is received by the correction module, it is in the form of a place cell (an index). If the index from the correction module is not the same as the one derived from the input grid-code, it is the correction module which defines the 'true' location and whose grid code is fed back to the grid modules and the navigation module. This is similar to the 'sensor'-weighing done by young animals, as described in section 6.2.

Because the different grid modules are thought to operate independently, preprocessing of the correction grid-code must be made in the place module. The grid-code, indicating which grid cell in each layer has the maximum firing, must be transformed to represent the complete firing of each layer, so as to represent an actual location in the environment. Take the grid cells firing from 0.0 to 0.1667 in figure 6.9 as an example. If the grid cell in the layer with smallest scale have firing = 1, i.e. in the center of the firing field, the most active grid cell in the next layer must have a firing corresponding to a position

halfway to the center of its firing field, and so on. In addition to the firing of the active cell, the firings of the inactive cells must also be derived, in order to know which side of the firing center the subject is located at. The following relations are used to calculate the firing in one layer, from the subsequent smaller scaled layer:

$$\alpha = 1 \texttt{ if } mod^*(2 \cdot j - 1, 3) = i, \ (mod^*(3, 3) := 3) \tag{6.5}$$

$$\alpha = 2 \texttt{ if } mod^*(2 \cdot j, 3) = i, \ (mod^*(3 \cdot N, 3) := 3, N = 1, 2) \tag{6.6}$$

where α indicates which half of the firing field center the subject is located in, i is the active grid cell in the smaller scaled layer , and j ($i, j = 1, 2$ or 3) is the active grid cell in the larger scaled layer. The mod^* function is the same as an ordinary modulo function only with $mod^*(m \cdot n, n) =: n$ rather than zero. The position of subject within the active cell's (j) firing field, given by θ_{λ_k} ($\frac{\pi}{6} \leq \theta_{\lambda_k} \leq \frac{5\pi}{6}$) is:

$$\theta_{\lambda_k} = \frac{(\theta_{\lambda_{k-1}} - \frac{\pi}{6})}{2} + \frac{\pi}{6}, \texttt{ if } \alpha = 1 \tag{6.7}$$

$$\theta_{\lambda_k} = \frac{\pi}{2} + \frac{(\theta_{\lambda_{k-1}} - \frac{\pi}{6})}{2}, \texttt{ if } \alpha = 2 \tag{6.8}$$

From this, the firings of all cells within layer k are calculated from shifted sine functions as in equations 6.2, 6.3 and 6.4, only with period 2π and the active place cell represented by the un-shifted sine function. Note that the above given equations only hold for relative scales of grid layers and number of grid cells in each layer as defined here. In order to work on a different set-up, the above equations would have to be generalized. However, a seemingly better way of integrating correction input is given in the system discussion.

6.4.3 Correction of Grid Firings by 'Allothetic' Input

As mentioned earlier, corrections from allothetic input are represented by place cell IDs. The correction data could come from any

sensor, and the conversion from the sensor's frame of reference to the place cell representation is not considered here. The correction module is simply implemented by correction data being generated as the current grid based place cell ID minus 2. The correction is applied at a few time instances, whenever the rule $mod(t \cdot 1000, a) == 0$ holds (time t given in seconds, a is a user set constant determining the frequency of correction application).

6.4.4 Navigation to Goal Place Cell by Probing

The probing mechanism is implemented much like done in [Erdem and Hasselmo, 2012], only no diffusion process is included and only two probing directions are available in this 1D scenario. When the subject is set to probe, the grid firings are updated in the same way they would if the subject was moving down the probing path. If no goal location is found during probing, the subject is set to move to the right and re-probe after half the probing distance, this time only probing further to the right (as the left probe will only reach already probed place cells). If a correction comes in during the travel to the new probing location, probing is re-initiated, thus probing in both directions. This is done because the system does not know the distance or direction between the corrected place cell and the uncorrected one. The run velocity is kept constant. Abstracted pseudo-code for the navigation module is shown in algorithm 6.2.

Algorithm 6.2 Navigation Module Pseudo-code

```
 1: procedure NAVIGATETOGOAL(goalPC, currentPC)
 2:     constant drivVel, maxProbeLen
 3:     if no goal location is set then
 4:         Start moving randomly
 5:     else if goal location is reached by robot then
 6:         Stay there until new goal arrives
 7:     else if goal location is reached by probe then
 8:         Start moving toward goal
 9:     else if the goal location is not found then
10:         if new goal just came in then
11:             Start probing after new goal
12:         else if probing has started then
13:             if One probe came out empty then
14:                 Start probing in the opposite direction
15:             else if  Both probes came out empty then
16:                 Start moving to the right before reprobing only
17:                 to the right
18:             else
19:                 Continue probing
20:             end if
21:         else if moving toward next probing location then
22:             if correction is received then
23:                 Start probing seance over again
24:             else
25:                 Continue moving toward the next probing location
26:             end if
27:         else if goal is defined but not currently probing then
28:             Start the first probe
29:         end if
30:     end if
31:     return velocity, direction, probingOrNot
32: end procedure
```

6.4.5 System Performance

Figure 6.11 shows the shift in activity levels of one cell from each of the different modules, caused by linear motion. Figure 6.12 shows the probing mechanism of the system with only one goal defined and no corrections. Figures 6.13 and 6.14 show the performance when multiple goals are defined, still without corrections. Figure 6.15 shows the performance of the system when a multitude of corrections are inflicted on the grid system.

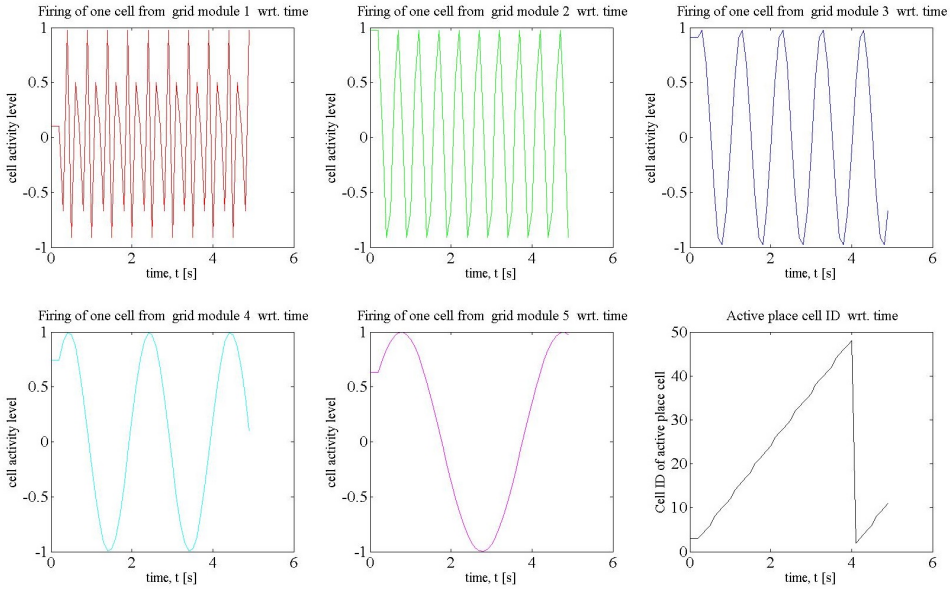

Figure 6.11: *Activity frequencies of the different grid modules. The bottom right figure shows the linear movement of the 'robot'. The remaining figures show the activity level of one grid cell in each module, starting with the grid module with smallest scale in the top left corner. As can be observed from the figures, the frequency of the activity level is far greater in the module with smallest grid scale than the one with the largest (bottom-center figure).*

