Map-building and position estimation in mobile robots using self-organizing maps

George Palamas

A submission presented in partial fulfilment of the
requirements of the University of South Wales/Prifysgol De Cymru
for the degree of Doctor of Philosophy

January 2015

Table of Contents

Table of Figures

List of Acronyms

SOM: Self Organizing Map

NG: Neural Gas

GNG: Growing Neural Gas

SLAM: Simultaneous Localization and Map Building

ER: Evolutionary Robotics

AR: Adaptive Robotics

DR: Developmental Robotics

RSOM: Recurrent Self Organizing Map

CEV: Common Evidence Vector

PCA: Principal Component Analysis

IEV: Independent Evidence Vector

XOR: Exclusive OR

CBIR: Content Based Image Retrieval

GA: Genetic Algorithm

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Andrew Ware for the continuous support, for his patience and understanding. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to Dr. George Papadourakis, for offering me the opportunity to work on diverse exciting projects and for helping me to shape my interest and ideas. I would also like to thank Dr. Manolis Kavoussanos for his continuous support and encouragement from the very early stage of this research.

This thesis is dedicated to my beloved wife for her understanding and encouragement throughout this journey.

Abstract

The area in which this thesis is based on, is referred to as Adaptive Robotics. The main objective of this approach is to synthesize agents that evolve or develop their skills autonomously through interaction with a natural or artificial environment. The research aims at identifying the possibility of self-organizing systems to build an internal representation of the input space, able to handle the case of an unknown environment, and moreover, examine the causes, consequences and solutions to the conflicting problem of catastrophic forgetting that these systems are prone to.

The ability to navigate is arguably the most fundamental competence of any mobile agent, besides the ability to avoid basic environmental hazards. Recent studies of insect behavior and navigation reveal a number of elegant strategies that can be valuable when applied to the design of autonomous robots, without the need for higher level cognitive processes such as object identification and labelling. These bio-mimetic approaches have been reviewed, focusing on a self-organizing cognitive model of mental development, that allows for a common description of biological map building behavior. The motivation came from the assumption that self organizing systems could be used to reduce the amount of predefinition put in by a human operator and the ability to address noisy, inconsistent or no meaningful information with respect to the task being performed. In addition, a visual interpretation scheme, mimicking simple cells in the primary visual cortex, have been examined and critically analysed.

The research undertaken resulted in the development of a novel rehearsal map building scheme that is proven to build a representation of the environment, sequentially, from acquired visual snapshots of physical locations. These results also demonstrate the ability of the scheme to efficiently address the plasticity elasticity dilema presented by various connectionist models such as the self organizing map (SOM), neural gas (NG) and growing neural gas algorithm (GNG). This is encouraging enough to prompt further research that could result in an autonomous agent capable of self-localizing with a satisfactory degree and reliability in unknown and dynamically changing environments.

This thesis also explores the advantages of evolutionary sub-goal robot navigation with a cognitive map architecture. Experiments in simulation show that an evolved robot, adapted to

both exteroceptive and proprioceptive data, was able to successfully navigate through a list of sub-goals minimizing the problem of local minima in which evolutionary process can get trapped. The results demonstrate that a navigation behavior could be learned without the need for an in depth knowledge of the problems to be solved, especially in highly complex environments.

1. Introduction

Chapter 1 provides general information on the scope of research and an insight on simultaneous localization and map building for autonomous mobile robots. In addition, a formulation of the problems is described, as well as the key issues that need to be addressed. An overview of the aims and objectives is also provided, and reference to the software and tools used is presented. The chapter concluded with an outline of the systems proposed as well as an outline of the remainder of the thesis.

1.1 Autonomous Robots: The Case

Autonomous robots are robots which can perform desired tasks in unstructured environments without continuous human intervention. Many kinds of robots have some degree of autonomy and different robots can be autonomous in different ways. A high degree of autonomy is particularly desirable in fields such as space exploration, where communication delays and interruptions are unavoidable. Other more mundane uses benefit from having some level of autonomy, like cleaning floors, mowing lawns, and waste water treatment.

Today's "autonomous" robots in industrial assembly lines perform in directed environments within limited degrees of freedom. Unstructured environments in the workplace are challenging and can lead to unpredictable or even chaotic behavior. In more advanced factories robots can be capable of independent action in unpredictable terrains. The robot may need to navigate in dynamic areas with moving objects and humans. Thus, the robot needs to be able to determine a feasible path for navigation.

Navigation is one of the most fundamental competencies of any moving agent. Without the ability to localize, to identify goal directions and to plan paths towards the goal it is impossible to exploit the benefits of mobility fully. Furthermore, if mobile robots are ever to have an impact on the way humans live, or industrial production processes, they need the ability to navigate. More specifically, mobile robots intended for real world applications need to be able to:

- navigate autonomously, without human supervision,
- navigate in unstructured dynamically changing environments
- navigate without prior explicit models of the environment or other pre-supplied map knowledge.

The operation of simultaneous localization which serves to navigate an autonomous robot and map building mechanism which provides an environmental model is called SLAM (Simultaneous Localization and Map Building). Thrun [2008], provides an introduction to SLAM and a survey of paradigms which applies non parametric density estimation methods such as the particle filter method. Dissanayake et al., [2000] refers to the solution to the SLAM problem as the "Holy Grail" of the autonomous vehicle research community. While various sensors are used for this algorithm, vision-based approaches are relatively new and have attracted more attention in recent years.

1.2. Approaches to robot autonomy

The research field that is involved with the development of autonomous robots is very extensive and diverse. The area in which this thesis is based on, is referred to as Adaptive Robotics (AR). The main objective of this approach is to synthesize robots that evolve or develop their skills autonomously through interaction with a natural or artificial environment.

1.2.1 Developmental robotics

Developmental robotics (DR), also known as epigenetic robotics is a subfield of robotics in which ideas from artificial intelligence, developmental psychology, neuroscience and dynamical systems theory are used to develop complex cognitive architectures. For a brief review on methods and approaches on developmental robotics, see [Lungarella et al., 2003]. The aim of developmental robotics is to model the development of complex mental processes in natural and artificial systems. This approach focuses on the autonomous self-organization of general purpose, task nonspecific, control systems. Unlike evolutionary robotics which operates on populations of many individuals, developmental robotics learning comes from within the system and operates on single individuals (or small groups of individuals).

1.2.2 Evolutionary robotics

Evolutionary robotics (ER) is another emerging subfield of robotics research within the much larger field of autonomous robots. ER is related to developmental robotics, although different, and the primary goal is to develop automatic methods for developing autonomous robot controllers or even whole robots [Harvey et al., 2005]. The advantage is that they do not require in depth knowledge of the problems that have to be solved especially in highly complex environments that humans do not understand well. Yamada, [2005] proposed a behavior based evolutionary strategy for a robot to recognize environments. The study dictates that behaviors could be learned better than hand-coded ones.

ER frequently operate on populations of candidate controllers. Initially a population of randomly configured controllers is created and then repeatedly modified according to a fitness function in order to evolve a specific task. The most common approach is to use genetic algorithms (Gas) to evolve the population of candidate controllers in a repeating way that mimics natural evolution. Usually ER applies to control behaviors but can just as well apply to the evolution of the physical structure of the robot. Artificial neural networks are a common choice when designing a controller because of the applicability to relate sensor inputs to actuators outputs. In that context, Suzuki et al., [2006] explored a landmark-based navigation method by evolving a neural network controlling both vision and action of a mobile robot.

1.3 Research Aims and Objectives

The elaboration of this study involved two complementary schemes. The first scheme relates with the sensory coverage and interpretation of raw sensorial data in order to form discrete perceptual signatures for each robot position. The sensor that was chosen as the most appropriate is a standard panoramic camera, while visual descriptors such as color and texture used to describe the content of the images. In this manner, the localization of a robot on a pre-computed map could be reduced to an appearance based image recognition task [Akers et al., 2010]. The methods utilized to extract color visual descriptors are the widely used color histograms. For texture representation and discrimination, both Gabor filters [Gabor 1946] and wavelets analysis used [Chui, 1992], since these have been found to be particularly appropriate in modelling the visual cortex of mammalian brains [Lee, 1996]. Field [1987] compared various coding schemes for natural image representations, based on Gabor filter responses in a way that resembles the spatial-frequency tuning of mammalian simple cells.

The approaches adopted for robot mapping tasks, based on theories of self organizing systems and cognitive models of mental development. The motivation came from the assumption that self organizing systems could be used to reduce the amount of predefinition put in by a human operator and the ability to address noisy, inconsistent or no meaningful information with respect to the task being performed. The research undertaken resulted in: the development of a novel rehearsal map building scheme that is proven to build a map of the environment in which a robot operates. The results also demonstrate the ability of the scheme to efficiently address the plasticity elasticity dilema presented by various connectionist models such as the self organizing map (SOM), neural gas (NG) and growing neural gas algorithm (GNG). Based on this cognitive map model, a simulated agent proved to be capable of self-localizing with a satisfactory degree and reliability.

By examining these approaches, several research objectives were formed which are: analyse the potentiality of an appearance based visual methodology, to describe what the robot perceives as surrounding environment, in a way that is both discriminant and fault tolerant. The mechanism must be noise immune with the ability to generalize, while not depend on identification and recognition of distinct objects in the environment that need to be a priori known.

- Identify the possibility of self-organizing systems to build an internal representation of the input space able to handle the case of an unknown environment.
- Develop a system that simulates an autonomous moving agent as a basis for studying situated artificial intelligence for autonomous agents.
- Examine the causes, consequences and solutions to the conflicting problem of catastrophic forgetting in neural networks. Topological representations should be plastic enough to adapt to changing environments and learn new information, while maintaining important information preserved over time.
- Explore the advantages of robot navigation with a cognitive map architecture and

investigate the possibility of practical applications such as path planning and goal reaching.

1.4 Thesis Outline

The first section of this report critically reviews the relevant literature. It describes previous research into spatial and temporal reasoning for mobile robots and animal vision perception. It seeks to identify weaknesses in such research fields as well as highlighting areas that provide opportunities for further work. The major part of this section analyse the structures and the occurring complexities in the framework of this scheme.

The second section of this thesis articulates previously idealizing assumptions with the properties of artificial world environments and discusses the occurring problems. In order to circumvent these problems, the thesis introduces the concept of automatic topology preservation, which can be used to describe environments preserving similarities as much as possible. This section of this thesis also presents an application demonstrating the ability of a mobile robot to plan a route and navigate autonomously using an appearance based topological representation of space. Since it is difficult for humans to interpret this spatial representation, with respect to the environment that have been mapped, a genetic strategy incorporated along with a neural motion controller in order to autonomously perform the assigned task. The third section of this thesis brings into focus the problem of catastrophic forgetting in connectionist architectures, such as self organizing maps, and a proposed solution from literature that could prevent this undesirable effect is analysed and presented.

More analytically this thesis is organized as follows: Chapter 2 provides an analysis of the robot simulator that have been implemented to support this thesis. Chapter 3 provides an insight into the problem of robot map building as well as the key issues that need to be addressed in order to provide an effective solution. Chapter 4 provides a brief introduction to the technology of Artificial Neural Networks focusing on unsupervised learning and presents the the most relevant research studies in the field of autonomous robot navigation, that employ self organizing algorithms. Chapter 5 refers to the concept of spatial representation and a variety of inspiring biologically plausible models are introduced and analysed. Chapter 6 focuses on the problem of catastrophic forgetting in connectionist architectures, and propose solutions from the literature ranging from rehearsal to recurrent learning of sequences. Chapter 7 gives an overview of the techniques used for global feature detection and extraction and the problem of robot map building is standardized to a content based image retrieval problem. Chapter 8 demonstrates research results in the fields of visual feature selection, extraction and scene interpretation, alongside with a map building mechanism based on a self organizing map algorithm. Chapter 9, examines the way in which a robot might use the sub-symbolic representations of an environment that have emerged through self-organization. Experiments carried out using both proprioceptive and exteroceptive information, through an evolutionary strategy. Chapter 10 presents a comparative study that evaluates the effectiveness of applying the self-refreshing learning procedure to three well known unsupervised learning algorithms.

2. Experimental Setup

This chapter refers to experimental methodologies and approaches. It also provides an in depth analysis of the simulator that have been implemented to support the needs of this thesis. Finally, the terrain exploration strategies are being analysed and the challenges that met are being discussed.

2.1 Simulation Arguments

Developing controllers for developmental or evolutionary strategies requires a large number of evaluations or large populations of robots. In the case of evolutionary strategies, initial population controllers may behave harmfully by crashing to nearby obstacles, destroy the robotic platform or even cause injuries to humans. Usually, evolutionary strategies are used to reach an optimum solution for some complex optimization process and the major challenge is to transfer the evolved controller from the simulator to a physical robot [Zagal et al., 2004]. The reason is that evolution is free to explore all possibilities to obtain a high fitness value, including any inaccuracies of the simulation [Nelson et al., 2009]. The main advantages are summarized below.

- Rapid prototyping of algorithms.
- Simple basis for studying situated artificial intelligence for autonomous robots.
- Inexpensive, especially in multi-agent applications.
- Adjustable environment conditions (lighting, sensor noise).
- Faster and safer than a real robot.

The main drawback of using a simulator is that the generated results may perform poorly in real world conditions because of sensor readings, motor response inaccuracies and differences in interaction dynamics between robots and the environments [Brooks, 1992]. But robot controllers could be able to evolve to match the specificities of a simulation that differs from real world conditions [Mouret & Doncieux, 2012]. Improvements in software frameworks that simulate physical properties and dynamics like collisions, friction and forces, have led to simulation results that closely model real world situations. These software tools are usually faster than real time for most evolution scenarios. Physics based simulators are widely used by the research community for every type of robot from simple differential drive to articulated ones.

2.2 Why a 3D Robot Simulator

A 3D robot simulator must accurately simulate the dynamics of the robots and of the objects in the environment, thus allowing for a faultless evaluation of robot behavior. Another important feature is the support of flexible robotic platforms, different scenarios and terrains, as well as support for high-end sensors and visual realism. An omnidirectional vision camera is such a high-end sensor, increasingly popular in the robotics research area. These sensors combine conventional cameras and mirrors primarily to obtain large field of views. The available solutions regarding 3D robot simulators, even in high-end commercial solutions, did not provide support for such sensors, which led to the decision of building a simulator with the desired features to support the needs of this thesis.

2.3. Simulator Features

The core of the simulator was written in the C# programming language with Matlab based scripting for the simulation scenarios. It is able to simulate collision avoidance behavior, with infrared like sensors and bumpers, and navigation strategies, such as random walk, evolving on a flat surface. The main process cycle can be described in three steps (figure 2.1). First, the controller sends control signals to each wheel and the robot updates its position. Then, the sensors collects the measurements. Finally, the controller update its state based on current sensor signals.

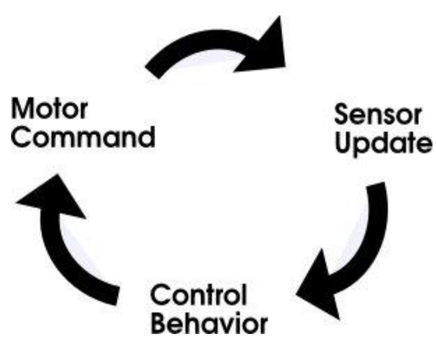


Figure 2.1. Behavioral circle of the simulator.

Both the robot and the simulated environment use the well known Bullet physics library, a freely available software package that models gravity, mass, friction, and collisions. In order to model the environment as well as the view from the omni-directional camera, the simulator make use of the XNA graphics framework. The choice of using Microsoft's DirectX technology, based on the photorealistic quality of produced results and advanced optical effects, such as simulations of light and shadows. Trivial image processing tasks like color histogram extraction, color space transofrmations and image conditioning are addressed by the well known open Computer Vision (openCV) library.

The simulator use the MATLAB engine as an automation server from C# via COM automation. Matlab is well suited to perform numerical calculations and data visualization, especially for complex machine vision tasks and neural network applications. Another advantage of the interface is that allows for simultaneously debugging C# application from both the C# side and the MATLAB side, using debuggers on each side.

2.3.1. The robot

All experiments were carried out using a simulated differential drive mobile robot model. This is a mobile robot whose movement is based on two separately driven wheels placed on either side of the robot body. It can thus change it's direction by varying the relative rate of rotation of its wheels and hence does not require an additional steering motion (figure 2.2). If both the wheels are driven in the same direction and speed, the robot will go in a straight line. Otherwise, depending on the speed of rotation and its direction, the center of rotation may fall anywhere in the line joining the two wheels. Since the direction of the robot is dependent on the rate and direction of rotation of the two driven wheels, these quantities should be sensed and controlled precisely. Differential wheeled robots are used extensively in robotics, since their motion is easy to program and can be well controlled. Virtually all consumer robots on the market today use differential steering, primarily for its low cost and mechanical simplicity.

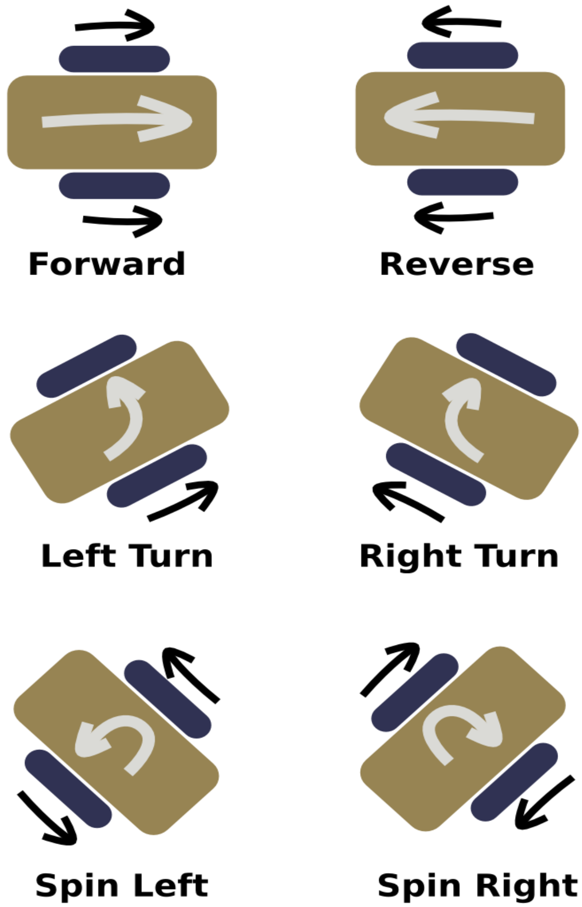


Figure 2.2. Differential drive is a model of controlling a robot with only two motorized wheels placed on either side of the robot body. The direction change by varying the relative rate of rotation of each wheel.

2.3.2. The Environment

The simulator make use of the XNA scene model to structure objects within the environment. The scene requires simulation of lighting, geometry, textures and model surfaces and objects are defined as 3D meshes which may be modified by a set of transformations. The robot can be placed at the specified position and orientation in the environment. All objects may have physical properties and can be either movable or static. Figure 2.3. illustrates how objects are coded in the simulator.

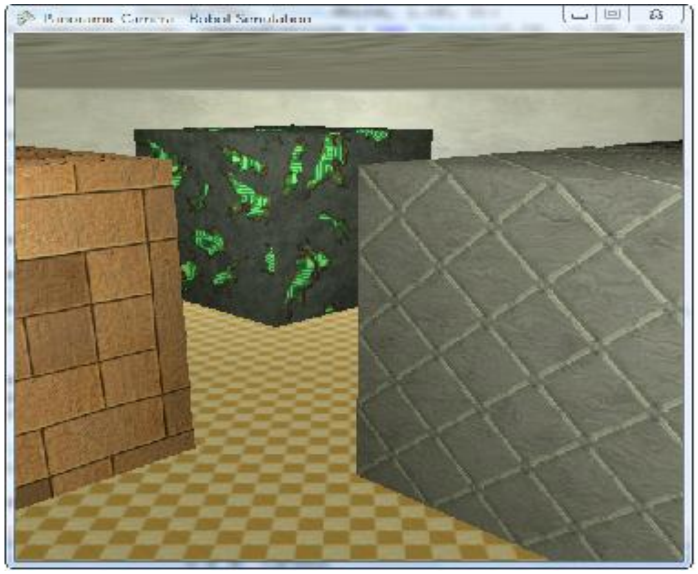


Figure 2.3. Close view of a simulated scene with various geometries and textures.

2.3.3. Omnidirectional Camera

Since traditional cameras suffer from the problem of having a limited visual field, an omnidirectional camera is used to obtain 360-degree field of view of the global scene. A common omnidirectional vision system in robotics is the catadioptric camera. The catadioptric camera consists of a vertically oriented color camera and a spherical mirror floating in front of the lens.