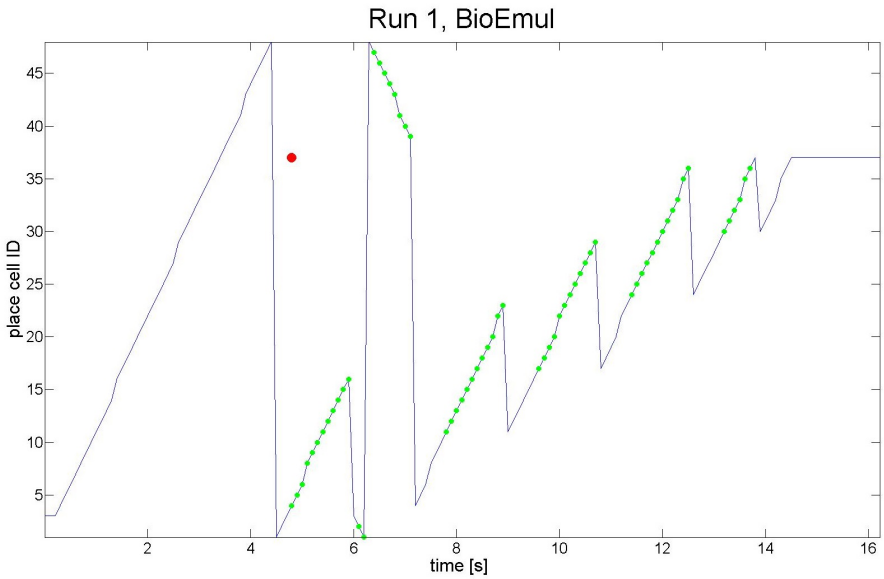

Figure 6.12: *Simulation test of probing mechanism of the system. (Blue line) Place cell occupancy trajectory. (Red circles) goal locations at time of arrival. (Green circles) indicates that probing is being performed (thus, the robot is not actually moving). Note that edges of the enclosure are not defined, so when the virtual robot exits one side it enters the other, analogue to a 1D circular enclosure. Hence the discontinuities. When probing is initiated both directions are probed, in case the goal is in the vicinity. If the goal place cell is not found, the 'robot' moves to the right and reprobes when half a probing-length has been traversed.*

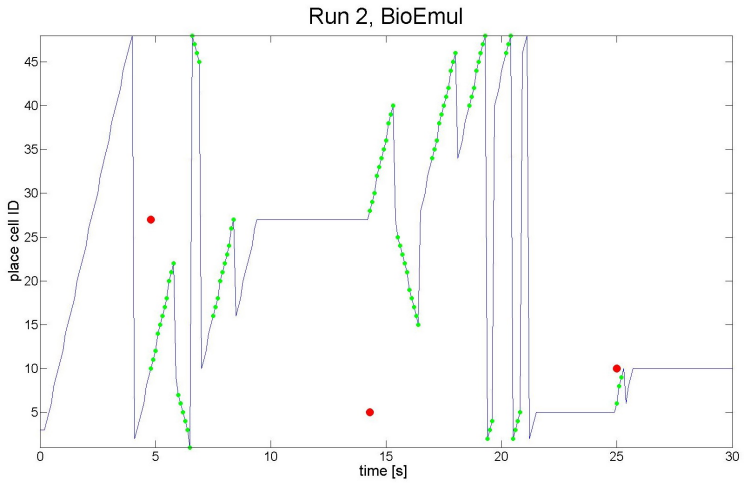

Figure 6.13: *Simulation test of the system with multiple goals defined. (Blue line) Place cell occupancy trajectory. (Red circles) goal locations at time of arrival. (Green circles) indicates that probing is being performed (thus, the robot is not actually moving). The figure shows that when new goals come in, probing starts in order to locate these goal locations.*

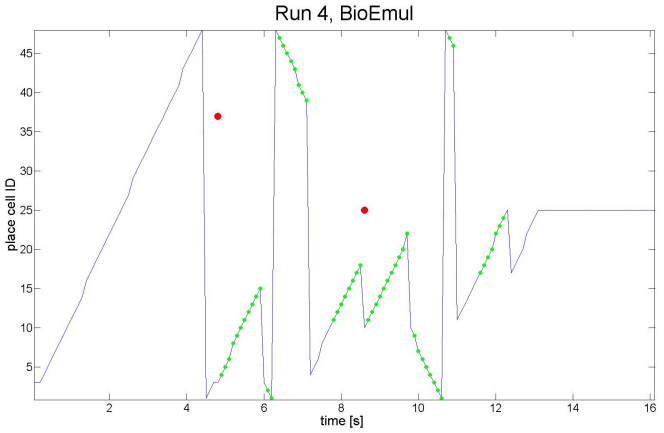

Figure 6.14: *Simulation test of the system with multiple goals defined, when a new goal arrives before the last one is reached. (Blue line) Place cell occupancy trajectory. (Red circles) goal locations at time of arrival. (Green circles) indicates that probing is being performed (thus, the robot is not actually moving). The figure shows that when a new goal comes in, probing restarts in order to locate the new goal. The old goal is discarded.*

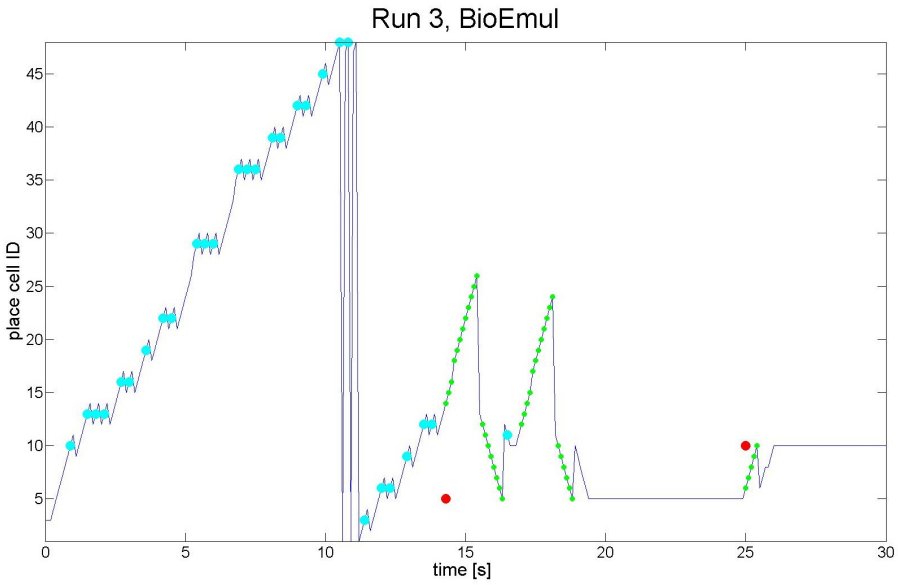

Figure 6.15: *Simulation test of the system with corrections. (Blue line) Place cell occupancy trajectory. (Red circles) goal locations at time of arrival. (Green circles) indicates that probing is being performed (thus, the robot is not actually moving). (Light blue circles) indicate that a correction has come in, which moves the occupied place cell two 'places' to the left (down in the figure). The figure shows that corrections are frequent, and that when a correction comes in between probing instances, the probing seance starts over at the corrected location.*