To model the reflective surface of the sphere, a method of environmental mapping known as Cube Mapping [Greene, 1986] has been applied. This is a technique for approximating the appearance of reflective surface by means of a precomputed texture image (figure 2.4). The image is generated, for every simulation step, by projecting the surroundings of the sphere onto the six faces of a cube. Then this cubical texture is wrapped onto the sphere to represent reflection lighting properties. An un-warped panorama of the simulated catadioptric camera can be seen in figure 2.5.

figure 2.4. Cube textures and sphere mapping. Visual snapshots obtained from the robot moving in a 3D artificial environment.

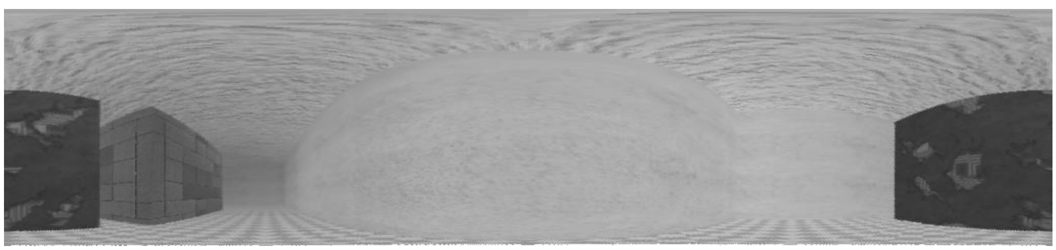


figure 2.5. Spherical to cylindrical coordinates. An un-warped image from the omni-directional sensor.

2.3.4. Dynamics

The simulator dynamics module supports the Bullet physics libary. Bullet is an open source physics engine that provides an approximate simulation of physical systems such as collision detection, rigid body and fluid dynamics. The main uses are in video games in which simulations are real time but is also well suited for high-performance scientific simulations. The primary limit of a physics engine is the precision of the numerical values representing the positions of forces and forces acting upon objects. Bullet physics library is published under the zlib license. The main features are:

- Rigid body and soft body simulation with discrete and continuous collision detection.
- Collision shapes include primitive shapes like sphere, box and cylinder.

- Convex hull shapes using Gilbert – Johnson – Keerthi distance algorithm (**GJK**).
- Non-convex and triangle mesh.
- Soft body objects like cloth, rope and deformable objects.
- A rich set of rigid body and soft body constraints with constraint limits and motors.

The dynamics module allows for simulating object interactions close to real world object interactions. Although, physics engines uses many approximations to optimize for speed and are relatively imprecise, it is difficult to substitute with kinematics when simulating mobile robots that collide or physically interact with its environment.

2.4. Control Behaviors

The robot considered to move in a plane by rotation and by forward translation. At each time step, the robot may issue either of two types of commands. First, the robot may rotate by a certain amount. Second, a translation command may be issued, instructing the robot to advance forward by a given distance. Rotation and translation commands are executed precisely but the actual distance traveled may be less than the commanded distance due to wheel slippage, friction and simulation inaccuracies. The robot uses only relative odometry to plan each successive step.

2.4.1. random walk

The general intuition is that robots should be like insects, equipped with simple control mechanisms tuned to their environments. Therefore, a model of movement using a simple two dimensional Brownian random walk has been implemented. Such an approach can imitate the navigation behavior of simple moving animals and microorganisms. The motion model is described as follows. Each successive step taken in a random direction is completely independent of previous steps taken and as the direction moved at each step is completely random (figure 2.6). Another potential advantage of this approach is minimization of simulation artifacts such as cyclic behavior.

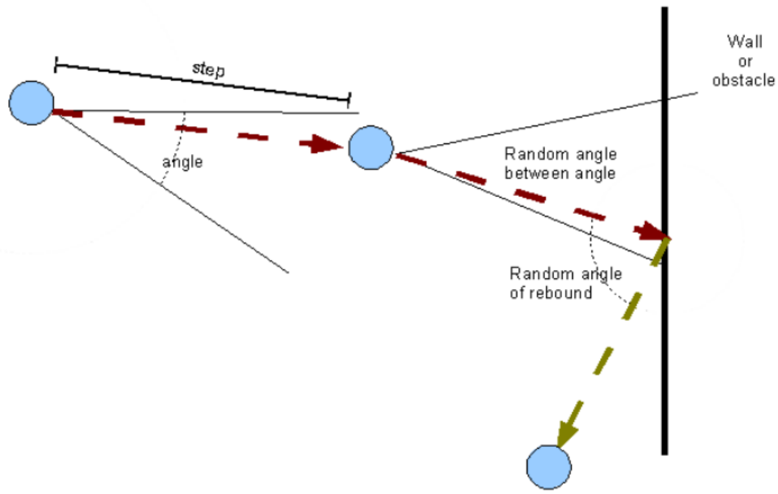


Figure 2.6. Motion model of two dimensional Brownian random walk.

2.4.2 Obstacle Avoidance and Navigation

Simple soft bumpers and proximity sensors are used as the main navigation sensors for collision detection and reaction. As soon as an object enters the impact area of a bumper, the robot reacts by moving backwards and then rotating, left or right, to some random degree between 60^0 and 120^0. Proximity sensors are used mainly for a specific behavior of a wall following control loop. When following a wall, the robot tries to keep a constant distance from the wall. If an object appears in front of the robot, it should veer away of the wall and around the obstacle.

2.5 Summary

Recently, the use of a robotics simulators for autonomous robotics is highly recommended regardless of whether an actual robot is available or not. A simulator is a safe and cost effective solution and a final version of the controller may be transferred on an actual robot but the success of off-line testing and evaluation depends on how similar the real environment of the robot is to the simulated environment. Although a simulator is only an abstract model of real world situations, it can be very useful in exploring aspects of the problem of autonomous agents, especially in the fields of developmental and evolutionary robotics.

In this chapter, the simulator that was used to conduct the experiments of this research, was presented and the technical and functional characteristics were analysed. Behavior based robot simulators allows for actions that are more biological in nature when compared to simulators that are more computational, because robots can learn from mistakes and are capable of demonstrating complex behaviors such as evolving the capacity of moving in the environment. The simulator integrates the Bullet dynamics library, which provide excellent real time simulation, and embeds them in a 3D graphics engine that serves for the optical results.

The simulator also hosts a set of algorithms representative for different approaches such as neural networks and genetic algorithms. All these algorithms have been built in the Matlab environment. The robot exhibits basic hardwired behavior for navigation purposes such as obstacle1.0 avoidance and random walk navigation and is able to respond and adapt in real time. Accuracy can be in varying degrees for an optimum blend of speed and accuracy, being as computationaly intensive as it needed. A matlab interface has also been implemented for rapid prototyping, evaluation of behaviors and for data visualization purposes.

3. Robot Localization

Chapter 3 Begins with a definition of what is an autonomous mobile robot and what are the prerequisites to incorporate such capabilities. The chapter also provides general information on the problems of determining the position of a mobile robot and map building as well as the key issues that need to be tackled in both problems. Moreover, an overview of sensor technologies is provided with a reference to the types of data obtained from various sensors. Finally, the most widespread methods of self-localization are highlighted and briefly analysed.

3.1. Autonomous Robots

Service robots capable of every type of autonomous movement for general-purpose use in environments such as sickrooms and offices encounter unique challenges, and demand the ability to operate without failure. Therefore it is necessary that they possess special skills.

- Realistic robotic applications of the future demand a high degree of system autonomy in unstructured environments. These environments and their sensor signatures are hardly ever known a priori at the time the robot is designed. Therefore, it is a requisite for these robots to incorporate learning capabilities so as to perform in previously unexplored environments.

- The fundamental requirement of this skill is position estimation (self-localization) as this is a precursor to addressing issues relating to a robot's mobility such as planning efficient routes, avoiding cyclic behavior, and predominantly preventing the robot losing track of its position relative to the rest of the environment.

- For a robot to associate behaviors with a place requires it to make positional estimation and to be able to plan and follow routes. Early robotic platforms in the 1970s routinely used wire or rail guidance before progressing into more flexible, geometrical triangulation algorithms based on active beacons or self supporting, tracking of salient natural or artificial features.

3.2. Sensing the Environment

To be fully autonomous, a robot must rely on its own perceptions to localize. Perception of the world generates representation concepts, topological or geometrical, within a mental framework relating new concepts to preexisting ones. The space of possible perceptions available to the robot for carrying out this task may be divided into two categories:

- Internal perception (proprioception) or perceptions of its own interactions with the world, associate changes of primitive actuator behavior like motor states.
- External or sensory perception (exteroception) is sensing things of the outside world. A robot's exteroceptors include all kind of sensors like proximity detectors or video cameras.

Sensors such as whiskers cannot sense their environment without physical contact with the world. Proximity sensors such as ultrasonic sonar and infrared range finders have a short range and imply interference and wraparound, thus precise sensing of the environment requires high degree of directionality. Proximity sensors may help avoid running into an obstacle and can be used to handle geometrical concepts such as distances to help build geometrical and topological maps.

Vision sensors are the richest source of information. However, they provide challenges in terms for example of extensive storage requirements and the time to process the vast amount of data available. Information extracted from images can be low level primitives such as color and texture information, together with higher level information such as the shapes of objects, optical flow, etc.

A modern approach in machine vision is to deal with images as combinations of different colored and textured regions. The content of these regions should be as similar as possible and the content of different regions should be as dissimilar as possible.

A special issue of vision sensor is the panoramic or omni-directional vision system with a single camera [Baker & Nayar, 1998]. Images are obtained by placing a convex mirror a short distance from a camera as shown in figure 3.1. These systems provide a 360^o view of the robot's environment around the vertical axis in addition to ~30 above horizontal axis. They have become increasingly popular and are now used in many applications including autonomous navigation and teleconference. Argyros et al., [2001] have proposed a solution for a mobile robot homing behavior that is based on the extraction of low level visual ques extracted from an omnidirectional camera. Their main advantages include:

- Largest field of view compared to orthographic or standard cameras. Landmarks are always in the field of view except for occasional occlusion. This is advantageous when utilizing topological representations as the more information the image contains the more stable it is.
- Orientation independency when employed with statistical methods such as histograms and distribution functions
- No rotation mechanism is required, thus they consume less power and therefore have increased reliability.

The main disadvantage is that acquired images are of lower resolution therefore not well suited for applications like local fine-grained texture perception.

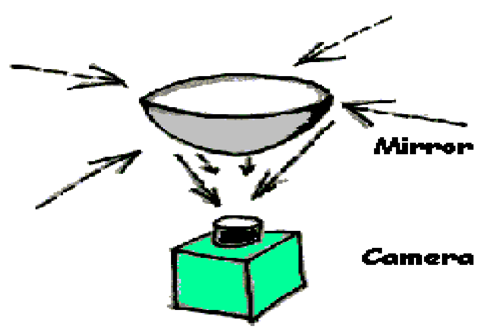


Figure 3.1. Omnidirectional vision sensor.

3.3. Localization

Localization refers to the task of identifying places in the environment after prior exploration and map-building by the robot. Localization is one of the fundamental problems to be solved when designing a navigation system. If a robot does not know where it is, it cannot effectively plan movements or reach target positions. Map-based localisation depicts the following advantages.

1. Robot's position may be available to human operators and have also been used as a medium for human-machine interaction [Topp, 2008].

2. Even if robot position is difficult for humans to interpret, with respect to the environment that have been mapped, the robot autonomously can perform a variety of assigned tasks such as navigation.

3.4. Types of localization

Broadly speaking, there are two major options to achieve robot navigation. One is to use proprioceptive sensors such as wheel encoders, and to perform navigation through path integration (often referred to as dead reckoning). The other option is to use exteroceptive sensors, and to navigate using landmarks or general features of the scene. Position estimation in an outdoor environment can be realized with GPS systems which are becoming standard equipment. Furthermore, differential GPS (DGPS) has an accuracy of a few centimetres if enough satellites are visible hence fulfilling the potential for robot navigation. Unfortunately, the signal from the GPS satellites is too weak to penetrate most buildings, making GPS useless for indoor localization.

3.4.1. Continuous Localization

Continuous localization, known also as position-tracking or relative positioning, is a technique that allows a robot to maintain an accurate estimate of current location by performing regular, small corrections to the odometry. However, as dead reckoning is based on pro-prioception, i.e. completely independent from outside information, errors in the estimate are accumulated over time (wheel slippage, uneven floors, etc). Methods to correct tracking of the position, whithin the framework of dead reckoning, are impossible and rely on mechanisms that can update the correct location of the robot, by referring to external information, for example landmarks. The prototype of algorithms proposed to solve position tracking is the Kalman Filter. [Leonard & Durrant, 1991] . In the context of vision based robotics, Lowe presented a method to extract invariant image features in order to perform matching between different views in a scene [Lowe, 2004].

3.4.2. Absolute Localisation

Absolute localization is based on signals from external sensors to determine the global position and orientation of the robot. The robot has become lost through some arbitrary circumstances and is unable to localize using past experience, i.e., no initial or approximate estimate of the position is available. To tackle the problem, an explicit model of a given environment is needed to estimate the location using sensor data. Common frameworks are the Multi Hypotheses Localization [Jensfelt & Kristensen, 2001], Histogram Filters [Burgard et al., 1996] and Particle Filters [Dieter et al., 1999].

3.4.3. Simultaneous Localization and Mapping

Simultaneous localization and mapping (SLAM) generally refers to processes used by autonomous agents to build up a geometrical map within an unknown environment while at the same time keeping track of their current position. SLAM is not straightforward due to inherent uncertainties in discerning the robot's relative movement from its various sensors. If at the next iteration of map building, sensor measurements are erroneous or even slightly inaccurate, then any additional features will distort the map. Without a frequent update of the robot's correct location, these positional errors build cumulatively, and the robot loose the ability to know its precise location. There are various techniques to compensate for this such as landmark extraction and map update from previously detected landmarks so as to re-skew recent parts of the map. Some of the statistical techniques used in SLAM include Kalman filters [Casarrubias et al., 2010], particle filters (aka. Monte Carlo methods) and scan matching of range data [Vargas et al.,2010], [Nieto et al., 2007]. For an in-depth survey methods for map building, see [Thrun, 2002].

SLAM has not yet been fully perfected, but it is starting to be employed in unmanned aerial vehicles, autonomous underwater vehicles, planetary rovers and newly emerging domestic robots. It is generally considered that "solving" the SLAM problem has been one of the notable achievement of robotics research in the past decades. Pioneering work in the field of SLAM was conducted in the mid 1980s and early 1990s by two independent teams [Smith & Cheeseman, 1987] , [Leonard & Dyrrant-Whyte, 1991].

3.5. The Kidnapped Robot Problem

A robot at an unknown position must surmise an occupied location based on recent sensory information against prior knowledge of the environment. When a well localized robot is transferred to some random location without being told it needs to embody some kind of spatial reasoning.

The kidnapped robot problem [Engelson & McDermott, 1992], is a special issue which differs from the global localization problem because a kidnapped robot is unable to estimate it's own position via a localization process. If the agent is not aware of the beginning location it would not know how to go somewhere else. Guessing that you are positioned in your bed when you wake up in the morning may be right unless you have been transferred to another location in the middle of the night. Awareness of this initial position is very important if your next navigation plan is to visit your work space location. The kidnapped robot problem is often used to test a robot's ability to recover autonomously from localization failures.

3.6. Traditional Position Estimation Methods

3.6.1. Beacon based localization

Position is calculated through simple triangulation. A popular implementation is the Global Position System (GPS) promising to become the standard navigation solution for almost all Automated Vehicle Systems (AVS). Similar systems using infrared beamers or radio beacons demand modifications to surrounding environment. Usually these systems cannot be used indoors.

3.6.2. Kalman Filters

The Kalman filter is a recursive solution to the discrete-data linear filtering problem [Welch & Bishop, 1995]. A lot of research has been done on this algorithm and it has been used extensively throughout the topic of robot localization. This method is mainly used for position tracking, i.e. the initial position is known and subsequently the movement is tracked. The Kalman filter gives an estimate of the state of a dynamic system from noisy measurements.

The Kalman filter is a recursive estimator, this means that only the estimated state from the previous and current steps are required to predict the new state. The algorithm works in a two step-process and uses a series of measurements observed over time. For every time and measurement pair, previous a posteriori estimates are used to calculate the new a priori estimates and from them the new a posteriori estimates.

Kalman filter can be used to compensate for the errors that occurs when odometry is employed during robot movement. As a condition for this method, both initial position and orientation, have to be known in advance. Kalman filters are known to be efficient and accurate in determining position and heading of the robot but are generally not well suited to solve the kidnapped-robot problem. Kurz, [1996] proposed a free space partitioning method to generate environmental maps through a learning classifier and an extended Kalman filter algorithm to compensate with the dead-reckoning drift.

3.6.3. Probabilistic Localization

3.6.3.1. Markov Localization

Markov localization utilizes a probabilistic algorithm [Fox et al., 1999]. This means that rather than maintaining a single assumption on which is the best estimate of the pose of a robot, the technique maintains a probability density over the space of every possible place and direction. These densities occurs in several formats and each format represents some kind of information about the positional status of the robot. To look at a case, because of lack of any information concerning the position of the robot, the density will appear as a uniform distribution. When, on the other hand, there is great confidence as to the position this will appear as a distribution with a peak located around the supposedly true position. This procedure is also capable of tracking multiple hypothesis, i.e. the cases for which there is not a clear view of the position occupied by the robot. Positions for which there is a high probability that the robot is located have a higher density and positions where the robot is not likely to be found, have comparatively low probability densities (figure 3.2).

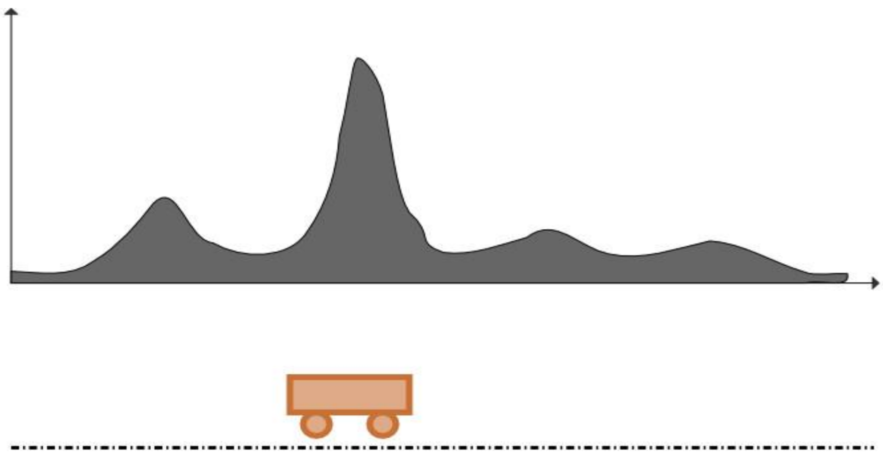


figure 3.2. Markov localization uses an explicit, discrete representation for the probability of all positions in the state space. Upper side shows belief state where higher peaks means higher probability. Down side shows actual robot position in a hypothetical 1D space.

Unlike Kalman Filters, Markov Localization has the ability to recover from situations in which the robot, after a certain amount of time, has absolutely no idea about its current position, but have trouble localizing when placed in a dynamic environment where people might be moving around or pieces of furniture are continually repositioned. Some approaches (grid-based) tend to use enormous amounts of memory and the resolution and size of the state space have to be determined before starting with the computational part. During each prediction and measurement steps, all the cells are updated. If the number of cells in the map is too large, the computation can become too heavy for real-time operations.

As an example, consider a robot moving in a physical space with dimensions of 50x50 meters. Suppose also that the required accuracy is a cell with dimensions of 0.1x0.1 meters and orientation step of 1 degree. Calculating, in this case, all possible positions and orientations of the robot leads to a result which includes 50x50x100x360 = 90,000,000 cells which need to be updated at each step consideration. Of the above leads to the conclusion that fine fixed decomposition grids result in very big state space.

3.6.3.2. Monte Carlo Localization

The idea of identifying the position of an agent with Monte Carlo method, also known as Particle Filter, is different compared to approaches such as Kalman filters and Markov localization. In this case, the probability of the robot being in a particular position is represented by a set of samples that are randomly drawn from it [Fox et al., 1999]. The method is based on a recursive Bayes filter that estimates the posterior distribution of a single or multiple positions, which are conditioned on the sensor measurements.

The main advantage of this method, compared to the above, lies in the fact that is computationally less expensive, especially when compared to grid-based Markov localization. The additional advantage is that it has the ability to track the position of a robot locally and localize it globally. On the other hand, particle filters are not well suited to perform with high dimensional data obtained from sensors such as cameras. Beyond this, cases may arise that demand a large number of particles to be sampled for convergence. If the number of particles increases significantly the results tend to be less accurate.

3.6.4. Vision Based Localization

Real world applications demand more detailed sensor information to provide the robot with better world understanding. Visual sensors are potentially the most powerful source of information and appear to be the best candidates for autonomous robot applications. The two main methodologies widely used today are divided into those that are based in distinctive objects [Se et al., 2005] and those that catch the general appearance [Ascani et al., 2008], or the nuance, of the scene view. In case of Landmarks placed at various points in the environment, robots can better estimate their position in the environment through their representative visual information. However, landmark-based methodology relies heavily on identification and recognition of distinct objects in the environment that need to be a priori known. Also, due to noisy measurements and dynamically changing environments, the process of object identification might become quite challenging. All the above issues, may be efficiently tackled in the context of an appearance-based navigation strategy. However, some disadvantages of appearance based systems including lack of depth information and image occlusions renders them unsuitable for specific applications.

3.6.5. Landmark based Localization

Often used complementary to odometry for recalibration and updating purposes. Landmark based position estimation mechanisms are challenging to apply in natural and unstructured environments. The difficulty essentially lies in the conception of the technique that extracts and model the landmarks from raw sensor data. When applied in a non structured environment, it is not easy to find distinguishable objects especially from different viewpoints. Moreover, landmark based techniques may require object recognition abilities.

Using these salient landmarks even if they can easily be recognized from one position, may be difficult to recognize for successive observation steps. Since pattern recognition techniques are prone to different scales, orientations and partial occlusions, a robot may fail to recognize a salient landmark from some perspective views. Usually these distinct locations are used only to update current estimation of location and tracking. [Loevsky & Shimshoni, 2010] proposed a localization method based on a triangulation system supported by artificial landmarks. [Bais & Sablatnig, 2006] proposed a system that based on range measurements of a single landmark from two arbitrary points. Many other approaches to visual servoing and mobile robot navigation have been based on tracking feature points or landmarks from panoramic imagery [Argyros et al., 2001], [Fiala & Basu, 2004].

However, sometimes the recognition of landmarks is not unambiguous ("perceptual aliasing"). This means that two different places can be perceived the same (figure 3.3). For example, in a building, a person may not be sure about knowing where he is, only with the visual information, because all the corridors look the same. Navigation can become unreliable, and additional methods have to be sought to establish the navigator's position unambiguously.

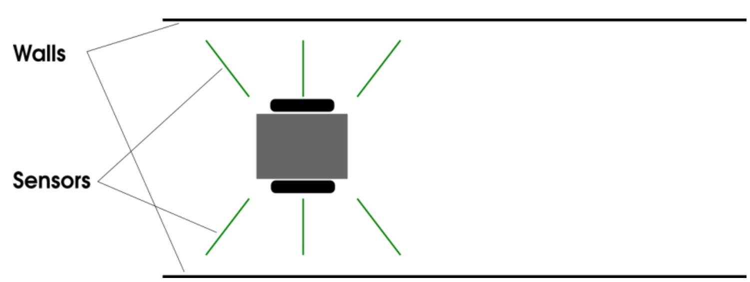


Figure 3.3. Perceptual aliasing problem.

3.6.6. Selecting image landmarks

Not all features points are equally effective as landmarks. Generally speaking, promising landmarks must be salient and distinguishable. Salient means highlighted objects that can be observed for a great range of locations, e.g. a red fire distinguisher in a white empty room, and distinctive means difficult to confuse during recognition with objects that look similar or have similar visual qualitative features.

Generally landmarks can be artificial or natural. Natural landmarks are more desirable because it is not necessary to modify the environment. But these landmarks are difficult to isolate from the background and recognition depends on lighting conditions viewing directions and occlusions especially in dynamically changing surroundings. On the contrary, artificial landmarks are easier to detect, specially designed to be independent to different lighting conditions, rotations or translations, but at the cost of modifications in the environment like drawing unique patterns in the walls. Various localization systems have been based on developing methods for selecting visually distinctive landmarks [Knapek et al., 2000], [Miller et al., 2011].

3.7. Appearance based localization

Landmark-based localization methods rely on the assumption that landmarks can be detected and accurately interpreted from raw sensors readings. However interpretation from sensor readings to accurate geometric representation is complex and error prone. From another viewpoint, an appearance-based method of environment is not encoded as a set of geometrical visual features, but as an appearance map that includes a collection of sensor readings obtained at known positions. Krose et al., [2002] applied a Principal Component Analysis to panoramic images to build an internal representation of the environment. A feature matching technique proposed by [Ascani et al., 2008] to address the issue of topological localization by matching the current view with reference images. Two different approaches were compared based on SIFT and SURF image features. An appearance-based place recognition method, that use a panoramic camera, presented by Ulrich & Nourbakhsh, [2000] that use nearest neighbor learning classification and image histogram matching. The proposed system proved robust on classifying correctly the input color images.

The advantage of this representation is that the raw sensor readings generate a qualitative estimate of position. The currently perceived image can be directly matched with past experiences stored in the appearance-based topological representation. Sensor readings, that used in this way, does not rely on precise metric measurements as opposed to traditional geometrical based maps.

In the field of computer vision the use of appearance based techniques have become widespread in recent years [Akers et al., 2010]. A comparison between the two families of vision based localization methods can be found in [Sim et al., 2003], showing that appearance-based methods are more robust to noise, occlusions and changes in illumination than landmark based-methods. The source of inspiration is that small animals, such as insects, navigate through natural environments seemingly with little effort [Collet, 2010]. Despite their relatively simple nervous system and hence limited memory capacity, bees and desert ants are able to find their way back. Such a level of efficiency indicates flexible representations of the surroundings based on visual cues taken from target locations like home and food sources. [Cartwright et al., 1982],[Dill et al., 1993],[Collett et al., 1998]. These representations seems to have an appearance based flavor rather than a Cartesian arrangement of landmarks. To visit target locations after prior exploration, insects traverse in a way that reduce discrepancies between the stored snapshot and their current retinal image.

As stated before the main drawback of appearance-based methods is that localization is only possible in previously mapped areas. Recently, several applications have shown promising results. [Booij et al., 2007] proposed an appearance based topological map method extracting semantic information about scenes. Another approach, that also accounts for the perceptual aliasing problem, proposed the use of appearance only information to localize a robot in a known map [Cummins et al., 2007]. Like landmark based mechanisms, appearance based navigation systems suffer from the problem of perceptual aliasing, the fact that different locations gives identical sensory perceptions. A possible solution whould be the incorporation of temporal information or odometry to resolve any conflicts.

3.8. Summary

Localization and map building is a fundamental task in order to achieve high levels of autonomy in robot navigation thus different approaches have been proposed that exhibit satisfactory behavior, most of them in a probabilistic framework. Visual sensors are potentially the most powerful source of information and appears to be the best candidates for autonomous robot applications. Vision based localization methods can be divided in two families. First, landmark based methods that rely on the assumption that distinct element position can be accurately extracted from sensor data. Second, appearance based methods that model the environment from a set of sensor readings obtained at known positions and can be directly compared with the previous observations stored in the appearance map. This simple way to use sensor readings makes the appearance-based approach really appealing, although, an appearance map is always required to store previous sensory perceptions of the environment. Taking inspiration from nature, insects seem to use memorized visual representations to find their way back to places of interest, like food sources and nests. The fact that small insects with simplistic brains exhibit good localization capabilities, with apparent ease, makes it a good candidate for developing and testing bio-inspired methods for robot navigation and mapping. Moreover, using visual snapshots without requiring intensive processing, such as identifying special landmarks, immediately relates to an appearance based context.

4. Neural Networks

Chapter 4 presents a brief history and an introduction to the technology of Artificial Neural Networks focusing on the well known variations of unsupervised learning. Additionally, the most relevant research studies in the field of autonomous robot navigation, that employ self organizing algorithms, is presented, in order to establish the current problems and decide the methodology of addressing them.

4.1 Introduction

The first neural network was proposed in 1956 by Frank Rosenblatt. It was called a Perceptron. Thirteen years later a publication known as Perceptrons, by Minsky et al., [1969], resulted in a significant fillip to neural network research. Their publication laid the foundations for the general neural network architecture but also pointed out some serious limitations. The main issue was that Perceptron could not perform a basic logical computation of a XOR (exclusive-or). The second significant issue was that computers were not sophisticated enough to effectively handle the long run time required by large neural networks. These findings almost devalued research in neural networks but later advances like the back-propagation algorithm [Rumelhart et al., 1986] along with greater processing power brought back the research interest and within a short time became widespread.

Designed around the brain-paradigm of Artificial Intelligence, neural networks attempt to model the biological brain. Neural networks have been used to model complex relationships between inputs and outputs, and to find patterns in data. Neural networks mimick the biological neural networks in that functions are performed decentralized and in parallel by all units. Neural network theory has served both to better identify how the neurons in the brain function and to provide the basis for efforts to create artificial intelligence. Because an ANN can capture many kinds of relationships it allows the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or impossible to explain.

Neural networks are also fault tolerant in that a set of inappropriate data or a number of destroyed nodes will not render the network useless. What has attracted the most interest in neural networks is the possibility of learning. Given a specific problem to solve, learning means using a set of observations to find a solution to the problem by optimizing the general state. In general, there are two types of learning: supervised and unsupervised.

4.1.1. Supervised Learning

The correct answers are known and used in a feedback manner to train the network for the given problem. This type of learning utilizes both input vectors and output vectors. The input data are fed to the input and the output data are being associated with the inputs. In a special type of supervised learning, reinforcement learning, the network is only told if it's output is right or wrong. Back-propagation algorithms make use of this style.

4.1.2. Unsupervised Learning

Unsupervised learning is a type of training that tries to reveal data structures from data that seems to be heterogeneous. Since the data given to the neural network are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning. The greatest advantage is that no human interaction is needed for unsupervised learning. This can be an extremely important feature, especially when dealing with a large and/or complex data set that would be time-consuming or difficult to a human to compute.

4.2. Clustering

Clustering is the process of organizing a collection of n-dimensional vectors into groups whose members share similar features in some way. Each of these groups is represented by a n-dimensional vector called a codebook vector. The aim of clustering is to categorize large data collections by classifying in smaller sets of similar content. The most well known soft clustering algorithm is the K-means algorithm by MacQueen, [1967]. This method of cluster analysis aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (figure 4.1)

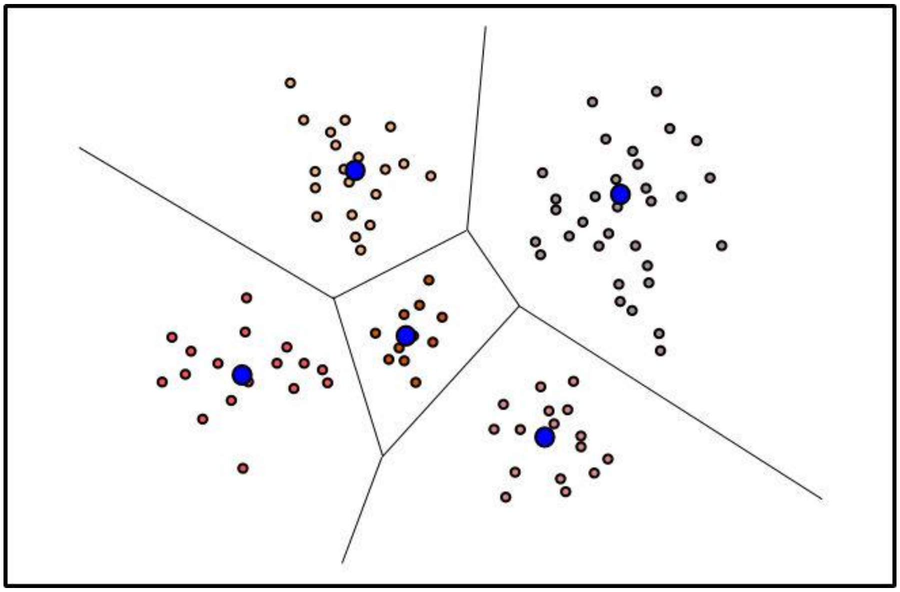


figure 4.1. Clustering results of the K-means algorithm. Blue dots correspond to cluster centroids and lines separates the Voronoi cells.

This results in a partitioning of the data space into Voronoi cells. Since it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. As the algorithm is usually very fast, it is common to run it multiple times with different starting conditions. However, in the worst case, *k*-means can be very slow to converge: in particular it has been shown that there exist certain point sets, even in 2 dimensions, on which *k*-means takes exponential time, that is $2^{\Omega(n)}$, to converge [Vattani, 2008]. Other algorithms widely used for vector quantization is the Kohonen's Self Organizing Map (SOM), [Kohonen, 1982] and the Growing Neural Gas algorithm described by [Fritzke, 1995]. For a brief overview of data clustering, and well known clustering methods, see [Jain, 2010].

In some cases little or no information is available about the input distribution or the size of the input data set, in these cases it is hard to determine a priori the number of nodes to use, such is the case in Kohonen's SOM and in the NG algorithm and also in classical K-means clustering. The GNG is an incremental algorithm that only has parameters constant in time, thus, there is no need to determine the number of nodes a priori since nodes are added incrementally. Insertion of new nodes stop when a user defined performance criteria is met or if a maximum network size has been reached.

4.3. Topology Preserving Networks

In unsupervised learning the only fact available is the input set. What such a network can serve for is among the others topology preservation and vector quantization. Topology preserving means that close input signals are mapped to neurons which are close in the lattice structure and conversely, close neurons in the lattice structure come from close input signals in the input space preserving similarities between data as much as possible (figure 4.2). The key to a topological relationship is based on an abstraction of knowledge in terms of connectivity by mapping an input set of information into a data structure. A graph is such a kind of abstract data structure that consists of points or nodes connected by links called lines or edges retaining similarity relations between the original data and the data after mapping.

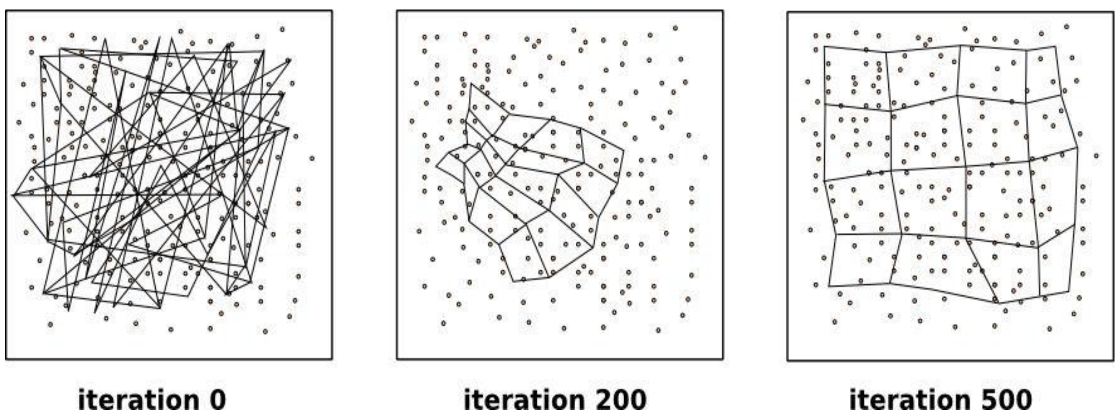


Figure 4.2. Topology preservation through vector quantization. A 5x5 self-organizing map network expanding to capture a 2D input manifold. Initially all nodes are located in random positions.

Unsupervised learning architectures are often considered topographic or topology preserving networks as a consequence of the competitive learning method. The term definition of topology preservation and a mathematical relationship with geometrical structures such as Voronoi Diagrams and Delaunay Triangulation have been the case study of Martinetz et al., [1991].

Vector quantization is based on competitive learning so it is closely related to self organizing models. It works by dividing a large set of points into groups having approximately the same number of points closest to them. Each group is represented by a neuron and consists of all the points that belong to the area of influence of this particular neuron by means of some distance metric. This abstraction of knowledge can be seen as a semantic model which does not involve memory of a specific event.

4.4 Kohonen Feature Maps

Sensory pathways in the brain are organized in such a way that its arrangement reflects some physical characteristic of the external stimulus being sensed. Kohonen self organising feature map, tries to mimic two-dimensional arrangements of neurons in the brain [Kohonen, 1995]. SOM, maybe the most popular unsupervised artificial neural network, is an algorithm that maps similar input vectors, which are close to each other onto contiguous locations in the output space. The dimensions of the node lattice and node number are chosen in advance. Kohonen feature maps are popular also because of their dimensionality reduction capabilities, meaning that they project multidimensional input space into normally one or two dimensional space. Ishii et al., [2004] proposed a navigation method, based on a SOM in order to reduce the dimensions of parameters, for an autonomous underwater vehicle. It is also possible for interconnected structures of more dimensions although difficult to implement. For every input presented to the net, the distance between input and every node in the map is calculated to find the winner node. The node with minimum distance is the winner node or best matching unit. The winner node and neighbours update their corresponding weights through some function, typically a Gaussian.

A self-organizing map consists of a single-layer feed-forward network where the outputs are arranged in a fixed grid of neurons (figure 4.3). Each input is connected to all output neurons. Every neuron holds a weight vector of the same dimension as the input vectors. The number of input dimensions is usually much higher than the output grid dimension. Winner node, or best matching unit is the node closest to input vector with respect to a metric as is euclidean distance metric.

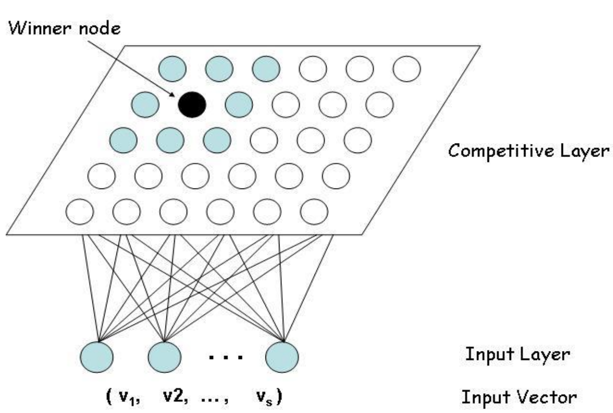


Figure 4.3. Kohonen Self-Organising Map

4.4.1. Practical application

The Self-Organizing Map algorithm can be broken up into 6 steps.

1. For each node, weight vector is initialized.

2. An input vector *Vi* is randomly chosen from the set of training data and fed to the network.

3. All nodes and their associated vectors *Wi* are examined to find the closest one to the input vector. The winning node is commonly known as the Best Matching Unit (BMU). The metric used is the euclidean distance (equation 4.1).

$$\text{distance}=\sqrt{\left(\sum_{\mathrm{i=1}}^{\mathrm{i=n}}(V_i - W_i)^2\right)} \quad (4.1)$$

4. The radius of the neighbourhood of the BMU is calculated. The radius starts from an initial value and diminishes on each time step. An Initial value equal to the radius of the network is a frequent choice. A common exponential decay value is the following (equation 4.2).

$$\sigma(\tau)=\sigma_0\exp\left(\frac{-t}{\lambda}\right) \quad (4.2)$$

5. Any nodes in the range of the BMU are adjusted to make it more closely to the input vector (equation 4.3a, 4.3b). The amount that should be changed to look like the input signal is determined by the distance from the BMU. Weight vector adjusted according to the following equation. Where *L* is the decay of the learning rate λ.

$$W_{(\mathrm{t+1})}=\mathrm{W}_{(t)}+\mathrm{L}_{(t)}\left(V_{(t)} - W_{(t)}\right) \text{ (4.3a)}$$

$$L_{(t)}=\mathrm{L}_0\exp\left(\frac{-t}{\lambda}\right)\mathrm{t}= \text{ (4.3b)}$$

6. Repeat from step 2 for N iterations.

4.5. Neural Gas

The neural gas (NG) is a biologically inspired adaptive algorithm coined by Martinez et al., [1991] partially inspired by the physical properties of uniform gases and partially the work of the self organizing neural networks. A gas does not have a fixed shape or size but in the absence of gravity will expand to fill a container. Once a container shape of points in a given distribution has been defined, freely moving "particles" expand in the shape in a uniform way, creating a grating of particles. NG is topology representing neural network, that is after reaching convergence, the network nodes would be representing the distribution being modelled. NG is applied where there is a need for data compression or vector quantization and often referred to as a robust alternative to k-means algorithm. Given a distribution $P(x)$that consists of vectors $x$the algorithm involves the following steps.

1. All weight vectors w_i,i=1,...,N are initialized.

2. For each time step ta data vector x is randomly chosen and the distances between the data vector and every weight vector are calculated and sorted. $i_0$denotes the index of the closest weight vector, i_1 the index of the second closest weight vector etc. and i_{N-1} the most distant weight vector from input signal x.