6.4.6 System Discussion

With respect to the navigation methods introduced above, the implemented system shows most properties similar to the system introduced in [Kubie and Fenton, 2012] in that the tiling property is explicitly defined, and the path integration mechanism resembles that of conjunctive cells, although in a simplified frame. However, goals are defined and associated with place cells, and probing is performed in both directions, as in [Erdem and Hasselmo, 2012].

In the defined map, following the 48^{th} place cell is the 1^{st} place cell, see figure 6.9, resulting in a circular 1D environment. Boundary cells are not included in the design. If they were to be included in a future version, two ways become apparent. The first alternative would include an independent set of boundary cells following the grid cell modules, which fires when the ensemble of grid activity levels are in particular constellation, indicating that a boundary has been reached. The firing of boundary cells would be transferred to the navigation module together with the currently activated head direction cell, which would generate compliant control actions. This would be the most biologically plausible choice, in that the boundary cells are interacting directly with grid cells and head direction cells, cells which are co-localized in the entorhinal cortex. A different approach would entail associating place cells with boundaries, so that a boundary signal is sent to the navigation module when certain place cells are reached. The navigation module would still need head-direction information to know which directions remain unobstructed.

The system method for handling corrections is overly complicated. The calculations used to calculate the grid cell firing levels corresponding to the corrected place cell become excessive if the grid firings of a place cell are not represented as just the most active grid cell in each grid module, but rather what the modules' firing levels are. This way, when a correction comes in, the grid cells associated with the correction place cell can be fed right back to the grid modules, without intermediate calculations. Such a mechanism of activity association is probably also closer to the activity drive of biological

systems.

The only noise inflicted on the designed system is the bias caused by drift in odometry. The bias is not modelled and is only handled by the correction with idealized 'visual' measurements. Thus, in between corrections the location estimate will drift uncontrollably, and the estimate drifts accordingly from the true location. Furthermore, the state of the system is deterministically represented, a set-up which renders the system design useless in any practical implementations. Inclusion of noise modelling and -detection and probabilistic representation of state will be further discussed in a comparison between the Kalman filter and the designed system, given in the next chapter. Inclusion of noise handling mechanisms and probabilistic representations of location in the designed system are tasks left for the future.

6.5 Chapter Summary

In this chapter the biological way of updating the current position by integration, known as path integration, has been introduced. How animals use both visual- and self-motion information to decide their location was discussed, however with limited elaboration on how data from the different information sources are fused, due to limited knowledge of how this is done in nature. Two models of biological navigational strategies were given, both relying on a linear probing mechanism. Finally, a simple system was implemented, which visualizes the biological mechanisms described previously in the chapter. While visualizing some key properties, the system lacks mechanisms for coping with noise and uncertainty. How these mechanisms are represented in animals remain undisclosed.

In the following chapter connections between the biological mechanisms for localization and navigation and state-of-the-art mechanism in robotics will be given, revealing more of the characteristics of the biological mechanisms.

Part III

Comparisons and Implications

Chapter 7

Comparison of Localization Methods in Biology and Robotics

7.1 Spatial Representation

7.1.1 Maps

The three map types commonly used in robotics are topological-, feature- and grid-based maps. Biological systems employ versions of all three types of maps. The local environment representation system uses the grid-cell metric map fused with a sort of feature based map interpreted by the vision processing system, to represent the location (place cell) in the environment. The connection between individual local maps is done by a higher level topological map. However, the combined feature- and grid-based map applied in biology differ from the maps applied in robotics.

Feature map The feature based representations in biological systems have not been discussed previously in this thesis, thus it will not be elaborated. However, the irreproducable ability of biological systems to capture the 'gist' (classification) of a scene and to identify objects and landmarks has been thoroughly documented [Siagian and Itti, 2009].

Grid map The biological grid- and place cell map show many similarities to the grid maps used in robotics. The activity level of the place cells indicate if they are occupied or not, just like in robotic occupancy-grid maps. Furthermore, they both facilitate computation of shortest path in navigation and they are both dependent on good motion models to accurately update location with movement.

The main difference between metric maps in robotics and in biology is the way relative locations are represented. In robotics, relative distances between grid vertices are given in a Cartesian frame, with only one grid used. In biology several grids are used to encode a location, determined by the overlap of grid vertices. Relative distances are represented by the ensemble of grid scales. The grid-cell map only needs a few grid-cell modules to represent a large environment, while robotic metric maps need one variable for each grid vertex. However, because each location represented by the grid-cell map is associated with a place cell, more variables are needed to represent a local environment. The remapping paradigm which reuses the grid cells in new environments, indicate that the biological metric map is used universally across environments. The metric of robotic maps is anchored to the single local environment.

7.1.2 Hierarchies

Biological localization shows significant similarities with hierarchical methods currently applied in robotics, only with extensively more obscure and advanced connections and mechanisms. The literature on hierarchical methods in robotics show that most research present hierarchies with only two levels, commonly as one top-level topological- and one bottom-level feature- or grid-based layer. An obvious parallel can be drawn to the combination of the distributed landmark based top-level and the lower-level grid and place cell firings in biological localization. In addition, the literature suggests biological global maps are created in a top-down manner, like most robotic hierarchical maps, referencing the global remapping procedure applied

when salient landmarks are encountered.

The recognition of the grid-cell hierarchy in the robotics literature is less appreciable. The metric hierarchy in [Menegatti et al., 2004] (section 3.2.2), compares frequency components amounting to increasing levels of localization-resolution for each comparison. The instinct would be to draw parallels to the increasing resolution of grid scale in subsequent modules in the EC. However, an important distinction becomes apparent: in the hierarchy of grid-cell modules, candidate locations are not mutually adjacent in the environment. Because one grid cell's firing is distributed in the local environment, grid-cell firings of all modules are necessary for even approximate localization.

The robot motivation for representing space hierarchically is reduction of computational complexity with respect to metric maps, and heightened localization accuracy with respect to a topological map only. The same reasoning may underlie the hierarchical biological map. With the entire environment (the earth) being represented by a set of grids anchored to one landmark, the number of grid cells needed to define all place cells would be infinitely large. Furthermore, by only using a topological map to represent the whole environment, the accuracy of the location estimate might have been to low for animals to operate with.