3. Each weight vector (k=0,...,N − 1) is adapted according to the formula 4.4.

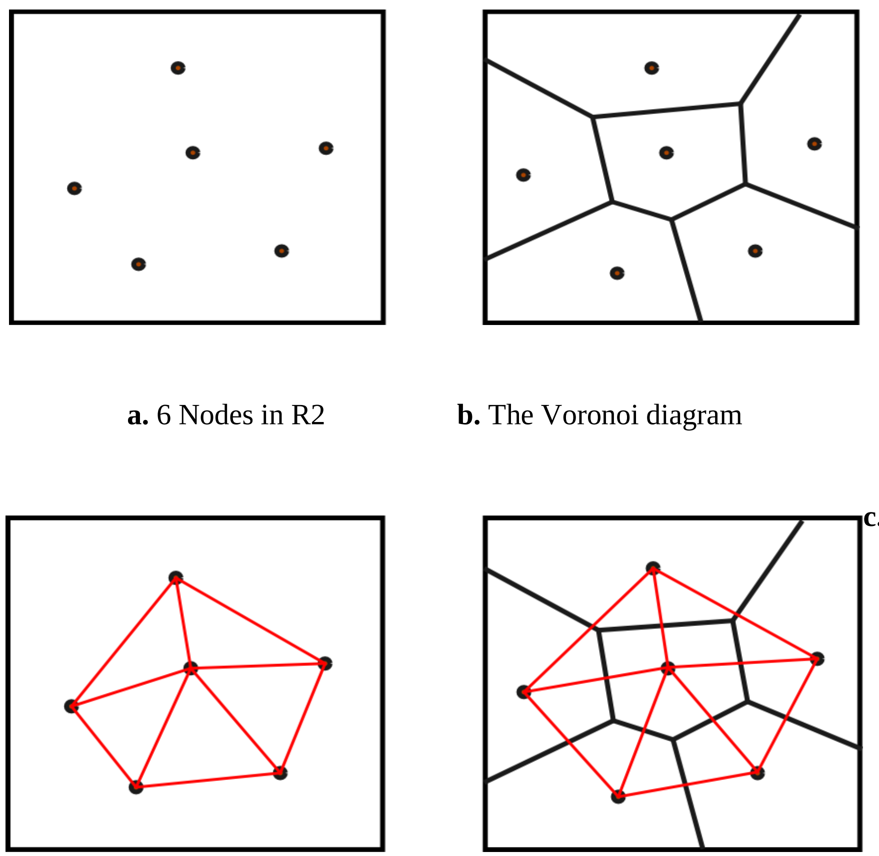
$$w_{\mathrm{ik}}^{t+1} = w_{\mathrm{ik}} + \varepsilon \cdot e^{\frac{-k}{\lambda}} \cdot (x - w_{\mathrm{ik}}^{t}) \quad (4.4)$$

Where $\varepsilon$is the adaptation step size and λ is the neighbour range. Both $\varepsilon$and λ are decrease while time step $t$increase. After a sufficient number of adaptation steps NG nodes, represented by their corresponding weight vectors, cover the data distribution with minimum error.

NG algorithm does note create or remove nodes. For each training pass all nodes are adapted as opposed to GNG where only the two closest nodes are being affected. The NG model can be seen as a Gradient descent optimization algorithm on a cost function since all weight vectors are adapted with a decreasing step size while increasing distance order.

4.6. Growing Neural Gas

Growing Neural Gas, an incremental neural network [Fritzke, 1995] can learn the topological relationships from an input set of vectors using a variation of the Hebbian rule. GNG is a network that dynamically add or remove nodes and can approximate the input space with higher accuracy compared to a network with predefined structure such as the self organizing feature map. General applications of GNG are vector quantization, interpolation and clustering. Conversely to SOM, GNGs are mainly used for expansion rather than dimensionality reduction.



a. 6 Nodes in R2 **b.** The Voronoi diagram

c. The Delaunay triangulation **d.** both

figure 4.4. The Delaunay triangulation and the Voronoi diagram.

Before analyzing the algorithm operation it is considered necessary to analyze some concepts from computational geometry. Starting with a simple example, assume there exists five vectors in

R2 as depicted in figure 4.4.(a) these vectors can be referred to as nodes. The main property of a Voronoi representation is that for each node a region exists around that node where belonging points are closer to that node than to any other nodes. The Delaunay triangulation is a graph structure where nodes with a common Voronoi edge are connected by an additional edge (Fig. 4.4.(c)). Alternately, it can be defined as a triangulation of the nodes with the additional property that for every triangle the circumcircle of that triangle does not contain any other nodes. Since GNG algorithm is self-adaptive, i.e. it can deploy its nodes and move them independently to represent any given distribution and to create or destroy interconnections, renders it suitable for applications with distributions varying over time. It appeared that cases that are most appropriate are those where change is occurring slowly.

GNG start with two nodes and gradually build a graph in which nodes are neighbour nodes and are connected by an edge. The neighbour information is maintained throughout execution by a variant of competitive Hebbian learning (CHL). An edge is inserted between the two closest nodes when an input signal is applied, measured in Euclidian distance. Depending on the needs of the application, any metric may be used. The shape produced by CHL is called the "induced Delaunay triangulation" which is a special sub-category of the Delaunay triangulation. The induced Delaunay triangulation optimally preserves topology in a very general sense. The action of CHL is of great importance since it regulates movement and insertion of new nodes. The algorithm uses a small number of dependent variables which are constant in time. Additionaly, there is no need to indicate before execution the number of nodes because this will continue to expand until some criterion is met. This criterion may be a performance threshold or an index that reached the maximum size network.

4.6.1 GNG algorithm explained

The execution steps of the algorithm are analyzed bellow. This analysis was considered appropriate to facilitate better understanding of application to experiments.

1. Initially, two nodes randomly positioned are created and connected with a zero age edge. The error of both nodes is 0.

2. One input vector is randomly chosen from the distribution.

3. Find the two best matching units with respect to the input signal $\bar{x}$. If the reference vectors of the nodes are $\bar{W}_s$and $\bar{W}_t$, the distances from input vector are $(\bar{W}_s - \bar{x})$ and $(\bar{W}_t - \bar{x})$

4. The winner node *s* update its local error by adding the squared distance between $\bar{W}_s$and $\bar{x}$

$$\text{error}_s \leftarrow \text{error}_s + (\bar{W}_s - \bar{x})^2 \quad (4.5)$$

5. Best matching unit *s* and all connected neighbors are shifted *towards* $\bar{x}$*by fractions* e_w*and* e_n*of the distance.* e_w,$e_n \in [0,1]$

$$\bar{W}_s \leftarrow \bar{W}_s + \mathrm{e}_w(\bar{x} - \bar{W}_s) \quad (4.6)$$
$$\bar{W}_n \leftarrow \bar{W}_n + \mathrm{e}_n(\bar{x} - \bar{W}_n), \forall n \in \text{Neighbors}(s) \quad (4.7)$$

6. All emanating edges from node *s* increment their age.

7. If *s* and *t* are connected with an edge then this age becomes 0. If they are not connected create a new edge between.

8. Scan all edges for an age larger that $a_{\max}$, remove if any found. If then any nodes appears with no connections, remove them also.

9. If current iteration is an integer multiple of λ and maximum node count has not been reached, insert a new node. This is done as follows.

- Find the node u with largest error
- Find the neighbor v of node u with the largest error.
- Create a new node r between u and v in the position:

$$\bar{W}_r \leftarrow \frac{(\bar{W}_u + \bar{W}_v)}{2} \quad (4.8)$$

- Create edges between u and r, v and r and remove edge between u and v.

- Error variables of u and v decrease and error of node u is asigned to error of node r.

$$\mathrm{error}_u \leftarrow a * \mathrm{error}_u \quad (4.9)$$

$$\mathrm{error}_v \leftarrow a * \mathrm{error}_v \quad (4.10)$$

$$\mathrm{error}_r \leftarrow \mathrm{error}_u \quad (4.11)$$

10. Decrease error variables of all nodes j by a factor β

$$\mathrm{error}_j \leftarrow \mathrm{error}_j - \beta * \mathrm{error}_j \quad (4.12)$$

11. Repeat from step 2 until a stopping criterion is met.

4.7. Self Organization in Robot Localization

Static sensor signals clustering based on self organizing maps have been used before for robot localisation. Early adoptions of this method include an autonomous robot with map building ability through a SOM network [Nehmzow et al., 1991]. Different unsupervised neural network architectures have been used to realize topological relationships between input and output space. Wichert, [1997] proposed an image based navigation system based on self-organization of visual snapshots. Werner et al., [2006] developed a system that extracted color histograms as input vectors for a SOM algorithm. Growing neural gas has been used for a visual based self-localisation of a mobile agent in indoor environment by Baldassari et al., [2003]. Images acquired from a camera, moving in a pathway, in order to build an implicit topological representation of the environment. These simulations dictated the effectiveness of the GNG model in recognition speed, classification tasks and in particular topology preserving as compared to the popular SOM model. The performance gap ascribed to the fact that nets dynamically adding or removing nodes can approximate the input space more accurately than a network with a predefined structure and size such as SOM. As can be seen (figure 4.5) a predefined grid such as SOM can not describe a topology with the same accuracy as compared to a network that dynamically add or remove nodes and edges. This is more clear when the distributions depict irregularities such as convexities and holes. This is true also since SOM resembles a losy compression scheme by applying a data projection from a multidimensional space, where perceptual signatures are described, to preferably only a two dimensional space. Other proposed methods based on principal component analysis (PCA) can compute global only image properties [Tamimi et al., 2004]. These navigation systems suffer from the problem of perceptual aliasing, the fact that different physical locations can give rise to identical sensory perceptions.

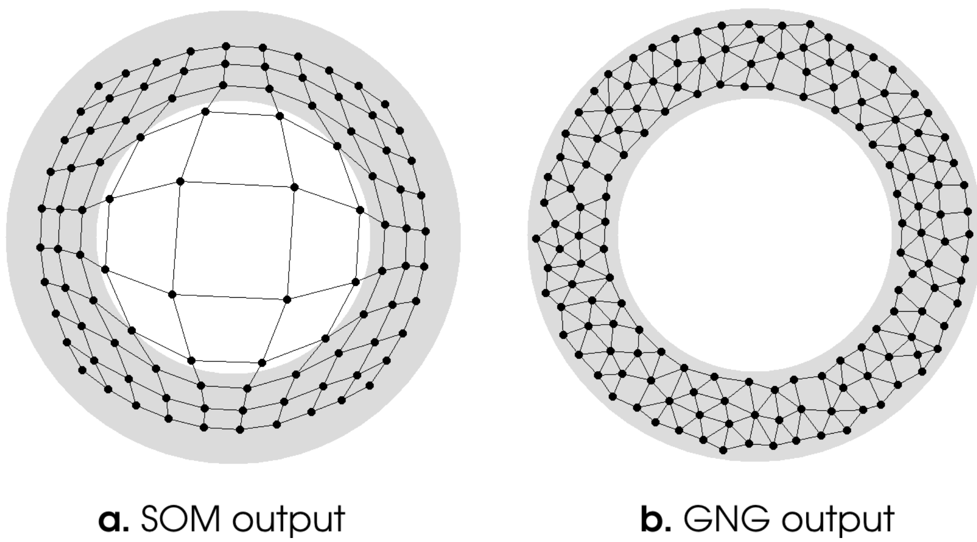


Figure 4.5. Topology preservation for SOM & GNG

4.8 Modular neural networks

Biological studies showed that the human brain functions not as a single massive network, but as a collection of small networks. The combination of the desirable features of different neural computation ways gave birth to the concept of modular neural networks, in which several small networks cooperate or compete to solve problems. An idea behind a multi-network approach is to build a voting system engaging a number of dedicated networks assigned purposely.

Gerecke et al., [2003] proposed an approach to mobile robot localization that makes use of ensembles of self-organizing maps. A test and select approach applied with the individual evidence vectors (IEVs) and common evidence vectors (CEV) architectures. The study was run on a simulation of a Nomad-200 mobile robot, encircled evenly with 16 ultra sonic and 16 infra red sensors. Ensembles showed significant improvement over their single SOM counterparts but with a loss of speed. Yamada, [1996] described the application of a SOM that recognizes rooms using behavioral sequences with only low-sensitive local sensors, and results evaluated in a real world robot.

These approaches applied in the standard way, however, has its drawbacks. One of them is the fact that the localization stage is strictly separated from the learning stage. All images are being captured during the exploration phase, and only then it is possible to construct the model. The model built in this way can not be modified unless original images have been reserved. To update the model with new images, a new data-base with training vectors should be made. Therefore, standard approaches are not optimal for performing simultaneous learning and localization. Furthermore, the original images take a lot of storage space.

4.9. Summary

There are two reasons why self organizing maps should be used in preference to pre-installed, fixed mechanisms in autonomous mobile robot navigation. The first one is a methodological consideration, the second reason has to do with the nature of the robot's perception of its environment [Nehmzow, 2000].The first argument, therefore, for using self-organisation in mobile robot navigation is that it can reduce the amount of predefinition put in by a human operator. Robot sensors are subject to noise. Sensory perception of a robot can be plain wrong and therefore misleading, contradictory, e.g. sensor information regarding the same object coming from different sensors, or useless, e.g. no meaningful information from sensors with respect to the task being performed by the robot.

Learning mechanisms of self-organization make use of the data that is actually available, without prior assumptions. This takes care of useless data. Self organizing mechanisms such as Kohonen SOM can cluster information topologically, which addresses the problem of noise. Finally, if a wrong sensor signal is inconsistent or in contrary to neighbor sensors providing the correct reading, self-organization have proven to be a good method of eliciting that information. The second argument in favour of self-organization, therefore, is that such mechanisms are well suited to process the kind of data obtained from robot sensors. Because of these properties, they considered to be suitable for the needs of this thesis. All unsupervised learning algorithms that are analysed in this chapter have been used to model topology-based mapping. By abstracting visual sensory data, these algorithms used to represent scene snapshots into discrete sub-spaces, forming a sensory representation of the environment's appearance.

5. Representing Time and Space

Chapter 5 refers particularly to the concept of spatial representation and outlines the qualitative characteristics that these representations should have. The chapter also review various methodologies as well as the critical points of current research contribution. In addition, an investigation of the combined representation of space and time and the ways in which robot navigation could benefit. Finally, a variety of inspiring biologically plausible models are introduced.

5.1. The need for a map

As explained in Chapter 2, during the design of an autonomous mobile robot, the space representation model must simultaneously support the functions of localization, navigation and map-building. The model of the environment should have the following attributes:

1. It should incorporate different types of features extracted from raw sensor data.

2. The spatial representation should be able to be searched efficiently. Features associated with places should be transformable to whatever format may be required for both prediction as well as correspondence matching between extracted and stored features.

3. Information accumulated in the form of a map representation should be diversified about each object to facilitate disambiguation from other objects.

4. Maps should allow for the easy inclusion of new areas and the update of the parameters of stored areas.

5. The environment should at least be distinguished into two main regions. Positive space representing current robot position and negative space the area in which the robot can move freely.

5.2. Environment Representation

The most natural representation of a robot's environment is a map. In addition to representing places in an environment, a map may include other information, including properties of objects, regions that are unsafe or restricted to traverse, or information of prior experience. An internal representation of space can be used by a robot to pre-plan and pre-execute tasks that may be performed later. Within this map a mobile robot must be able to estimate its pose. However, in many practical applications like exploration tasks, a map is not available or it is highly uncertain. Therefore, in such situations the robot must build its own map. The goal for an autonomous robot is to be able to both construct and use a map.

From a biological perspective, some animals, like insects, have evolved in such a way that a triggered response is sufficient to keep them alive. For these animals the environment is not interpreted as a map. A little more advanced navigation capabilities may dramatically improve skills related to autonomy [Trullier et al, 1997].

5.2.1. Geometric Representation

Geometric maps are quantitative representations made up of discrete geometric primitives like lines, polynomial functions, points and so forth. They characterized by large scale detail. The primary shortcoming of geometrical model based representation relates to the fact that they can be difficult to infer reliably from sensor data [Sim et al., 2003]. Geometric maps, such as occupancy grids [Martin et al., 1996] representation employs a tessellation of space into cells (typically 2D or 3D) where each cell stores a probabilistic estimate of its state. Hence, each cell represents a rectangular area of the environment and it stores the value that indicates the occupation state for this area. Usually this is done by labeling the cells with “unknown”, “free” or “occupied” values or with a value that represents the likelihood of the cell being occupied or not. To fill all the cells with the appropriate values using probabilistic methods can be very expensive in terms of computing power, which becomes more expensive as the environment map increases. For some real time applications occupancy grids may not be a viable solution [Dissanayake et al., 2000]. The ability to represent the environment and the possibility of incorporating new data even if they originate from different sensor types, are the main advantages of geometrical maps.

5.2.2. Topological Representation

The term topological map refers to a map which captures the connectivity of the environment and has been simplified so that only vital information remains and unnecessary detail has been removed. These maps lack geometric information like scale, distance or direction but the relationship between points is maintained. The simplicity of topological maps support much more efficient planning than metric maps [Sim & Dudek, 2003],

The key to a topological relationship is based on an abstraction of the environment in terms of connectivity between discrete regions or objects with edges connecting them. In the purest form, this may involve a complete absence of metric data. A robot employing this representation has no real understanding of the geometric relationship between locations in the environment but the enclosed information is sufficient for the robot to conduct point to point motion. The use of graphs has been exploited by many robotic systems to represent the environment. Early efforts include the work of [Nehmzow & Smithers, 1991] that used a self-organizing map prior to building a map. Another proposed method automatically generates topological maps based on Delaunay triangulation [Tarutoko et al., 2006].

A graph is a kind of abstract data structure that consists of points or nodes connected by links called lines or edges. Each node corresponds to one of the unique landmarks and each edge corresponds to known paths between landmarks. If the environment consists of networks of corridors and rooms like in many indoor environments such as office building or a hospital, it is less complex to specify the topology of important locations and their connection suffice.

Evidence also indicates that humans represent physical spaces topologically rather than geometrically [Lynch, 1972] . For example, when providing the clues needed to lead someone in a building, directions are usually of the form “go down the hall, turn right at the elevator, open the second door on your left," rather than in geometric form.

5.3. Spatio -Temporal Reasoning

Describes how the relationships between regions change with time [Lynch, 1960]. Refers also to the ability to visualize spatial patterns and mentally manipulate them over a timeline sequence. This ability, often referred to as "thinking in pictures", is important for generating and conceptualizing solutions to multi-step problems that arise in areas such as simultaneous localization and map building for mobile robots. The key is to identify an instantaneous relation occurring during a transition between events.

There have been a number of approaches to representing time in unsupervised neural networks. Temporal Kohonen Map [Varsta et al., 2001] is a variation of standard Kohonen SOM [Kohonen, 1995] with time delayed feedback. The network learns by associating current input to previous activity states. Hence, each neuron responds to a sequence of inputs. A possible drawback is the difficulty of determining the proper length for the delay line. Recurrent SOM (RSOM) [Koskela et al., 1998], can be presented as an enhancement of temporal Kohonen Map, finds regularities and nonlinear dependencies that exist in the data. Usually this model predicts the future of a temporal process. Both networks are good for prediction and auto completion of sequential patterns but proved not well suited for episodic memory representation.

Nehmzow et al., [1991] proposed a system based on self-organizing maps to recognize simple environments. Another behavior based approach proposed by Yamada, [2004]. In this work, sequences of motor action states tracked and transformed into input vectors for a self organizing network. Yamada cited that these behavioral input vectors are significantly sensitive to noise like small objects.

Another localisation mechanism for autonomous mobile robots based on current and preceding perceptions of the world was developed by Nehmzow, [1999]. This system uses both spatial and episodic information to establish the robot's position in the world. The system consists of two stages. The first stage processes raw sensory perceptions of the robot by clustering them using a self organizing map. The second stage then clusters the last τ perceptions in order to encode episodic information. Through this processes, meaningful internal representations of a mobile robot's environment emerged, without any external intervention. Episodic mapping mechanism outperformed static mapping mechanism offering disambiguation of two locations with similar perceptual signatures in different time-line locations. Although the learning process is performed offline.

5.4 Rats and Honeybees

Recently, a large number of autonomous agents has been built but non of these systems has reached the flexibility and navigation capabilities of animals or even insects. This has motivated robotics researchers to investigate biologically inspired mechanisms that can be implemented on autonomous mobile robots.

5.4.1. Local Rule Concept

The local rule concept [Dyer, 1991], [Wehner et al., 1990] proposes that navigation is guided by multiple and independent memories. In novel situations, the animal's behavior is dominated by one of these memories. Studies of insects [Collett, 2010] suggests that when they travel they memorize sequences of visual snapshots experienced on routes that lead to food and nest locations. In order to reach goal positions the insect correlates currently perceived images with an image previously memorized and sets an appropriate goal. The correlation is a measure of association (resemblance) between two images to find those portions that match according to the measure of correlation.