7.2 Localization

7.2.1 Uncertainty

The question now becomes how the activity patterns in the attractor networks, representing the location of the animal in the environment relates to the probabilistic definition of location used in robotics. How does the biological systems in reality account for uncertainty? The attractor network in itself does not represent the uncertainty in an animal's location [Friston, 2010], in fact, no research has been presented which clarifies how animals form these coherent activity patterns in

the presence of uncertainty. Furthermore, the representation of multiple beliefs, such as in non-parametric filters used in robotics, have not been found. One reason for this, argued in [Milford et al., 2010], is that studies on animals have been conducted in small enclosures, which do not present ambiguous situations. Moreover, only activity in small populations of the total grid- and place cell sets can be measured simultaneously [Milford et al., 2010]. Even without evidence on how animals represents the uncertainty in their environment, the belief is that animals represent sensory information probabilistically. The belief comes from an increasing body of evidence showing that human perceptual computations are 'Bayes' optimal' (it takes into account the uncertainty in the available information at each stage of processing). The functional forms of the probability distributions have not yet been disclosed [Knill and Pouget, 2004]. Thus, the direct comparison between representations of beliefs and posteriors in biology and robotics becomes a task for the future.

Although the representation of uncertainty is unknown, the earlier disclosed information indicates where in the overall network the uncertainty might be represented. The attractor network in the different grid modules are good candidates for representing the motion model of biological systems, and the resulting activity of the active place cell set (and the discrete attractor network between place cell sets) for representing the measurement model. The activity levels of these cells are not binary (i.e. on or off), the cells can take values in a continuous range. Thus, by allowing multiple beliefs to be represented by the activity level of different cells, uncertainty can be represented in the attractor networks. The motion noise manifested in the grid cell networks is transferred to the place cell network by summation of the activity in all modules. Consequently, the uncertainty in location caused by both measurement and motion noise is represented in the place cell network. Note however, that the way motion noise is represented in the continuous attractor network of grid cells is probably not straightforward, because of the distributed nature of the grid cells. As was indicated in the error handling mechanism, small activity perturbations in one of the modules may cause a large displacement in the represented location.

RatSLAM(Rat- Simulaneous localization and mapping), a research project at Queensland University of Technology, led by Michael J. Milford, presented a robotic system emulating place cells a decade ago [Milford et al., 2004]. Since then, the system has been upgraded to fit the new discoveries made. In 2010, they presented a version which integrated the use of grid cells, through which they showed that probabilistic representations of sensor information could be used in the attractor network, and that multiple location hypotheses can be represented in the same network [Milford et al., 2010]. Thus, the work shows that it is fully possible to represent probabilistic measurements and multiple hypotheses in the biological localization frame, the question is just how this is done in nature.

Even without knowing how beliefs are represented in biology, observations of which mechanisms are involved in solving the different localization problems given in chapter 2, can be made from the information disclosed in part II.

7.2.2 Global Localization and the Kidnapped Robot Problem

In chapter 2 the global localization problem was given as:

> The initial pose of the robot is unknown. The robot is placed somewhere in its environment, but the exact location is unknown.

The kidnapped robot problem is a version of this problem, only during operation the robot can get kidnapped and teleported to another location.

Biological Solution. Visual information is used to identify landmarks and thus the currently occupied local environment (attractor basin), from which the corresponding grid and place cell activity patterns are extracted. The firings are obtained by measuring the distance to a local landmark with known distance and angle to the reference point, and subsequently calculating the activity level of the

active grid cells, as illustrated by the equations given in 5.3.4. If the animal were to be teleported to another local environment, the exothetic input (including vision) would cause a change of basin in the discrete attractor network of the place module, from which the procedure used for global localization would be applied.

7.2.3 Position Tracking

In chapter 2 the position tracking problem was given as:

> Assumes that the *initial* robot pose is known. The location of the robot can be acquired by accommodating the motion noise.

Biological Solution. Self motion is integrated by means of the grid modules and the head direction network. Each grid module is thought to integrate velocity independently, by means of several conjunctive cell layers, one for each head direction. Erroneous localization estimates of individual modules caused by integration errors, will be handled by a highly sophisticated error handling mechanism system. This system, however, only handles noise which is uncorrelated across the grid-modules. Errors which manifest in the ensemble of grid modules, usually caused by drift and noise in the self motion input, are corrected by input from vision and other exothetic sensors (smell,touch, etc.). Either the exothetic and self-motion representations are weighted in some way to obtain the best estimate, or, as done in young children, the exothetic representation is assigned full weight.

7.2.4 BioEmul Versus the Kalman Filter

A simple comparison between the system's and the Kalman filter's tracking performances is made (through simulations), to illustrate the shortcomings of the BioEmul system in any real application. Updating the 1D position by integrating the velocity, as done in the

BioEmul system, can be modelled by a simple discrete linear model;

$$x(k+1) = \Phi(k)x(k) + \Delta(k)u(k) + \Gamma w(k) \tag{7.1}$$

$$y(k) = H(k)x(k) + v(k) \tag{7.2}$$

where x is the position, Δ is the time-step ($= 0.1$ in simulations), Γ is the process-noise matrix ($= 1$ in simulations), y is the measurement, while w and v are the modelled process noise and measurement noise, respectively. In the linear model for position tracking as in BioEmul, Φ and H are constants equal to one.

A plant model given by the equations above is implemented in Simulink, together with a Kalman filter using a correct plant model, only the variance of the added noise is assumed unknown. The plant's input velocity and position measurement are sent to both the Kalman filter and the BioEmul system, as in figure 7.1. The Kalman filter equations are given in appendix A, table A.1 [Mendel, 1995].

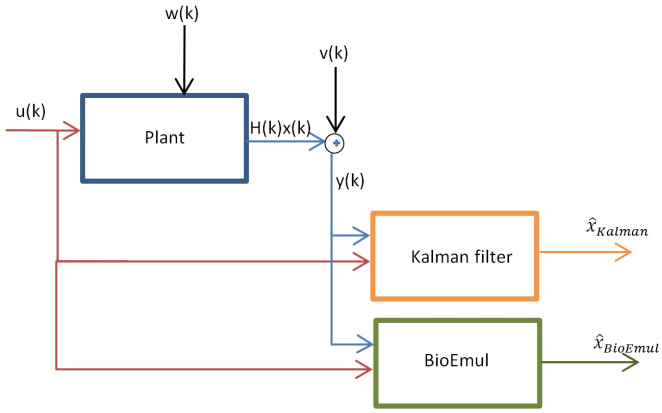

Figure 7.1: *The plant's input and measurement is sent to both the Kalman filter and the BioEmul system. Both process noise $w(k)$ and measurement noise $v(k)$ is added.*

The performance of the BioEmul system is comparable to a Kalman filter running at every time-step, with measurements only arriving at

some time-steps. Simulated position-estimate error for both systems is shown in figure 7.2. In the simulation shown, no measurement noise was added, while the variance of process noise was set to $\sigma = 0.8$. The design values were $R = 0.000000001$ (measurement received), $R = 1000000$ (no measurement) and $Q = 4$. Measurements were set to arrive every five seconds (starting at $t = 0$ s). From the figure it can be seen that in-between the arrivals of the measurements, the Kalman filter does not account for the process noise. This is because the Kalman-gain is equal to zero due to infinitely noisy measurements (which is analogous to the measurement not arriving). Thus, it only corrects to the true location whenever the measurements arrives, similar to the bioEmul system performance.

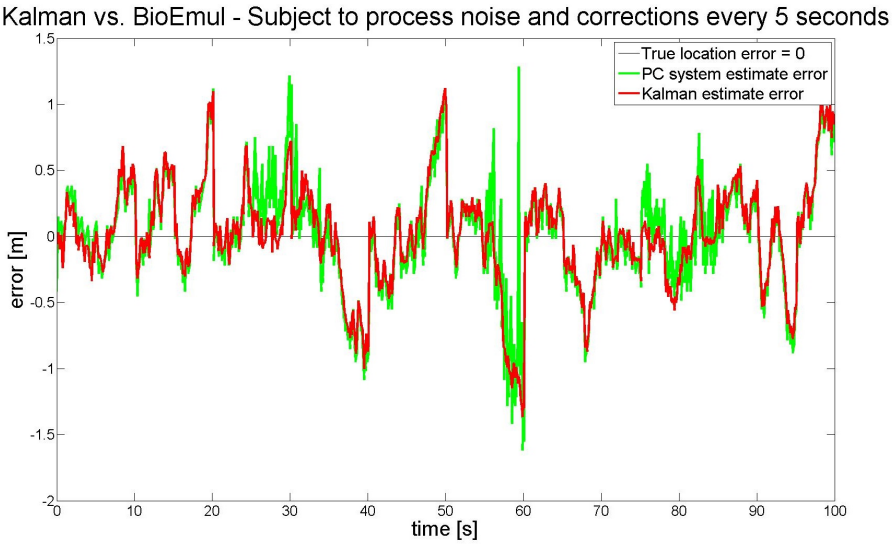

Figure 7.2: *Comparison of simulated performance - BioEmul and the Kalman filter, with measurements arriving every* 50^{th} *time-step. The figure shows the localization error of the two systems as a function of time. As can be seen, the two systems show similar position estimate deviations from the true value.*

When the measurements arrive at every time-step, the Kalman filter grossly outperforms the BioEmul system. When measurement noise is added, the superior performance of the Kalman filter is even

clearer. Figure 7.3 shows the simulated estimation error when measurements arrive at every time-step, measurement noise variance is $\sigma = 0.16$ and process noise variance is $\sigma = 0.0081$. Design values are set to $R = 700$ and $Q = 4$.

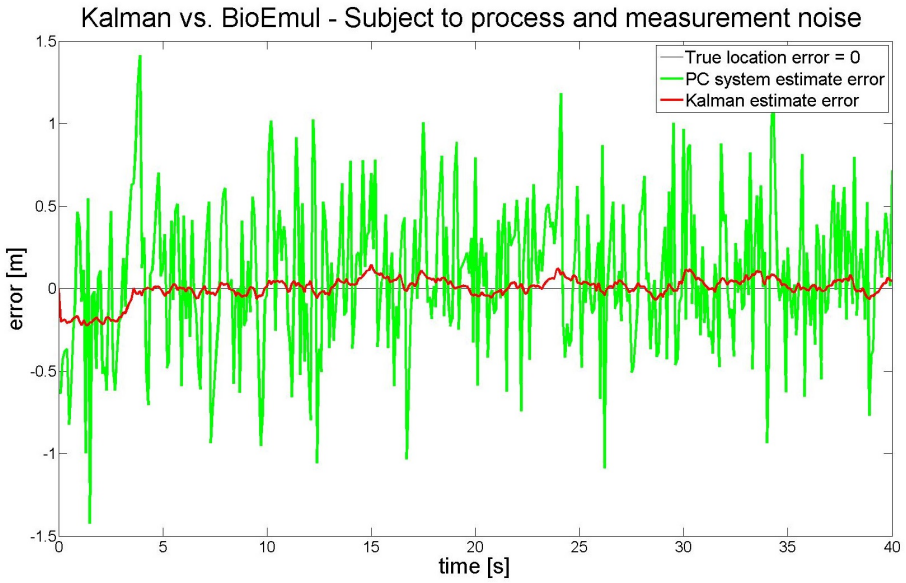

Figure 7.3: *Comparison of simulated performance - BioEmul and the Kalman filter, with measurements arriving every time-step. The figure shows the localization error of the two systems as a function of time. As can be seen, the Kalman filter positon estimate is far better than that of the BioEmul system.*

The performed simulations visualize the importance of including a noise handling mechanism in future bio-inspired systems. Furthermore, they visualize the imperfection of children's sensor fusion mechanism. Measurements, in both biological and technological systems are inherently noisy and ambiguous, which should be accounted for. For animals, this improved performance is gained through learning. For robotic systems emulating the biological mechanisms, either a learning paradigm must be set up like in biology, or a noise compensation filter, such as the Kalman filter, must be included in the system design.

7.3 Navigation

The payoff function of POMDP in robotics (chapter 2) can be set by the designer, allowing a great deal of flexibility in the navigation strategy, including obstacle avoidance. Similarly, the reward-signal diffusion process in biological navigation, as proposed by Erdem and Hasselmo, enables the linear probing mechanism to avoid obstacles. Again, however, the stochastic representation of states and measurements separates state-of-the-art robot navigation strategies from the proposed biological strategies. As described in chapter 2, the POMDP allows for both believed states and measurements to be estimated by means of probability distributions. Thus, when the value function is to be computed, it is computed for all states in the state (*belief*) space. Contrarily, the linear probing mechanisms assume that the state of the animal is known, i.e. deterministically represented. The process of linear probing, as it is now, would fall under the class of robot navigational models known as biomimetic ("simulating biology") models inspired by biology. These methods are rarely chosen any more for robotic applications, because they do not consider the uncertainty in perceived states and measurements. However, attempts have been made to fuse the two models to gain biologically plausible, stochastic navigation strategies [Diard et al., 2004].

7.4 Summary

The biological system's way of providing a metric component to their spatial representations, was in the beginning of the chapter found unrecognizable in the spatial representations commonly used in robotics. Both state-of-the-art robotic systems and animals show hierarchical representations. However, while robotic systems usually include only one lower-level representation, the biological hierarchy uses both feature-based and metric-based local representations of space. The different elements involved in the biological solutions to the localization problems given in chapter 2 were presented. A direct comparison between the biological solutions and state-of-the-art robotic solution was not made, due to the fact that biological representation of uncertainty remains unknown. The shortcomings of

a deterministic biological localization system in any practical implementations were demonstrated through simulations of performance of the implemented biologically inspired system BioEmul and a Kalman filter. The simulations showed that when the systems are subjected to noise, the Kalman filter outperforms the BioEmul system, considering BioEmul does not incorporate uncertainty. At the end navigational strategies were compared, and again the representation of uncertainty was shown to be the most significant difference between the believed biological strategies and robotics.