5.4.2. Cognitive map concept

Cognitive maps, a term originally coined by Tolman, [1948], are types of mental processing humans use to structure and store spatial knowledge, allowing the mind to visualize images in order to reduce cognitive load, and enhance recall, learn and decode information about the relative locations and attributes of phenomena in the spatial environment.

Humans and animals are thought to form maps from their environment to aid navigation. Lynch, [1972] developed a library of generic components which he hypothesized that humans use to construct cognitive maps of urban environments. He also introduced the concept of place legibility, which is essentially the ease with which people understand the layout of place.

Lynch isolated distinct features of a city, with an intention to describe the essentials that renders it so vibrant, and attractive to people. To understand the layout of a city, people first and foremost create a mental map. Mental maps of a city are mental representations of what the city contains, and its layout according to the individual. These mental representations, along with the actual city, contain many unique elements, which are defined by Lynch as a network of paths, edges, districts, nodes, and landmarks.

First, paths are linear separators, examples includes roads and sidewalks. The second element, edges, are all other linear separators not included in the path group, such as walls, and seashores. Third, districts are logically and physically distinct sections of the city, usually relatively substantial in size, which have an identifying character about them. A wealthy neighborhood such as Beverly Hills is one such example. Fourth, nodes are the strategic points which exhibit similar characteristics. Prime examples of nodes include a busy intersection with the same type of light posts. The fifth set of elements, are the physical objects, in sharp contrast to their immediate surroundings, that act as reference points. Landmarks can be a church spire, mountain, school, or any other object that aids in orientation when way-finding.

Gould, [1986] experimented with honey bees in order to show that the integration of multiple memories leads to the generation of novel information. In conflicting situations, animals would thus be expected to be able to find effective solutions to navigational problems. These representations suggest geostable map-like memory organisation supported by celestial landmarks. Considerable evidence indicates [Cosens & Toussaint, 1985] that wood ants Formica Aquilonia use local landmarks to aid goal reaching. When changes were made in the environment, such as reallocating rocks, the insect updated their paths towards goal reaching.
A self localization system based on the cognitive map model presented by Gerecke et al., [1999]. The system employs a self organizing map to provide a list of candidate locations for which the robot is likely to be located. The localization method based on the self organizing map to disambiguate the output by moving the robot a small distance away from the initial position and accumulating evidence. The results show that the location of the robot can be computed with a satisfactory degree of reliability and accuracy within a fairly small radius of uncertainty.

5.4.3. Place Cells

The existence of "cognitive maps" is inferred from the ways in which rats solve certain spatial problems. Problems like maze solving seem difficult and require efficient and intelligent solutions. Maps are postulated because spatial problem solving can not be accomplished with only the overall structure.

Place cells were first described by O'Keefe et al., [1971]. Based on this discovery, O'Keefe et al., [1978] hypothesized that the primary function of the rat hippocampus is to form a cognitive map of the rat's environment. Both internal and external stimuli support the firing of these cells. Place cells depends largely on visual ques but are also active in the dark, suggesting that place cells firing may also refer to a complementary odometry behavior. Other neurons with spatial firing properties are the grid cells, head direction cells, and spatial view cells. Franz et al., [1998] presented a topological representation scheme that employs visual homing strategies inspired by these findings of insect ethology. Moreover, Butz et al., [2010] inspired by the rat's hippocampus investigated the generation of a sensorimotor cognitive map based on a variation of the growing neural gas algorithm. A method have been presented by Hafner [2000], that use a computational model of cognitive maps for robot navigation purposes. The model, based on a self organizing algorithm, creates a topological map during an exploration phase. Hafner suggests that the self organizing code-vectors can be seen as equivalent to “place cells”. Another computational model of “place cells” spatial learning capabilities have been proposed by Strosslin et al., [2005], that use visual and self motion information in order to feed the input neurons of an unsupervised Hebbian learning algorithm. The approach validated both in a real and a simulated robot.

Takahashi et al., [2001] investigated a method for robot navigation based in self organizing maps and reinforcement learning. The proposed system, inspired by hippocampal place cells on rat's brains, consists of units which are being activated for specific locations within an environment (figure 3). In order to generate a map adapting to a real world environment, data derived from locations occupied by the robot and topologically organized with a Kohonen self organizing map.

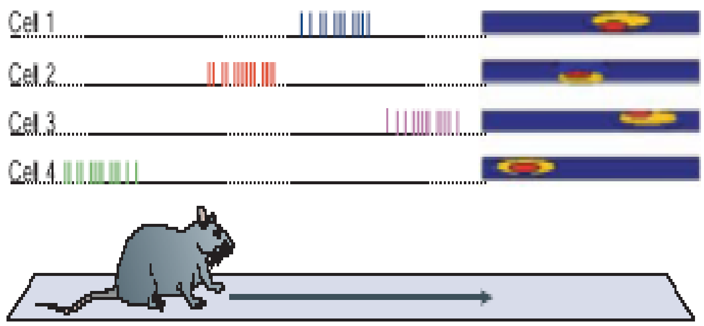


Figure 5.1. Hippocampal neurons firing patterns [Kazu Nakazawa et al., 2004]

Barrera et al., [2008] presented a robot architecture with spatial cognition and navigation capabilities mimicking place cells neurons in hippocampus. The robot controller, based on hebbian learning, was able to build a topological map during exploration using reinforcement learning by means of an Actor-Critic architecture to enable learning and unlearning of goal locations. For a review about the approaches that implement biomimetic behaviors in the field of autonomous robot navigation, see [Franz et al., 2000].

5.5. Summary

Animals and even insects depict spatial interpretation abilities that dramatically outperform current methodologies for robotic navigation. Evidence form the neurosciences supports different hypotheses of how navigation skills are structured in the animal's brain. Investigating the structure and function of biological systems as models for the design of autonomous machines, may lead to autonomous robots which exhibit great navigational abilities.

This survey provides empirical evidence which support a fundamental premise for a bio-inspired topological and appearance based map building model. More specifically, formations of neurons in the hippocampus of rat's brains exhibit high activation rates whenever an animal is in a specific location in an environment, while insects seem to integrate multiple sensor signals to generate novel spatial information. Moreover, according to the Cognitive Map Concept, surrounding environments may be decomposed into simple topological maps, in the form of a graph structure, consisting of significant places connected with known pathways.

6. Learning and Forgetting

This chapter first considers the nature of two memory models, semantic and temporal. Then focuses on the problem of catastrophic forgetting in connectionist architectures, such as artificial neural networks, and explore ways that would prevent this undesirable effect. The chapter continue with proposed solutions from the literature ranging from rehearsal to recurrent learning of sequences.

6.1. Semantic Memory

Semantic memory includes generalized knowledge that does not involve memory of a specific event. Semantic memory refers to the memory of meanings and understandings, and other concepts related to knowledge and unrelated with the experience. Semantic and episodic memory constitute what is known as declarative memory, which is one of the two major divisions in memory.

6.2. Episodic Memory

Episodic memory refers to the memory of events, times, places and associated emotions and other knowledge in relation to an experience. It is thought of as being a "one-shot" learning mechanism. and needs a single exposure to an episode to remember it. Semantic memory, on the other hand, can take into consideration multiple exposures to each referent and the semantic representation is updated on each exposure.

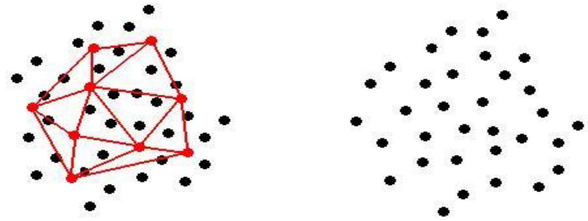
Episodic memory can be thought of as a data structure that ties together items in semantic memory. For example, semantic memory will reveal what a landmark looks like. All episodic memories concerning a district location will reference this single semantic representation of the landmark and, likewise, all new experiences of this landmark will modify the single semantic representation of this landmark. An episodic mapping mechanism presented by Nehmzow, [1999] use a dual layer SOM network to classify both perceptual and episodic information from a mobile robot.

6.3. Catastrophic Forgetting

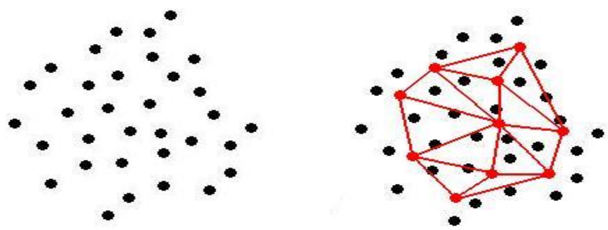
In most artificial neural networks, learning capabilities suffer from sudden and total forgetting of all previous learned information. Carpenter et al., [1988] describe this problem with the analogy of a person growing up in one city, before moving to a second city. Catastrophic interference would be if the process of learning about the second city prevented this person from remembering how to reach the house in which they grew up. Catastrophic interference is an implausible aspect of natural cognitive systems. Humans and animals do not forget suddenly. They are designed to adapt to unstructured and dynamically changing environments through interaction with their own experiences. This process of adaptation without being guided or managed is referred to the literature as unsupervised learning.

Neural networks memory function is served from a single weight set. This specificity gives to these networks the ability to generalize, to be fault tolerant and noise immune. Although this lack of modularity also causes interference to previously stored patterns with the newly arrived patterns during learning [French, 1999].

Catastrophic interference is a symptom which indicates a more general condition occurring in almost all memory systems, the so called plasticity-elasticity dilemma [French, 1999]. Representations developed by neural networks should be plastic enough to adapt to changing environments and learn new information, but stable enough so that important information is preserved over time. Both are desirable properties but the requirements of stability and plasticity are in conflict. In case of graph based topological representations, stability dictates mapping of input space while keeping a representation structure and relationships. Plasticity depends on extending input space and thus topology. A balanced model is difficult to achieve [Robins, 1995]. Excessive plasticity often dramatically labelled as 'catastrophic forgetting' can be summarized as follows. After every original training is finished the network is exposed to the learning of new information, then the originally learned information will typically be greatly disrupted or lost. The problem of catastrophic forgetting was initially mentioned by McCloskey et. al., [1989]. The problem of catastrophic forgetting can be seen in figure 6.1. During first step a Self-organising map learns a random tolopogy. During the next step a new distribution is presented to the network. The algorithm experience total forgetting of the old distribution. This is also known as the luck of elasticity problem.



(a) The network learning the first distribution.



(b) The network learning the second distribution.

figure 6.1. Catastrophic forgetting for a Self-Organising Network. Newly learned information completely erases all previously learned information.

6.4 Rehearsal and Pseudo-Rehearsal Learning

Several recent cases have highlighted the potential problems of catastrophic forgetting and explored various solutions mainly with variations of Hopfield [Robins et al., 1998] or Back-propagation type networks [French, 1999], [Ans et al., 2000].

A particular connectionist approach to avoid catastrophic interference is to ‘rehearse’ the history of events as new learning occurs. This solution is unrealistic for most applications since it requires permanent access to all previously experienced events. The key issue as proposed by Ans et al., [2004] is to use a pseudo-rehearsal mechanism in place of a true rehearsal process. Each time a new item is to be learned, a temporary set of ‘pseudo-items’ should be created and learned alongside the original item.

This simple algorithm works remarkably well, in that it appears to substantially reduce interference between sequential training items while still allowing new information to be learned.

Pseudo-pattern data set can be self generated from just feeding the target neural network with noise and tracking the corresponding output. These internally generated patterns represent an approximation of the static learned information so far.

Some pseudo-rehearsal mechanism implementations [Ans et al., 2000] use reverberating process to generate pseudoitems. These pseudoitems act as attractor patterns generated from reverberations within a recurrent network structure. The aforementioned implementations have in common the use of the back-propagation learning algorithm and a dual-network architecture where two complementary networks exchange pseudo-items. [McClelland et al, 1995] suggested that the hippocampus and neocortex act as separately but complementary memory systems. Specifically, the hippocampus short-term memory storage gradually transfers memories over time into neocortex for long term memory storage.

It has been suggested that this pseudo-rehearsal process could, in fact, be the mechanism that the human brain uses to correlate old and new information. Robins has viewed the pseudo-rehearsal method as an equivalent to the biological function of dream sleep consolidation hypothesis, which essentially explains integration of newly acquired information into existing long-term memory [Robins et al., 1999].

6.5. Recurrent Neural Networks

Unlike feed-forward networks, recurrent networks can be sensitive and adapt to past inputs. It is well known that conventional neural networks can be successfully used to define any function as long as there is a large enough number of hidden neurons. The fundamental difference compared with a feed-forward architecture is that they operate both in an input space and a space of hidden internal states. In its basic form, a recurrent network is a feed forward network with an additional connection from the hidden unit to itself.

This seemingly small change to the network has a big impact in the overall behavior. When an input occurs, neurons calculate their outputs in the same manner as that of a feed forward network. However, input now contains a term which reflects the state of the network (the hidden unit activation) before the pattern was seen. When patterns are subsequently presented, the hidden and output units states will represent a history of all patterns that have been learned so far. The behavior of the network is based on a temporal sequence of inputs. Feedback connections can be freely configured from any unit to any other, even to the same unit.

6.6. Elman network

This is a simple recurrent network originated by Elman, [1990]. In these simple variations for each time step, a copy of the data of the hidden layer is transferred to a temporary layer (figure 6.2). The operation is being analyzed as follows:

- Copy inputs for time *t* to the input units
- Compute hidden unit activations using net input from input units and from copy layer.
- Compute output unit activations as u1.0sual
- Copy new hidden unit activations to copy layer.

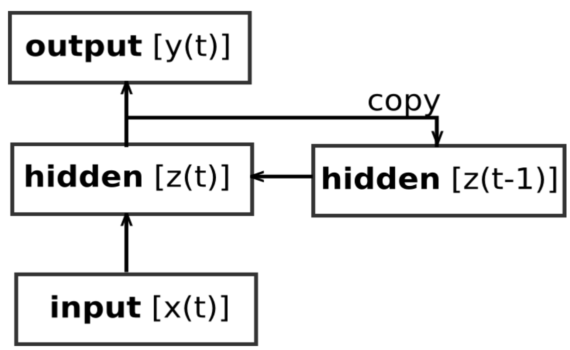


Figure 6.2. Elman network architecture.

6.7. Summary

Almost all natural cognitive systems gradually forget previously learned information. Artificial neural networks exhibit catastrophic forgetting of old information as new information is acquired. The ability of these networks to generalize is also the cause of completely erasing previous learned information. Gradual forgetting is not a totally undesirable effect. Forgetting can be interpreted as making space for new knowledge. In the case of an online robot mapping procedure, in a dynamic or changing environment , unlearning parts of the map is highly desirable in order to update the spatial representation. As long as the memory requirements are limited, when online learning is employed, past correlates with 'history' which leads to memory requirements independent of their size.

This survey explores various solutions to the problem of catastrophic forgetting which have not yet been applied to unsupervised learning networks. Additionally, regarding a case where a mobile robot navigates in an unknown environment, the space of possibl sensor perceptions are not available at once. By combining a self organizing algorithm, that acts as the main topology mapper, with a self-refreshing mechanism, a robot could be able to map unknown environments incrementally and online.

7. Scene Interpretation

Chapter 7 presents an overview of the techniques used for global feature detection and extraction. At this stage, the problem of robot self-localization is reduced to a content based image retrieval problem. Both color and texture image descriptors are proposed to ensure discrimination robustness. Finally, In order to analyse texture information, two common choices are presented, Gabor filters and Wavelet decomposition analysis.

7.1. Image analysis

Image representation as a grid of pixel values is rather unnatural. An image slightly rotated or translated, changes completely the order of pixel values but to a human observer the data still looks very similar. Moreover, human visual perception system is able to understand the *gist* of a novel image, very fast, independent of it's complexity [Oliva et al., 2006]. Image analysis aims to extract meaningful and quantitative information from such large data matrices. Siagian et al., [2009], presented a robot localization system using biologically inspired vision, where image *gist* computed as global statistical signatures of the images. These signatures where further processed using a Monte Carlo localization algorithm. Most techniques are useful for a small range of tasks compared to human visual system that is capable of generic analysis for a wide range of tasks.

7.2. Image Indexing

Content based image retrieval (CBIR) is the application of computer vision to the image retrieval problem, that is, the problem of searching for digital images in large databases. 'Content-based' means that the search will analyze the actual contents of the image. The term 'content' in this context might refer to colors, shapes and textures, or any other information that can be derived from the image itself. Without the ability to examine image content, searches must rely on meta-data such as captions or keywords, which may be laborious or expensive to produce.

The term CBIR seems to have originated in 1992, when it was used by T. Kato to describe experiments into automatic retrieval of images from a database, based on the colors and shapes present. Since then, the term has been used to describe the process of retrieving desired images from collections on the basis of syntactical image features. Smith et al., [1996] proposed a technique which use color content to evaluate retrieval from a database of images and video. The techniques, tools and algorithms that are used for CBIR originate from fields such as pattern recognition, signal processing, and computer vision.

A set of image signatures extracted by a mobile robot during terrain exploration, can be manipulated as a large abstract image database. Fraundorfer et al., [2007] proposed a vision-based localization and mapping method using image collections. Based on this method, robot's world represented as a linked collection of way-point images. The fundamental scheme behind a CBIR approach for self localization can be based on a measure of resemblance between the currently acquired image and the base of images stored as perceptual signatures regarding familiar terrain. Gonzalez et al., [2002] suggests a qualitative position refinement technique that localize a rover when it comes back in a previously perceived area, using an image indexing technique on panoramic views based on principal component analysis. The limitation of this procedure is that cannot perform incrementally, because all learning images are required to compute the subspace.

7.3. Image content descriptors

The remainder of this chapter describes methods to extract meaningful information from images so that they can easily be compared. Visual descriptors, are the building blocks to correlate pixel information with image content information as conceptualized by humans or animals.

7.3.1. Color Descriptors

Color is both subjective and personal. Color perception is a reaction in the brain to specific visual stimuli. Examining images based on the color information they contain is one of the most widely used techniques because it does not depend on image size or orientation. Color searches usually involve color distributions like histograms or correlograms, though this is not the only technique in practice. The aim of color spaces is to aid the process of describing color between people or between machines or programs. Each color space is suitable for a specific range of applications or specific optical devices. Color spaces, are abstract models, that describe the way colors are represented as a group of three or four components.

7.3.1.1. RGB Color Space

This is the most well known space that is based on a tri-chromatic theory. The system is additive and very common in computer systems. The space is defined as a combination of the three chromaticities of Red Green and Blue primary colors. Since human vision system works in a similar way the RGB color model is well suited for computer graphics. Main characteristics include ease of implementation and device-independency. The main drawback is that color interpretation is not linear with visual perception.

7.3.1.2. LAB Color Space

The LAB space consists of a luminosity layer L, chromaticity-layer *a* indicating where color falls along the red-green axis, and chromaticity-layer *b* indicating where the color falls along the blue-yellow axis. All of the color information is in the *a* and *b* layers. The advantage of such a representation, is the ability to measure the difference between two colors using a simple distance metric like Euclidean. Moreover, LAB color space has been reported to perform better than RGB in terms of clustering and classification when unsupervised learning algorithms such as NG and GNG are used [Marimpis et al., 2012].

The L*a*b* color space designed to approximate human visual perception rather than the output of physical devices. Compared to other color spaces like RGB, perception of lightness or color uniformity matches more accurately the human visual perception.

7.3.2. Color Histograms

Color histograms are considered to be robust for robot map-building [Werner et al., 2007] and due to their statistical nature, provide a complete rotationally invariant representation when employed with panoramic cameras. The main drawback, when referring to classification tasks, is that the representation ignores shape and texture lying on images. Color histograms exhibit considerable overlap when used to describe an image. Two images with different content may have identical histograms because they share the same color information. Additionally, color histograms are subject to noise interference such as varying lighting conditions and quantization errors. However, color information is faster to compute, making it suitable for real-time applications, and also suitable for appearance based image retrieval tasks where object detection and recognition is not practically applicable.

7.4. Texture

There is not an actual definition that describes the concept of texture. Structural approach deals with image textures as a set of fundamental units named textured elements (texels). This approach is oriented on analyzing artificial textures and it is possible to describe their spatial relations by using voronoi tesselation applied on texels. Another approach,   provides information about the spatial arrangement of color intensities in the image. This approach, based on statistical analysis, tries to describe them with clear quantitative measures. In general, this is well applied to natural textures because these are made of irregular patterns.