Chapter 8

Discussion

In this thesis the spatial representation of the brain has been investigated, accompanied by localization and navigation mechanisms associated with this representation. The biological spatial representation, localization and navigation have been compared to their equivalents in state-of-the-art robot systems, including discussions considering the findings of the thesis. Hence, the following discussion will focus on approaches taken.

8.1 Part I: Robot Spatial Representation and Localization

In part I the different map types commonly used in robotics, together with state-of-the-art methods for localization and navigation were presented. In addition, different hierarchical representations of space were disclosed.

The presented material on robotics focus mainly on describing state-of-the-art methods for localization and the way these methods represent and update their probabilistic position estimates. In retrospect, since it is not known how biological systems represent uncertainty, other factors contributing to the performance of the robotic systems should have been weighted more heavily; performance in different situations, which mechanisms are involved in the localization process

(e.g. how maps are represented in the computer), and different limitations encountered. With these considerations accounted for, better comparisons could have been made.

8.2 Part II: Biological Spatial Representation and Localization

In part II the basic elements contributing to biological representation of space were given, together with their functionality. The spatial representation's connection to memory was described, and the attractor dynamic model used to describe the update of the activity patterns in grid- and place cell networks was presented. Moreover, the geometric foundation for the grid cell spatial firing patterns was disclosed, and it was shown how many grid cells were needed to uniquely define place cells covering the local area. Simulations confirmed that a linear increase in active grid cells resulted in an exponentially larger area uniquely defined by the active grid-cell modules, as claimed in [Sreenivasan and Fiete, 2011]. In chapter 6 proposed navigation methods using the biological spatial representation were exposed. The navigation models presented do not have as much experimental support as the described mechanisms, they are only proposals on how the spatial representation may be exploited in navigation. A simple 1D system, BioEmul, was implemented to visualize the disclosed mechanisms for biological position tracking and navigation.

The investigations made in this thesis do not include any information about how mapping is performed in biological systems. This choice was made to limit the scope and rather focus on presenting the basic components in a systematic way. However, mapping is a huge part of many robotics systems, such as those trying to solve the *simultaneous localization and mapping* (SLAM) problem. Moreover, the mapping paradigm of biological systems might reveal important characteristics of the biological systems, which may further help in the understanding of why such systems surpass state-of-the-art robotic

systems. The same can be said for the choice not to focus on the visual processing of biological creatures. As an extension of this work, these areas should be investigated.

In the biological spatial representation system, more brain parts contribute than those presented here. The studied literature show that other parts of the brain are likely to be part of the spatial representation system. However, the focus of research has been on the entorhinal cortex and the hippocampus, as they are believed to form the basic operations. Thus, not much of the literature disclose what the contributions of other likely participating brain parts are. Theories on the tasks of these seemingly secondary contributors have not been discussed in this thesis, so as not to present functionality with little supporting experimental evidence.

As a whole, the biological part of the thesis present the basics of the biological spatial representation system. However, functionality essential to understanding what makes the biological systems work so incredibly well, has not been disclosed here. This functionality could perhaps be derived from the published literature. Alternatively, the functionality currently could still lie hidden in the neurons of our brains, and hopefully future research will reveal it.

8.3 Part III: Comparison of Biology and Robotics

In part III the biological and robotic mechanisms for spatial representation, localization and navigation were compared. The biological local map was found to be fundamentally different from similar maps in robotics. However, in order to distinguish the situations in which the biological map would outperform robotic maps, implementations of the two map types should be made and compared in different settings. As for localization and navigation approaches, the comparison suffers from the unknown uncertainty representation in biological systems. Investigating how the different robot localization methods can be altered to be applicable to the biological representation of space,

and determining which of the approaches is best suited, is a reasonable extension of the work done here.

The comparison of the BioEmul system and the Kalman filter was made to demonstrate the effect of not representing uncertainty. It was shown that the BioEmul system, without handling noise, did not estimate location precisely. An interesting extension would be to include a noise compensating mechanism in the design and make a new comparison.

Considerations of how the two systems handle the different sensor data sequences have not been addressed in this thesis. An important sensor in biological systems is vision, which was not discussed here. The focus was rather on presenting an elaborate description of the brain's intrinsic spatial representation system. Hence, sensor data consideration is a taks left for the future.

All in all the biological system show some interesting properties which may be beneficial to emulate in robotics. Of course, the usefulness of certain mechanisms is not apparent. The grid cell network is far more complicated than representing metric measures with Cartesian coordinates. However, it allows for integration errors to be easily detected, a feature which may be desirable in many applications. Further investigations of the biological mechanism might reveal more features which would improve the performance of localization systems, even if a full biological system emulation does not provide superior performance.

Chapter 9

Conclusion

This thesis has systematically presented how biological systems represent the local environment, how different localization problems are solved using this representation, and how the spatial representation facilitates navigation. It was shown that the grid cells serve as the metric components in subconscious biological maps, just like the metric system serves as the metric of conscious maps. Furthermore, the brain's spatial representation showed significant hierarchical properties, similar to hierarchical maps used in robotics. Indications were made as to how localization and navigation is performed with the given spatial representation. However, a full display of these mechanisms could not be made, due to the lack of knowledge on how biological systems represent uncertainty.

The biological space representation was compared to the different maps applied in robotics, and showed most resemblance with the popular occupancy grid maps. The comparison of localization and navigation methods suffered from the unknown biological representation of uncertainty. The importance of representing uncertainty was demonstrated through a simulation comparison between the created biologically inspired system, BioEmul, and the Kalman filter, in which the Kalman filter grossly outperformed the biologically inspired system. Hence, robotic systems aiming at emulating biological mechanisms must for the time being use an established method for representing uncertainty, until the biological representation is dis-

closed.

In conclusion, this thesis serves as a good first-step in the association of biological- and robotic localization methods and facilitates further work in this regard.

Chapter 10

Future Work

The investigations made in this study facilitates an abundance of future focus areas. These include:

- Design a biologically inspired system which explicitly incorporates the attractor network dynamics.
- Investigate how motion uncertainty can be represented in the grid cell networks.
- Investigate which known representations of uncertainty are best suited for the biological system. Such an investigation and comparison of representations may indicate which experiments should be conducted to establish the true uncertainty representation of the brain.
- Implement a system with the best uncertainty representation, and compare it to state-of-the-art methods in robotics to further reveal strengths and weaknesses of the biological system
- Design a navigation system based on the mechanisms made in this thesis, and compare its performance to that of state-of-the-art robot navigation approaches.
- Investigate how learning and mapping is done in biology, which may reveal more strengths of the biological systems.

- Take a closer look at the error handling mechanism introduced, and determine if such a mechanism can be beneficial in other robotic applications.

Bibliography

[Barry and Bush, 2012] Barry, C. and Bush, D. (2012). From A to Z: a potential role for grid cells in spatial navigation. *Neural systems & circuits*, 2(1):6.

[Blanco et al., 2006] Blanco, J. L., Gonzalez, J., and Fernández-madrigal, J. A. (2006). Consistent Observation Grouping for Generating Metric- Topological Maps that Improves Robot Localization. In *IEEE International Conference on Robotics and Automation*, number May, pages 818–823.

[Burgess and O'Keefe, 1996] Burgess, N. and O'Keefe, J. (1996). Neuronal computations underlying the firing of place cells and their role in navigation. *Hippocampus*, 6(6):749–62.

[Buzsáki and Moser, 2013] Buzsáki, G. and Moser, E. I. (2013). Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, 16(2):130–8.

[Courbon et al., 2008] Courbon, J., Mezouar, Y., Eck, L., and Martinet, P. (2008). Efficient hierarchical localization method in an omnidirectional images memory. *2008 IEEE International Conference on Robotics and Automation*, pages 13–18.

[DARPA, 2012] DARPA (2012). Darpa research on electronic neural architectures cited. `www.darpa.mil/NewsEvents/Releases/2012/02/06.aspx`.

[Diard et al., 2004] Diard, J., Bessière, P., and Mazer, E. (2004). A theoretical comparison of probabilistic and biomimetic models of

mobile robot navigation. *Robotics and Automation, 2004. ...*, (April):933–938.

[Erdem and Hasselmo, 2012] Erdem, U. M. and Hasselmo, M. (2012). A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *The European journal of neuroscience*, 35(6):916–31.

[Friston, 2010] Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews. Neuroscience*, 11(2):127–38.

[Fyhn et al., 2004] Fyhn, M., Molden, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2004). Spatial representation in the entorhinal cortex. *Science (New York, N.Y.)*, 305(5688):1258–64.

[Giocomo et al., 2011] Giocomo, L. M., Moser, M.-B., and Moser, E. I. (2011). Computational models of grid cells. *Neuron*, 71(4):589–603.

[Hafting et al., 2005] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–6.

[Ivan, 1994] Ivan, A. (1994). Mobile Robot Visual Mapping and Localization : A View-Based Neurocomputational Architecture That Emulates Hippocampal Place Learning. *Neural Networks*, 7:1083–1099.

[Jauffret et al., 2012] Jauffret, A., Cuperlier, N., Gaussier, P., and Tarroux, P. (2012). Multimodal Integration of Visual Place Cells and Grid Cells for Navigation Tasks of a Real Robot. pages 136–145.

[Jeffery, 2011] Jeffery, K. J. (2011). Place cells, grid cells, attractors, and remapping. *Neural plasticity*, 2011:182602.

[Knill and Pouget, 2004] Knill, D. C. and Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in neurosciences*, 27(12):712–9.

[Kubie and Fenton, 2012] Kubie, J. L. and Fenton, A. a. (2012). Linear look-ahead in conjunctive cells: an entorhinal mechanism for vector-based navigation. *Frontiers in neural circuits*, 6(April):20.

[Lisien et al., 2005] Lisien, B., Morales, D., Silver, D., Kantor, G., Rekleitis, I., and Choset, H. (2005). The hierarchical atlas. *IEEE Transactions on Robotics*, 21(3):473–481.

[McNaughton et al., 2006] McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., and Moser, M.-B. (2006). Path integration and the neural basis of the 'cognitive map'. *Nature reviews. Neuroscience*, 7(8):663–78.

[Mendel, 1995] Mendel, J. M. (1995). *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Signal Processing Series. Prentice Hall.

[Menegatti et al., 2004] Menegatti, E., Maeda, T., and Ishiguro, H. (2004). Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267.

[Milford et al., 2010] Milford, M. J., Wiles, J., and Wyeth, G. F. (2010). Solving navigational uncertainty using grid cells on robots. *PLoS computational biology*, 6(11):e1000995.

[Milford et al., 2004] Milford, M. J., Wyeth, G. F., and Prasser, D. (2004). RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping. *IEEE International Conference on Robotics and Automation*, pages 403–408.

[Moser et al., 2008] Moser, E. I., Kropff, E., and Moser, M.-B. (2008). Place cells, grid cells, and the brain's spatial representation system. *Annual review of neuroscience*, 31:69–89.

[Rady et al., 2010] Rady, S., Wagner, A., and Badreddin, E. (2010). Hierarchical localization using entropy-based feature map and triangulation techniques. *2010 IEEE International Conference on Systems, Man and Cybernetics*, pages 519–525.

[Samu et al., 2009] Samu, D., Eros, P., Ujfalussy, B., and Kiss, T. (2009). Robust path integration in the entorhinal grid cell system with hippocampal feed-back. *Biological cybernetics*, 101(1):19–34.

[Siagian and Itti, 2009] Siagian, C. and Itti, L. (2009). Biologically Inspired Mobile Robot Vision Localization. *IEEE Transactions on Robotics*, 25(4):861–873.

[Solstad et al., 2008] Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B., and Moser, E. I. (2008). Representation of geometric borders in the entorhinal cortex. *Science (New York, N.Y.)*, 322(5909):1865–8.

[Solstad et al., 2006] Solstad, T., Moser, E. I., and Einevoll, G. T. (2006). From Grid Cells to Place Cells : A Mathematical Model. *Hippocampus*, 16:1026–1031.

[Sreenivasan and Fiete, 2011] Sreenivasan, S. and Fiete, I. (2011). Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature neuroscience*, 14(10):1330–7.

[Stensola et al., 2012] Stensola, H., Stensola, T., Solstad, T., Frø land, K., Moser, M.-B., and Moser, E. I. (2012). The entorhinal grid map is discretized. *Nature*, 492(7427):72–78.

[Thrun, 2008] Thrun, S. (2008). Simultaneous localization and mapping. *Robotics and cognitive approaches to spatial mapping*, pages 13–41.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. First edition.

[Tomatis et al., 2003] Tomatis, N., Nourbakhsh, I., and Siegwart, R. (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1):3–14.

[Tully et al., 2007] Tully, S., Moon, H., Morales, D., Kantor, G., and Choset, H. (2007). Hybrid localization using the hierarchical atlas. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2857–2864.

[Wang et al., 2005] Wang, J., Cipolla, R., and Zha, H. (2005). Vision-based global localization using a visual vocabulary. *IEEE International Conference on Robotics and Automation*, (April):4230–4235.

[Weitzenfeld et al., 2012] Weitzenfeld, A., Fellous, J.-M., Barrera, A., and Tejera, G. (2012). Allothetic and idiothetic sensor fusion in rat-inspired robot localization. 8407:84070E–84070E–8.

Appendix A

The Kalman Filter Equations

The equations used to implement the discrete Kalman filter are shown in table A.1. The plant model underlying the equations is as given in equations 7.1 and 7.2.