Texture measures search for visual patterns in images and how they are spatially defined. Practically texture measures depend on how many textures are detected in the image and where in the image these textures are located. Later studies on human vision shows that the retina and brain have receptive fields (filters) responding only to specifically oriented lines within a region of the retina, known as the complex cells [Field, 1987]. Complex cells are insensitive to local changes in feature positions therefore attractive for low dimensional representation of images and generalization.

7.4.1. Gabor Filters

Gabor filters are a common choice for texture analysis. Gabor analysis [Gabor, 1946] is based on linear bandpass filters whose impulses response is a Gaussian windowed sinusoid. Because windowing operation express a point-wise product of the Gaussian and sinusoid function, the Fourier transform of the filter kernel is the convolution of the Fourier transform of the sinusoid function and the Fourier transform of the Gaussian function. A 2D Gabor function is described as follows:

$$g(x,y;\lambda\vartheta\psi\sigma\gamma)=\exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\psi) \qquad (7.1)$$

where

$$x'=x\cos\vartheta+y\sin\vartheta \qquad (7.2)$$

and

$$y'=-x\sin\vartheta+y\cos\vartheta \qquad (7.3)$$

In this equation, λ represents the wavelength of the cosine factor, ϑ represents the orientation angle of the Gaussian envelope in degrees, ψ is the phase offset in degrees, and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

The main motivation to use 2-D Gabor filter banks is to represent images in a way somewhat similar to complex cells. Other advantages of Gabor filters is noise cancelation and redundancy reduction. Gabor filters have been proved very efficient on detecting relationships among image elements, as collinearity, parallelism, connectivity and repetitivity [Yvas et al., 2006].
Gabor filter banks are directly related to Gabor wavelets (Morlet wavelets), since a family of functions is built from dilations and rotations of a single mother function. In general, Gabor wavelets are not computationally cost effective. Usually, frequency domain sampling is achieved via a Gabor filter bank with various scales and rotations as can be seen in figure 7.1. The filters are convolved with the image, resulting in a so-called Gabor space. This process is closely related to processes in the primary visual cortex. Relations between filtered regions are very distinctive between objects in an image and important activations can be extracted from the Gabor space in order to create a sparse object representation.

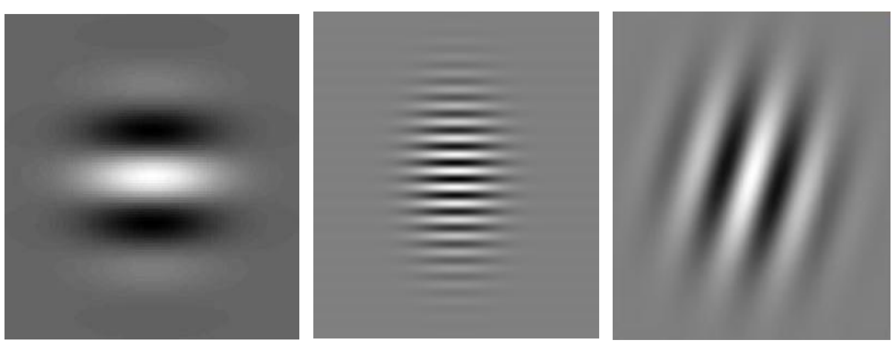


Figure 7.1. Examples of 2D Gabor filter kernels for different scales and orientations.

Actual discovering and localization of textures requires filters localized both in spatial and frequency domain. The main approaches are either a predefined tesselation of the frequency plane (figure 7.2) or a filter kernel configuration suited to a particular problem or family of images [Manjunath et al., 1996]. The first approach leads to large number of filtered images thus large dimensional feature space [Campbell et al., 1997]. Also, this approach may not be optimal for specific tasks because Gabor wavelets are non-orthogonal and the filtered images can represent redundant information [Manjunath et al., 1996]. Many attempts have been reported trying to optimize the number of filters used. Another common approach to filter set selection is that of peak localisation in which local peaks in the frequency domain are found and filters centered around these are chosen.

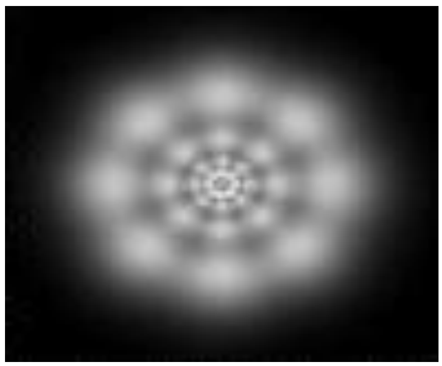


Figure 7.2. A daisy shaped filter kernel configuration is a predefined tessellation of the frequency plane, consisting of overlapping filters whose centre frequencies lie on concentric circles, logarithmically spaced, and centered at the origin.

7.4.2 Wavelets

Wavelets are wave-like oscillations that exhibits specific properties that make them useful for signal processing tasks. Wavelets are mathematical functions that are used to decompose a given function or a signal into different self-similar components. The analysis is similar to Short Time Fourier Transform (STFT) analysis. The target signal is multiplied with a wavelet function just as it is multiplied with a window function in STFT, and then the transform is computed for each segment generated. However, unlike STFT, the width of the wavelet function changes with each spectal component. The advantage over STFT is that at high frequencies presents adequate time resolution and poor frequency resolution, while at low frequencies gives good frequency resolution and poor time resolution. In case of a continuous wavelet transform the wavelet functions are scaled and translated versions of a 'mother' wavelet function.

$$X_{\mathrm{WT}}(\tau,s) = \frac{1}{\sqrt{((s))}} \int x\,(t)\psi\left(\frac{\tau-t}{s}\right)\mathrm{dt} \quad (7.4)$$

Where τ is the translation parameter that locates the wavelet function as it is shifted along a signal. This parameter holds the time information of the signal. The s parameter defines the scale of the wavelet and denotes the frequency information. Scaling expands or compress a signal. Large scales correspond to low frequencies and compress the signal providing detailed information, while small scales correspond to high frequencies and compress the signal and provide global information.

Wavelet analysis is a promising tool and provides an appropriate starting point for image representation in a way also resembling that of complex cells. Wavelets have been used for mining general image characteristics out of images in spatial-frequency domains. Images can be analysed, containing both natural objects like trees, sea-surface, vegetation [Palamas et al., 2006], and artificial ones like buildings or furniture.

Actual discrete wavelet analysis is performed by applying a degradable filter bank to a signal (figure 7.3). These filters have different cut-off frequencies at different scales so different time-scale representations of the signal are obtained. In the case of a 2D signal, such as images, the wavelet transform decomposes an image into a low resolution image and a series of detail images. Low resolution image is obtained by iteratively down-sampling the target image. Detail images contains information isolated with a hi-pass filter. The low frequency content is the most important part since it is what gives the signal its identity. The high frequency content imparts flavour or nuance.

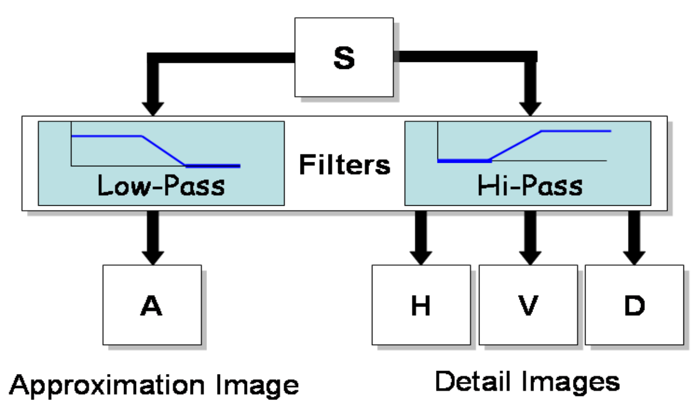


figure 7.3. Discrete wavelet transform with a degradable filter bank.

7.4.2.1 Haar wavelet

This wavelet is one of the oldest and well known. It is also one of the simplest. Actually, Haar wavelet is a simple step function $\psi(t)$taking values 1 and -1 on [0,1/2) and [½,1) respectively (figure 7.4). Every continuous function can be approximated with a Haar wavelet. Scales and translations can be described with the following formula.

$$\psi_{\mathrm{jk}}(x)=\mathrm{const}\cdot\psi\left(2^{j}x-k\right) \quad (7.5)$$

The Haar wavelet operates on image data by calculating the sums and differences of adjacent elements. The Haar wavelet operates first on adjacent horizontal elements and then on adjacent vertical elements.

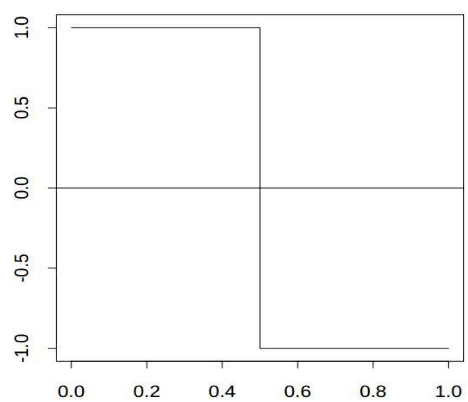


Figure 7.4. Haar wavelet

7.5. Shape

Shape refers to a particular region which contains important semantic information. Shapes are often determined with segmentation or edge detection operators. Accurate shape detection is very difficult to completely automate and require highly structured and controllable environments.

7.6. Summary

Image feature extraction refers to the problem of transforming the input data from a camera into a reduced representation. Low level, global image features, based on color and texture, have been proved particularly descriptive for content based image retrieval procedures and are widely used.

Moreover, the human visual perception system can understand the meaning of an image in a single glance, independent of it's complexity. Behavioral evidence on fast scene perception, suggests that a scene can be estimated from global image features, providing a statistical summary of the spatial layout properties. This section tries to reduce the problem of localization and map building to a content based image retrieval problem and to substantiate the choice of feature selection and extraction methods that have been used for carrying out the crucial experiments of this thesis. The main motivation to use 2-D Gabor filters and Wavelet analysis for texture discrimination is to represent images in a way somewhat similar to complex cells in the virtual cortex.

8. Appearance-based Map Building

The following analysis carried out in order to verify the validity of the models, which have been analysed in the previous chapters, limiting the scope to the problem of map building for a mobile robot. The first section of this chapter, encompass research results in the fields of visual feature selection, extraction and scene interpretation. The second section, refers to a map building method based on a self organizing map algorithm. The results and ideas are discussed in the remainder of this chapter.

8.1. Experimental Procedure

In this section, a comparison is made taking into consideration different methods to extract the most relevant information from a set of images, based on their global appearance. Three different descriptors were used, based on color histograms, Wavelet decomposition and Gabor filters. The desciptors were extracted directly from omnidirectional data, without un-warping the images. The study demonstrate how these descriptors affect the performance and the accuracy they offer, within a content based image retrieval context. Tests were conducted with the simulator that has been described in detail previously, in chapter 2.

The elaboration of these experimental procedures employed two robot exploration strategies. First, the robot was allowed to move freely in order to explore the largest possible surface. Second, the robot moved along a pre-specified closed trajectory. During both exploration strategies, the robot was collecting visual snapshots which then used to build a map based on the correlations between sensor signatures and actual robot positions. The robot receives sensor information only from a panoramic camera. The fundamental principle adopted was the following: While the robot navigates, continuously collects snapshots from the environment and for each snapshot color and texture descriptors are extracted. These perceptual signatures were then used to build the environment representation that may also serve as an example image description that it will then base its search upon.

A SOM algorithm was used as the main topology preservation mechanism since it is able to cluster high-dimensional data into a low discretized representation with, typically, two or three dimensions. This property make SOMs useful for exporting qualitative characteristics and for visualization purposes. A graphic representation of the steps followed during the experimental procedure can be seen in figure 8.1.

Goals

- find a useful internal scene representation
- let the robot build/learn the map itself

challenges

- Efficiency on handling multidimensional sensor information.
- Implementation and computation simplicity

8.2. Preprocessing

Image preprocessing encompass all these techniques required for enhancing data images prior to computational processing. In general preprocessing operations involve background noise suppression, intensity normalization, color conversion etc. The captured snapshots were in the RGB color space, in the BMP format with a size of 256 x 256 pixels.

8.2.1. Histogram equalization and filtering

In order to ensure image quality a histogram equalization technique was applied to all images to bring up better image detail and enhance colors. This method usually increase the global contrast of an image especially when an image has close contrast values. A histogram adjustment operation creates a better distribution of tonalities especially for low local contrast areas. Histogram equalization achieves this effect by expanding the most frequent intensity values of an image. Another important advantage is the fact that textures appear more vivid and with increased detail. In addition, a median filter, a nonlinear digital filtering technique that shows certain advantages compared to linear ones [Pitas et al., 1990], was applied to the images in order to suppress fine detail and preserve edge information. The resulting images were then fed to the feature extraction stage.

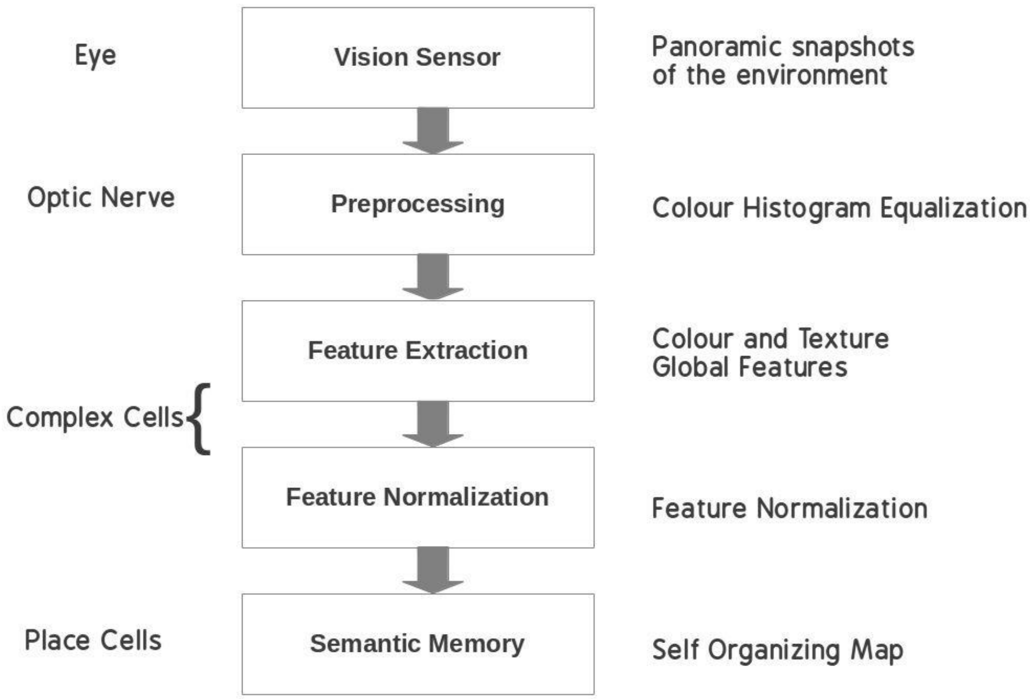


Figure 8.1. Steps of the bio-inspired procedure.

8.3. Feature Extraction

Sensor information coming from omnidirectional cameras is rich but extremely high dimensional. Thus a feature extraction mechanism is necessary to represent scene information in a more compact way. Raw input data must be transformed into a compact representation of a small number of global features. These feature extraction techniques must be carefully chosen so as to extract only the relevant information, with respect to the task being performed, instead of the enormously sized input. An essential advantage of omnidirectional vision sensors, when employed with global statistical features, is that no image de-warping from spherical to cylindrical coordinates (panorama) is required.

8.3.1 Color features

Images converted from the RGB color space to Lab color space and then are decomposed to L, a and b components. Since luminosity component L only refers to perception of lightness, it was excluded from the process. The histograms were then divided into 12 bins in an effort to coarsely represent the content and reduce dimensionality. A feature vector was then formed by concatenating the two channel histograms in one vector (figure 8.2).

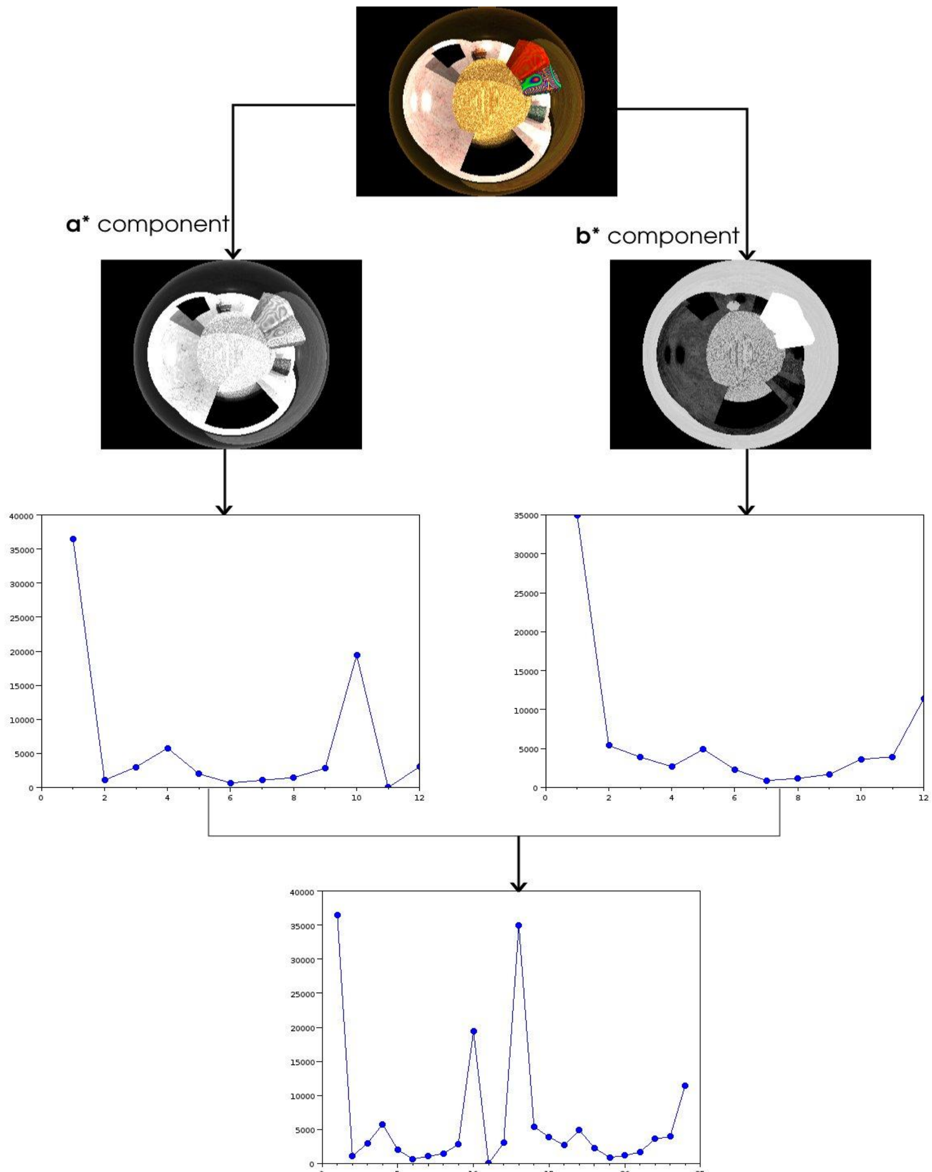


Figure 8.2 . Color feature extraction procedure. Images are decomposed to a* and b* channels. Both histograms extracted and concatenated in a single image signature.

8.3.2. Texture

This step was performed twice consecutively with a Gabor filter bank and a single stage Wavelet decomposition. For a comparison of different wavelet and Gabor transforms, for texture annotation, see [Ma et al., 1995]. The purpose of this approach is to determine whether it is possible to extract meaningful texture information directly from panoramic images. As stated before, these images presented low resolution, distorted textures. In order to apply these transformations it is necessary to convert the images to grayscale.

8.3.2.1 Gabor filter design

A Gabor filter samples the frequency space of an image providing information about oriented, band-pass, localised textures. Usually filters are reported with frequency bandwidth in octaves and orientation bandwidth in 30, 45 or 60 degrees. For this experiment three directions selected 0^0 ,60^0and120^0as recommended in [Clausi et al., 2000] and only one frequency with the formula suggested in [Zhang et al. 2002] and calculated as follows.

$$\mathrm{F}=0.25+\frac{2^{i-0.5}}{N_c}\,\mathit{for}\ \mathrm{i}=1$$

Where *Nc* is the size of the image, resulting on a value of F = 0.25552.