Table A.1: *The discrete Kalman filter equations [Mendel, 1995].*

Kalman Filter equations

Design matrices:

$$\mathbf{Q}(k) = \mathbf{Q}(k)^T \geq 0 \tag{A.1}$$

$$\mathbf{R}(k) = \mathbf{R}(k)^T \geq 0 \tag{A.2}$$

Initial conditions

$$\bar{\mathbf{x}}(0) = \mathbf{x}_0, \bar{\mathbf{P}}(0) = \mathbf{P}_0 \tag{A.3}$$

Kalman gain:

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^T(k)[\mathbf{H}(k)\bar{\mathbf{P}}(k)\mathbf{H}^T(k) + \mathbf{R}(k)]^{-1} \tag{A.4}$$

State estimate update:

$$\hat{\mathbf{x}}(k) + \ \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\bar{\mathbf{x}}(k)] \tag{A.5}$$

Error covariance update:

$$\hat{P}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]\bar{\mathbf{P}}(k)[\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]^T \tag{A.6}$$

State estimation propagation:

$$\bar{\mathbf{x}}(k+1) = \Phi(k)\hat{\mathbf{x}}(k) + \Delta(k)\mathbf{u}(k) \tag{A.7}$$

Error covariance propagation:

$$\bar{\mathbf{P}}(k+1) = \Phi(k)\hat{\mathbf{P}}(k)\Phi^T(k) + \Gamma(k)\mathbf{Q}(k)\Gamma^T(k) \tag{A.8}$$

Appendix B

Motion- and Measurement-models Applied in Robotics

In order to implement any of the described filters, the motion and measurement models must be defined. Both models can be represented in several ways, some of which will be presented here.

B.1 Motion Models

The *motion model* comprises the state transition probability $p(x_t|u_t, x_{t-1})$, an essential contributor to the prediction step of the Bayes filter. Probabilistic robotics treat robot kinematics as the stochastic variables they intrinsically are: the outcome of a control is inherently uncertain. Thus, determining the motion model becomes significantly more complex than for the deterministic case.

In a planar environment with a rigid-body mobile robot in consideration, the pose of the robot is described by the following vector:

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

where x and y are the planar coordinates and θ the angular orientation of the robot. Two motion models will now be introduced, which assumes different specifications for the control data.

Velocity Motion-model Assumes that the robot can be controlled through one rotational- and one translational velocity (u_t). Models the motion error as either a Gaussian or triangular distribution.

Odometry Motion-model Uses the odometry measurements as the basis for calculating the robot's motion over time. Because odometry data is only available in retrospect, the odometry motion model cannot be used for planning and control of the robot. Odometric measurements u_t are expressed by three parameters, an initial rotation, followed by a translation, and a final rotation. All three parameters are subject to (independent) noise.

Both the above described models can incorporate the added information of a map of the environment. An approximation to the complex map based motion model allows for tolerable computational efforts. The approximation factorizes the map based motion model into two components:

$$p(x_t|u_t, x_{t-1}, m) = \eta \frac{p(x_t|u_t, x_{t-1})p(x_t|m)}{p(x_t)} \tag{B.1}$$

where $p(x_t|m)$ is the *consistency* of pose x_t with the map m [Thrun et al., 2005].

B.2 Measurement Models

Measurement models describe how the sensor measurements are generated in the physical world; specifically, how noise affects the measurements. Probabilistic robotics models noise explicitly and account for the inherent uncertainty in the robot's sensors. The measurement model is defined as the conditional probability distribution $p(z_t|x_t, m)$, where z_t is the measurements at time t, and as before

m is the map and x_t is the state at time t. The measurement z_t is usually a sequence of K values;

$$z_t = \{z_t^1, ..., z_t^k, ..., z_t^K\}$$

The probability $p(z_t|x_t, m)$ is obtained as the product of the individual measurement likelihoods (*independence assumption*):

$$p(z_t|x_t, m) = \prod_{k=1}^{K} p(z_t^k|x_t, m) \tag{B.2}$$

Beam Models of Range Finders. Incorporates four types of measurement errors:

- small measurement noise, modelled as Gaussians in the sensing range, p_{hit}
- errors due to unexpected objects, modelled as exponential distributions, p_{short}
- errors due to failure to detect objects, modelled as point-mass distributions centered at the far limit of the sensing range, p_{max}
- random unexpected noise, modelled as a uniform distribution over the sensing range, p_{rand}

The four distributions are mixed by a weighted average, where $z_{hit} + z_{short} + z_{max} + z_{rand} = 1$, to create the complete measurement model;

$$p(z_t^k|x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_t^k|x_t, m) \\ p_{short}(z_t^k|x_t, m) \\ p_{max}(z_t^k|x_t, m) \\ p_{rand}(z_t^k|x_t, m) \end{pmatrix} \tag{B.3}$$

The intrinsic parameters of the model, collectively denoted Θ, include among others the mixing parameters z_{hit}, z_{short}, z_{max} and z_{rand}. These parameters will naturally greatly impact the correctness of the

measurement model. There are several ways to adjust the parameters, including by hand and finding the maximum likelihood of a reference data set [Thrun et al., 2005].

Likelihood Fields for Range Finders The beam-based model shows a *lack of smoothness* and is computationally involved. The alternative method *likelihood field* overcomes these limitations. The *likelihood field* method lacks a plausible physical explanation, but works well in practice.

The method uses the nearest distance in 2-D coordinates to model the probability $p(z_t^k|x_t, m)$. Similar to the beam model, it assumes three types of noise and uncertainty; measurement noise (Gaussians), failures (point-mass distribution) and unexplained random measurements (uniform distribution). An important characteristic of the method is that it disregards measurements at the far distance of the sensing range; hence, it ignores information pertaining to free space. In addition, it fails to consider occlusions in the interpretation of range measurements [Thrun et al., 2005].

Correlation-Based Measurement Models *Correlation-based measurement models* measure correlations between the measurement and the map. A popular technique is called *map matching*. It transforms maps into *occupancy maps* (binary occupancy-indicator), and compares a local map, m_{local}, created from several scans with the global map, m. The more similar m_{local} and m, the larger $p(m_{local}|x_t, m)$. The map matching technique explicitly considers free-space in the scoring of two maps.

Feature-based Measurement Models *Feature-based methods* extract features from the measurements and base the model on these features. This technique significantly reduces dimensionality of the problem, and inherently the computational complexity. From the

sensor measurement range (r) and *bearing* (heading, ϕ) to nearby objects are extracted. In addition, a *signature* (s) may be generated. A signature is a numerical value (or vector), characterizing the type of landmark an object embodies. The feature vector becomes a collection of triplets. Many algorithms assume conditional independence between features, $f(z_t)$:

$$p(f(z_t)|x_t, m) = \prod_i p(r_t^i, \phi_t^i, s_t^i|x_t, m) \tag{B.4}$$

The conditional independence allows for processing one feature at a time, simplifying algorithm implementation [Thrun et al., 2005].