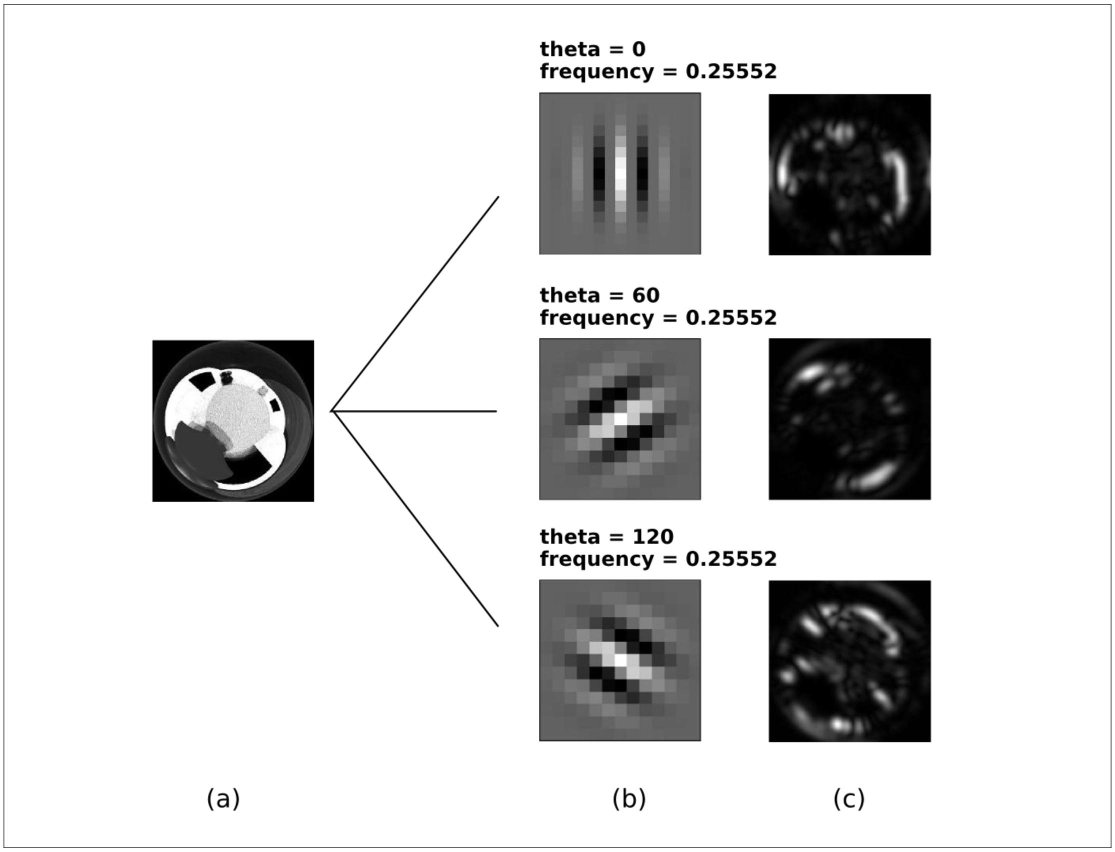


Figure 8.3. Output images (c) after the convolution between the Gabor filter kernels (b) and the input image (a) for three different orientations and one central frequency.

This bank of Gabor filters of different orientations and one central frequency convolved with the whole grayscale images. From these image responses (figure 8.3) the following five statistical moments calculated, to represent general texture descriptor, yielding for a feature vector with 15 elements. These statistical energies are commonly used to describe texture for classification and segmentation purposes [Howarth et al., 2006].

$$m = \sum_{i=1}^{L-1} z_i p(z_i) \qquad (8.1) \quad \textit{Average Intensity}$$

$$\sigma = \sqrt{\mu^2(z)} \qquad (8.2) \quad \textit{Average Contrast}$$

$$R = 1 - 1/(1+\sigma^2) \qquad (8.3) \quad \textit{Smoothness}$$

$$u = \sum_{i=0}^{L-1} (p)^3 \, (z_i) \qquad (8.4) \quad \textit{Uniformity}$$

$$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 \, p(z_i) \qquad (8.5) \quad \textit{Entropy}$$

8.3.2.2. Wavelet texture descriptors

For each snapshot a degradable filter bank was applied in order to decompose the image into a low resolution image and a series of detail images. For every detail image the afforementioned statistical attributes were calculated. It is noteworthy the fact that a wavelet decomposition procedure is computationally more effective compared to a Gabor filter bank implementation and convolution with target images. The wavelet, that image decomposition based upon, is the well known Haar Wavelet (figure 8.4). Again, the texture feature vector consist of 15 elements.

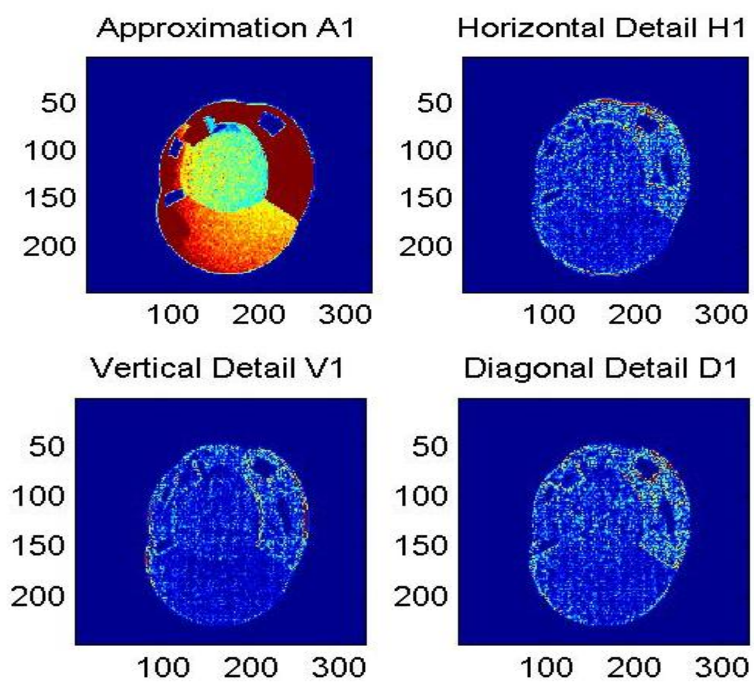


Figure 8.4. Output of the Haar wavelet image decomposition. Image A1 is the output of the low pass filter, and the H1, V1, D1 the Horizontal, Vertical and Diagonal detail images respectively.

8.4. Feature normalization

Scaling of variables is of special importance since the self-organizing algorithm use the euclidean metric to measure distances between vectors. If a variable moves between greater bounds than any another, then will completely dominate on the topological arrangement because of the superior impact on the distances measured. Typically, one would want the variables to be equally important. The standard way to achieve this is to scale all variables so that their variances are inside a predefined range. A linear scaling of the numerical variables were applied to all feature vectors, with data variance normalized to one, and values normalized between [0,1].

8.5. Feature performance

Good visual features are more stable than raw data under a variety of conditions such as illumination variance or different viewing angles. As have been mentioned before, rotating an omnidirectional camera results to raw data rotation in the opposite direction so the number of pixels seen by the camera remains constant. This implies the same color histogram from every angle. The downside of this invariance is that different images may result in similar features. Discriminant power may be altered by a combination of texture and color information, although, statistics applied globally to an image suffer from the same problem such as color histograms. A comparative study was carried out to evaluate the strength of the visual features, in terms of image retrieval performance. To test this approach these steps were followed:

- The robot placed in ten predefined positions, manually.
- For each test position the three feature vectors were extracted corresponding to each feature category, labeled and stored in a data base.
- For each test position, a number of 100 snapshots were acquired, in random positions, within an area of influence determined by a maximum allowed physical distance around each test position. Then associated feature vectors were extracted and stored in the data base.
- A range query was specified to retrieve all images up to a distance threshold. For each test position the threshold corresponded to the most physically distant random point in the current area of influence. The metric that was used to calculate feature vector distance is the standard euclidean distance and applied to all images in the data base.

From each image, three feature vectors were extracted and stored into three different categories. The first category contained the feature vectors with the color information, with a size of 24 elements. The other two categories contained the feature vectors that consisted of both color and texture information extracted from Gabor filters and Haar wavelet responses respectively (figure 8.5). The mixed feature vector had a size of 39 elements.

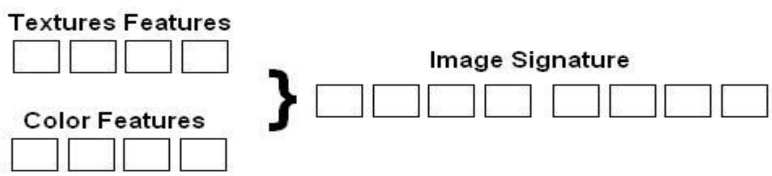


figure 8.5. Image signature as a combination of color and texture features.

The retrieval performance of the system can be measured in terms of it's recall and precision. Recall measures the ability to retrieve the correct image associated with every feature vector (figure 8.7).

$$Recall = A / C \quad (8.6)$$

Where A is the number of relevant retrieved images and C is the total number of images in the database. The precision in image retrieval can be defined as the measurement of the retrieved relevant images to the query of the total retrieved images (figure 8.6).

$$Precision = A / B \quad (8.7)$$

Where A is the number or relevant retrieved images and B is the total retrieved images. The precision and recall measure show the effectiveness of image retrieval with relevancy to the query and database. Hence, they can be combined to a single measure that describes the accuracy of image retrieval (figure 8.8).

$$F = \frac{2}{\frac{1}{r}+\frac{1}{p}} \quad (8.8)$$

Where r is the recall ability and p the precision in retrieving relevant images. Harmonic mean express a compromise between precision and recall and is high when both values are high.

As can be seen in figure, the combination of color and texture features provide better classification accuracy than color features alone. Storing both color and texture statistics is very common in CBIR systems [Datta et al., 2008] This increase in performance is in general true since adding channels of information can lead to perceptually distinguish between different positions of the environment. However, this depends on the visual information that is present on every scene and the distance metric that is used.

Combinations of color and texture features exhibited performance which was considered proximal (figure 8.8) without significant improvement from the color features alone. This may be attributed to the fact that the feature extraction methodologies that followed, were not optimized for selecting the best filters nor an adequate size of filter kernels was designed. Since overall performance of Gabor filter responses led to slightly better results, all other experimental procedures were performed with only these texture features.

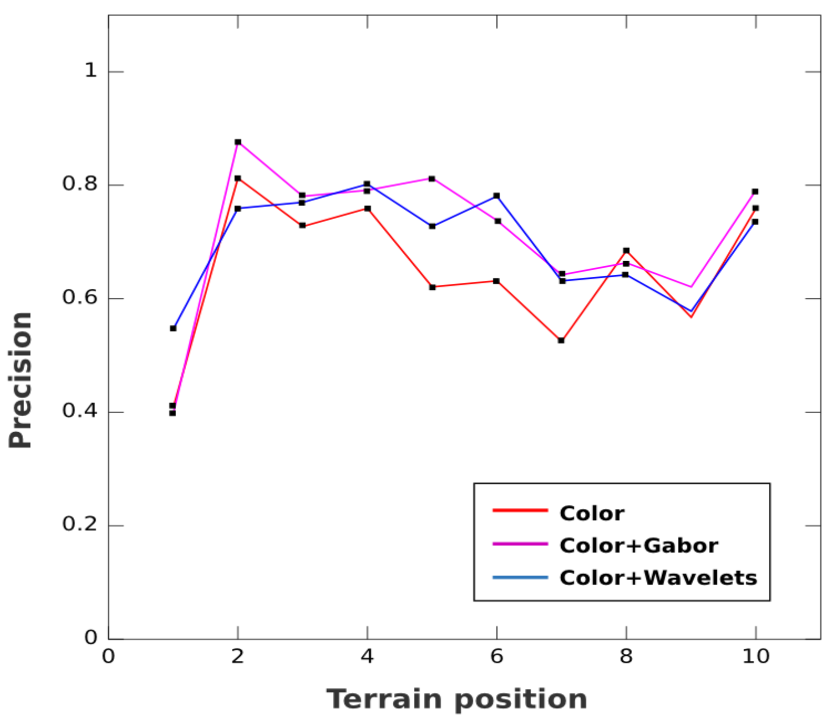


Figure 8.6. Precision in retrieving relevant images for every set of feature vectors.

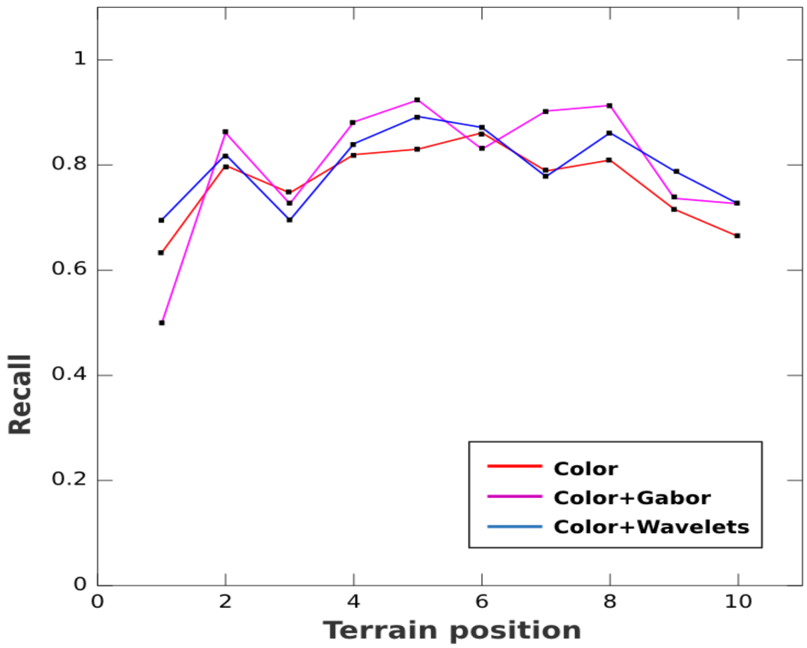


Figure 8.7. Recall ability to retrieve the correct image associated with every feature vector

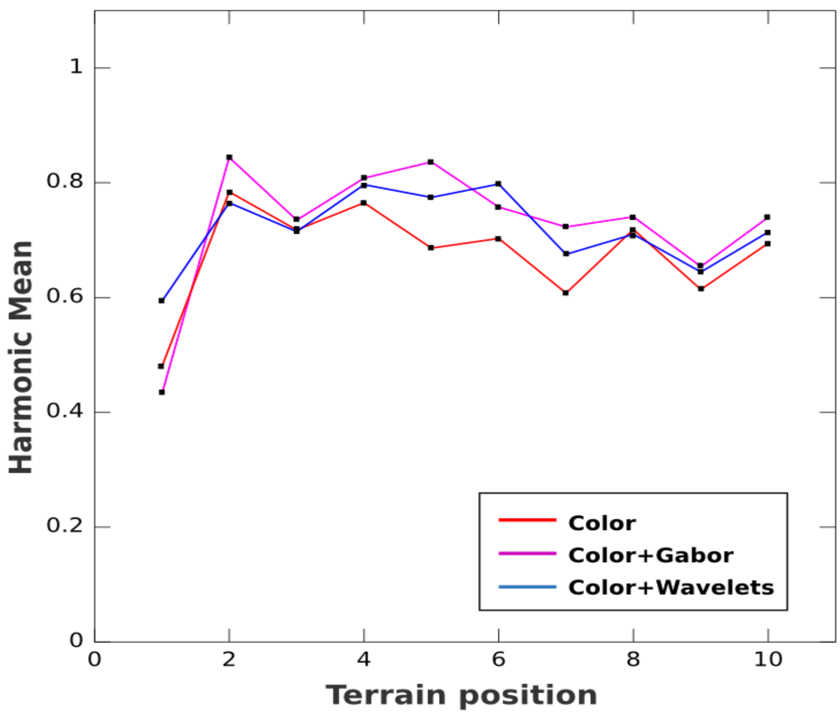


Figure 8.8. Harmonic mean express the accuracy of image retrieval with relevancy to the query and database.

8.6. Texture feature comparison

The procedures followed for texture feature extraction were not the optimal. Both filter banks have a small number of filters that could not describe with a great accuracy the textures being present in each image. Although, cooperatively with the color features, were able to minimize problems such as perceptual aliasing and assist on retrieval tasks. A potential problem is the low level of detail that omni-directional images provide. An un-warp from polar to rectangular coordinates may help to add discrimination robustness however requires more computational resources.

8.7 Self-organization of visual perceptions

In order to test the efficiency of the feature selection and extraction process, a SOM algorithm was trained for two sets of input data. For each set of data a robot performed a navigation scenario. First, the robot was allowed to explore the environment, autonomously, having as a key behavior to avoid obstacles. Navigation terminated after collecting 3000 image signatures. Second, the robot manually operated along a closed loop path in the same environment (figure 8.9). The dotted line in figure 8.7 exhibits the shape of the trajectory the robot manually traversed. At regular time intervals a new snapshot was obtained, for a total of 500 snapshots.

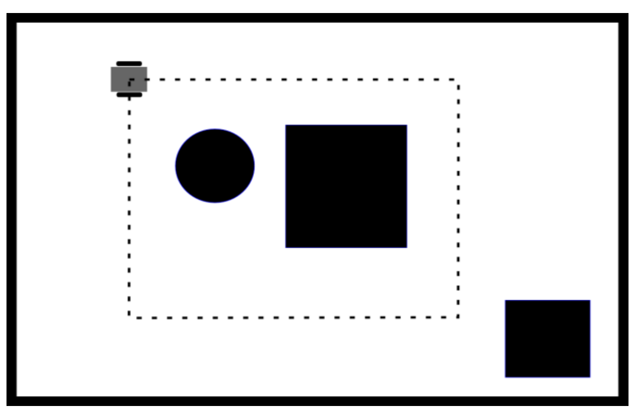


Figure 8.9. Plan of the simulated arena. The dotted line indicates the route that the robot followed during the second navigation scenario.

After the description of scenes to only a set of few features, in this specially prepared environment, a SOM algorithm applied on these set of visual perceptions to adapt a mobile agent to the surroundings without the need to define and model the relevant aspects of the environment.

The size of the network grid was 12x12 nodes as depicted in figure 8.8. The other learning parameters for the SOM algorithm were: Initial training radius = 2 and final training radius = 0.1 the neighborhood area of influence, and Gaussian, the neighborhood function of influence. Distance metric was the standard Euclidean.

The system proved capable of transforming the set of image signatures into a limited number of discrete perceptions that cover a small area of the environment and which could then be used for navigation purposes. The multitude of all these signatures quantize the input space in discrete regions with neighboring signatures mutually similar (figure 8.10).

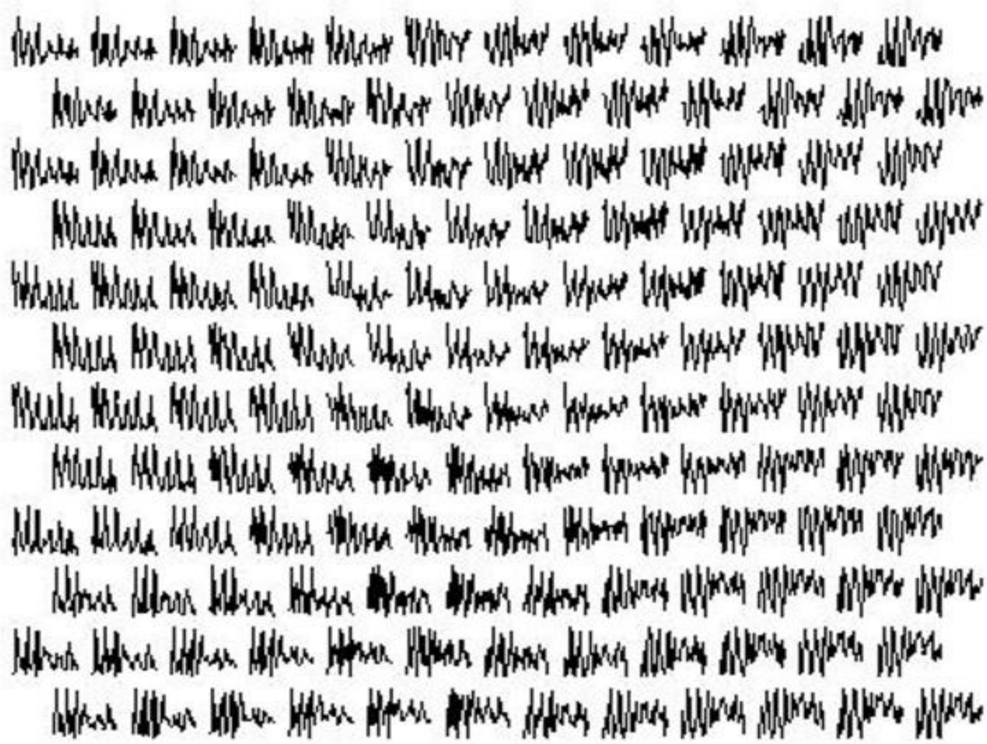


Figure 8.10. SOM codebook vectors. Each element in the grid corresponds to an image vector signature, with neighboring signatures mutually similar.

8.8. Data Labeling

In order to evaluate an information retrieval system, such as a SOM algorithm a labeling method is required to assist the user in understanding the data collection that was presented by the map. The SOM was labeled as follows. For each image find the best matching node. If this node is preoccupied, replace with the current image if this is closer to the node than the previous one. Then, for each best matching unit, the associated image replace the code vector in the visualization map. Labeling was done automatically and some units in the map may thereup remain unlabeled. The labeled graphical representations of the training results dictated that similar images were in adjacent nodes on the topological map.

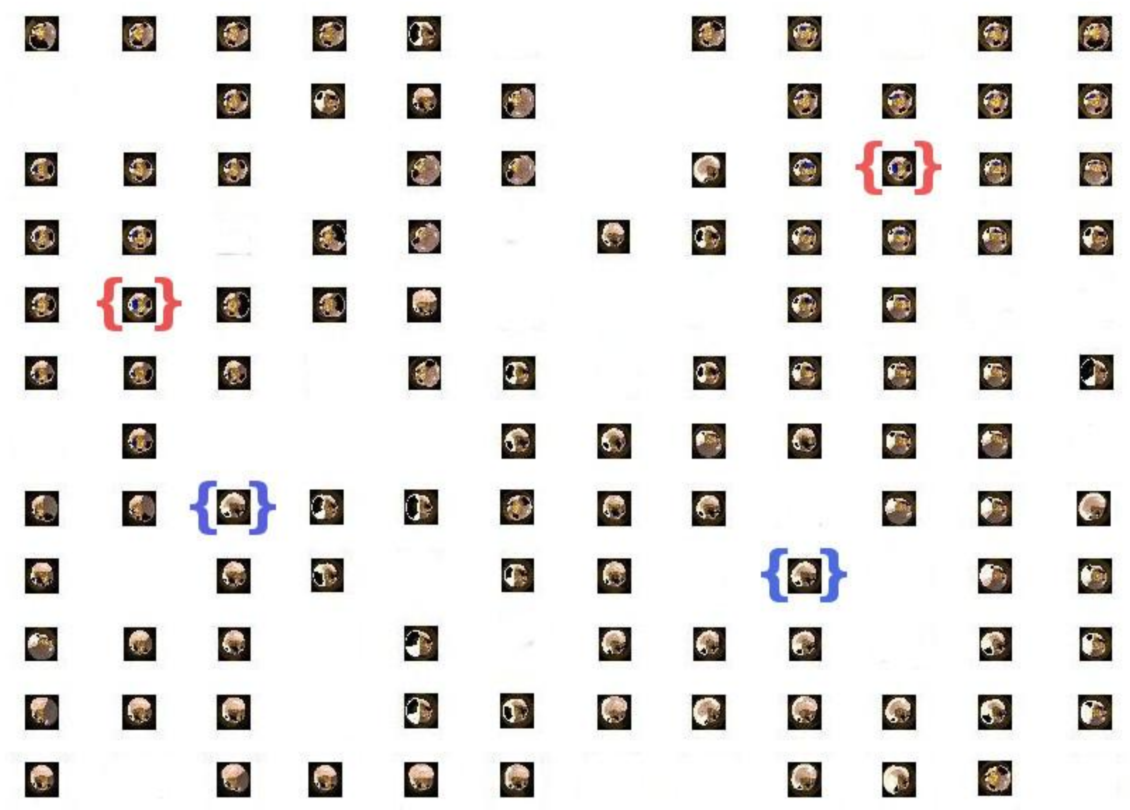


Figure 8.11. SOM similarity map resulted from the autonomous exploration scenario. Snapshots in curly brackets correspond to different places that perceived as the same. Some nodes were not assigned to images due to the fact that no images found to be closer to these nodes than any other node.

If best matching units are associated to each node, then SOM can be also

seen as a similarity graph. Some of the nodes may appear unlabeled as can be seen in figure 8.11 due to the fact that no images found to be closer to these nodes than any other node. This could also be explained by the presence of obstacles in the environment of the robot. Since SOM scheme is equivalent to a projection from multi-dimensional data (24 or 39 vector elements) to only two dimensions, information may lost in the process. Due to the nature of the global statistics methods that have been used, some snapshots appear in more than one location in the map (figure 8.11). In these cases, the robot encountered perceptual aliasing problems, which means that two or more different places perceived as the same. Moreover, some snapshots do not correspond to the areas that have been located.

As can be seen, in the case of the closed pathway experiment, map building appears to be satisfactory, but only a few nodes contributed to the procedure (figure 8.12). This can be attributed to the fact that the SOM rectangular lattice structure is fixed, and therefore, was a trade-off between continuity and resolution of the mapping.

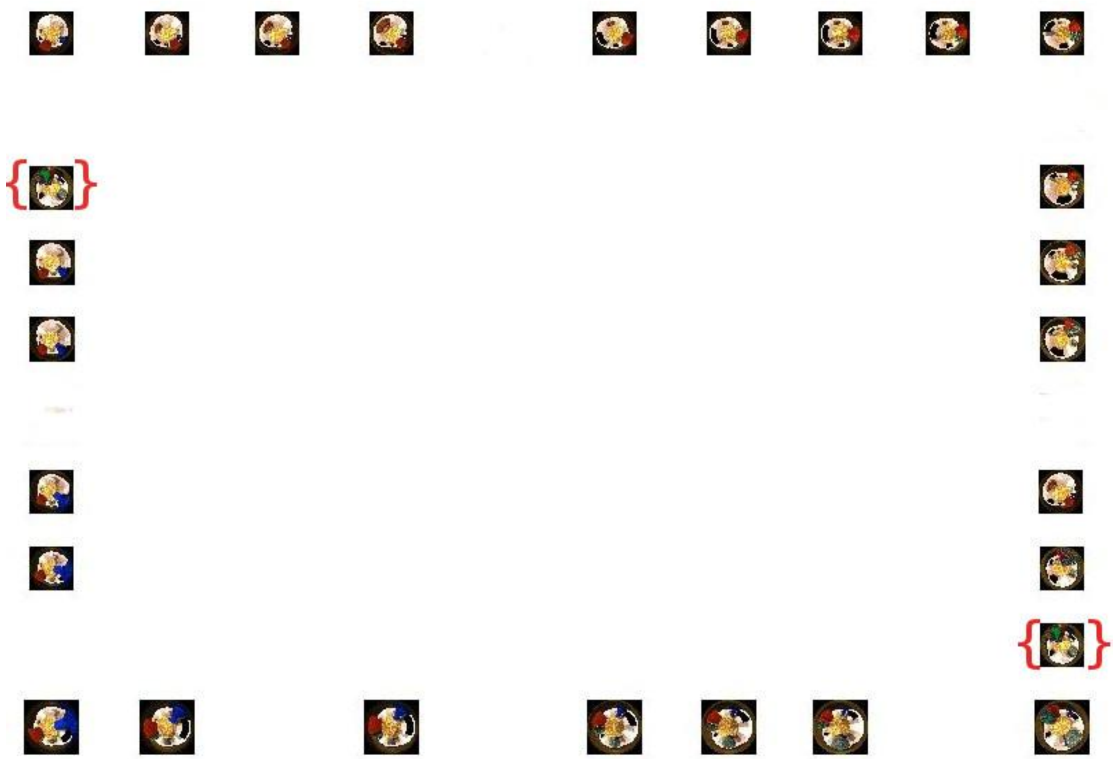


Figure 8.12. SOM similarity map for a closed loop trajectory.

8.9. Results

An appearance based neurocomputational architecture that emulates hippocampal place cells learning has been validated on a simulated mobile robot, and the results demonstrated the ability to efficiently map the surrounding environment. Three global measures of image similarity have been compared for use in topological mapping with a catadioptric camera as the only sensor. Experimental results with images acquired from a simulated scene indicated that a simulated agent could map the environment from a set of training images. The approach appeared satisfactory on processing spatial information without identifying special landmarks, for both exploration scenarios. The methodology, that make use of only visual cues about the environment, was able to generate “concept patterns” where a robot could search upon to identify it's position. Depending on the relationship between input and output space of a SOM, some information of the topological arrangement may be lost in the process. Since input dimension is higher than the dimension of the map grid, a representation mismatch can be detected between input and output spaces.

9. Case study: Robot Navigation

This chapter examines the way in which a robot might use the sub-symbolic representations of an environment that have emerged through self-organization. These representations are extremely difficult for a human to interpret, especially for multi-dimensional manifolds. Using both proprioceptive and exteroceptive information, through an evolutionary strategy, it was possible to build a robot controller that use this map for navigation purposes. In order to reach target positions, the robot move in a way that try to minimize the difference between the current perceived image and an image that corresponds to a goal position.

9.1. Introduction

With respect to vision based robot navigation, most research work is focused on four major areas: map building and interpretation; self-localization; path planning; and obstacle-avoidance. Of these four major research areas, self-localization is of key importance. The recognition of the initial position, the target position, and the current position occupied by the robot are all bound to a self-localization process. This chapter, describe a combination of a developmental method for autonomous map building and an evolutionary strategy to verify the results of the map interpretation in terms of navigation usability.

The strategy involves two discrete phases: map building and navigation phase. In the first phase an agent freely explores a pre-determined simulated terrain, collecting visual signatures corresponding to positions in the environment. After the exploration, a self organizing algorithm builds a graph representation of the environment with nodes corresponding to known places and edges to known pathways.

During the second phase, a population of robot controllers is evolved to evaluate map usability. Robots evolve to autonomously navigate from an initial position to a goal position. In order to facilitate successful translation, a shortest path algorithm is employed to extract the best path for the robot to follow. This algorithm also reveals all those intermediate positions that the robot needs to traverse in order to reach the goal position. These intermediate positions act also as sub-goals for the evolution process.

9.2. Sensing the Environment

To be fully autonomous, a robot must rely on its own perceptions to localize. Perception of the world generates representation concepts, topological or geometrical, within a mental framework relating new concepts to pre-existing ones [Ascani et al., 2008]. The space of possible perceptions available to the robot for carrying out this task may be divided into two categories: Internal perception (proprioception) or perceptions of its own interactions with the world, associate changes of primitive actuator behavior like motor states; external or sensory perception (exteroception) is sensing things of the outside world. A robot's exteroceptors include all kinds of sensors such as proximity detectors and video cameras. The system uses only visual information for map building and navigation.

9.3. Bio – Inspired Robot Navigation

The source of inspiration for this method comes from the animal kingdom. Small animals, such as insects, navigate through natural environments seemingly with little effort. For example, despite their relatively simple nervous system (and hence limited memory capacity), bees and desert ants are able to retrace their movements. Such a level of efficiency indicates flexible representations of the surroundings based on visual cues taken from target locations such as home and food sources [Collett et al., 1998]. These representations seem to have an appearance based flavor rather than a Cartesian arrangement of landmarks. To visit target locations after prior exploration, insects traverse in a way that reduce discrepancies between the stored snapshot and their current retinal image [Cartwright et al, 1982].

As stated already, the main drawback of appearance-based methods is that localization is only possible in previously mapped areas. Like landmark based mechanisms, appearance based navigation systems suffer from the problem of perceptual aliasing [Siagian et al., 2009], the situation that different locations produce identical sensory perceptions. A possible solution could be the incorporation of temporal or odometry information to resolve any conflicts. Another possible solution is to divide the goal into a set of sub goals of smaller tasks easier to fulfill. Such an approach, even if perceptual aliasing is present, is more efficient since subtasks are easier to manage and achieve.

9.4. Environment Representation

The most natural representation of a robot's environment is a map. In addition to representing places in an environment, a map may include other information, such as properties of objects, regions that are unsafe or difficult to traverse, together with information of prior experience. An internal representation of space can be used by a robot to pre-plan and pre-execute tasks that may be performed later.

A topological map is one which captures the connectivity of the environment and has been simplified so that only vital information remains and unnecessary detail has been removed. The simplicity of topological maps support much more efficient planning than metric maps [Butz et al., 2010],[Thrun, 2002].

The key to a topological relationship is based on an abstraction of the environment in terms of connectivity between discrete regions or objects, with edges connecting them. In the simplest form, this may involve a complete absence of metric data. A robot employing this representation has no real understanding of the geometric relationship between locations in the environment but the enclosed information is sufficient for the robot to conduct point to point motion. The use of graphs has been exploited in many robotic systems to represent spaces. The following example [10 franz, 1998] is representative.

A graph is a kind of abstract data structure that consists of points or nodes connected by links, called lines or edges. Each node corresponds to one of the unique landmarks and each edge corresponds to known paths between them. If the environment consists of networks of corridors and rooms (as found in many indoor environments, such as office buildings or hospitals), it is less complex to specify the topology of important locations and their connection suffice.

Such representations present some advantages difficult to ignore. First and foremost the computational and memory cost is relatively low. The path planning in metric maps can be computationally very expensive; unlike the lightweight planning nature of graph based structures. Second, they do not require accurate determination of robot's position and therefore are less sensitive to error accumulation, commonly occurring in metric mapping approaches. Topological visual navigation is usually based on key-frame matching to self-localize and navigate to a previously visited location [Booij et al., 2007],[Wang et al., 2009].

9.5. Map-building Phase

9.5.1. Terrain Exploration

The proposed approach considers robots to be like insects, equipped with simple control mechanisms tuned to their environments. Therefore, a model of terrain exploration using a simple two dimensional Brownian random walk was used. Such an approach could mimic the navigation behavior of simple animals and microorganisms such as insects. The advantage of this approach is minimization of simulation artifacts such as cyclic behavior. During this step the robot collects panoramic snapshots at regular time intervals.

9.5.2. Visual Feature Extraction

This collection of panoramic images represents a large amount of raw data and therefore it is necessary to extract some specific features that describe the content of each image. Image analysis based on color information is robust for robot map-building and image retrieval problems and, due to their statistical nature, provide a complete rotationally invariant representation when employed with panoramic cameras (figure 9.1). Moreover, they are also computationally cheap to implement.

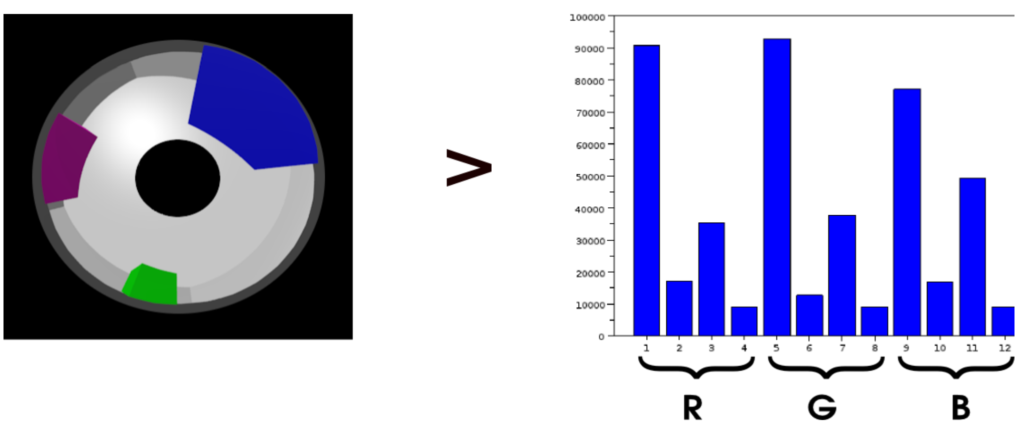


Figure 9.1. Omnidirectional snapshot and extracted RGB histograms.

The set of color signatures, extracted during terrain exploration, can be manipulated as a large abstract image database. This is the foundational scheme of a content based image retrieval approach. Self localization can be based on a measure of resemblance between the currently acquired image of the robot and the base of images stored as perceptual signatures representing familiar terrain. To measure color histogram similarity, the standard Euclidean formula have been used. This distance metric is a comparison between the identical bins in the respective histograms and all bins contribute equally to the distance [Chang et al., 1996]. The Euclidean distance between two color histograms h and g is given by

$$d_E(h,g) = \sum_{m=0}^{M-1}(h[m] - g[m])^2 \qquad (9.1)$$

9.5.3. Self-Organization of Visual Signatures

There are many reasons to use a self organizing system for robot mapping, preferred over other mechanisms that have no plasticity properties [Nehmzow, 2000]. The first reason is that less parameters, which describes the robot operation, need to be predetermined. Information given by sensors incorporate noise, leading to erroneous conclusions regarding spatial perception. Information may be contradictory when sensor readings come from different sensors but represents the same robot position. Data clustering addresses the problem of noise and handles meaningless information.

Growing Neural Gas is a network that can learn the topological relationships from an input set of vectors using a variation of the Hebbian rule. GNG dynamically add or remove nodes and can approximate the input space with higher accuracy compared to a network with predefined structure such as the Kohonen self organizing feature map [Kohonen, 1982]. Assuming that a given distribution of points is represented by a container shape, the algorithm will begin to create freely moving particles which will try to expand uniformly to fill the input space. After convergence is reached, the network nodes then represent the shape of the container.

9.6. The Navigation Phase

9.6.1. Path Planning

Prior to navigation, path planning is an important issue as it directs the robot on how to get from an initial position to a goal position. Since the environment is stationary with no other moving obstacles, the process of path planning is straightforward. Topologically, this problem is equivalent to the shortest path problem of finding a route between two nodes in the graph. Many algorithms have been developed to find a path in a graph. For example, Dijkstra's algorithm [Dijkstra, 1959] computes the optimal path between a single source point to any other point in a graph (figure 9.2). Since we compute the path once after the mapping phase, a real time algorithm is not necessary.

9.6.2. Self-localisation

The robot continuously keeps track of the current location. While the robot moves, collects snapshots and exports corresponding color histograms. Every newly acquired histogram is being compared with every stored histogram in the graph structure. The robot self-localizes when the closest histogram on the topological map is found. Each of the nodes in the graph represents a specific histogram and the closest one indicates the current position of the robot on the map.

9.6.3. Visual Navigation

For the robot to conduct point to point navigation, a controller is necessary that will move the robot through a set of intermediate points towards the final position. The proposed robot behavior controller realizes an Elman neural network (Elman NN) and a genetic algorithm (GA). Neural network architectures are particularly well suited for complex pattern classification tasks and genetic algorithms are good optimization procedures because they can explore large and multidimensional spaces to find global solutions. Hence, they are well suited for training neural networks [Palamas et al., 2013].

The neural controller is composed of a grid of input neurons whose activations are given by the color bins of the corresponding histograms. Two output neurons control the angular torque applied to the left and right wheel of the robot. A set of neurons with recurrent connections fed from hidden and output neuron layer, help to learn past instances and correlate them with new information. The input neurons of the neural network are activated by sensory data, and the output neurons control the motors of the robot. Within a population, each individual has a different genome describing a different neural network (different weight vectors), thus resulting in specific individual responses to sensory-motor interactions with the environment. These behavioral differences affect the robot's fitness, which is defined, by the number of successive milestones traversed by the robot.

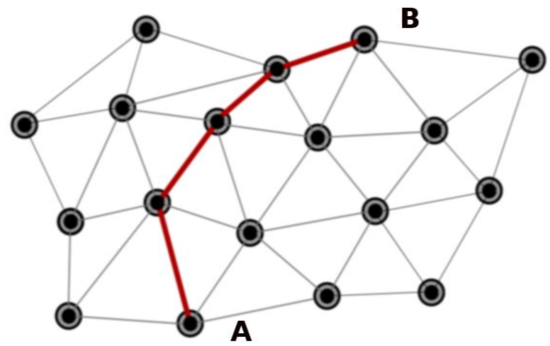


Figure. 9.2. Disjkstra's graph search algorithm output.

Evolutionary strategies require that a large population of individuals be evaluated over the course of many generations. In the case of evolutionary robotics it has been assumed that it would take far too long to do all of these evaluations in the real world. The main practice is to evaluate in simulation, whether partial or in whole. The aim of this evolutionary strategy is to create a population of agents with different genomes, each defining a set of parameters of the control system of the robot. The genome is this set of parameters whose translation into a phenotype, the actual behavior of the controller, can cause the system to depict biological behaviors. The artificial genome decodes the weight values associated to synaptic connections of an artificial neural network that determines the global visual navigation behavior.

9.7. Neural Network Controller

The neural network that is used (figure 9.3) is a typical feed-forward architecture with evolvable thresholds and discrete-time, fully-recurrent connections at the output layer [Floreano et al., 2005]. This type of neural network is used to do sequence processing, especially when these sequences are made of indexed data [Elman, 1990]. The processing occurs in steps and it is assumed that neuron outputs are computed instantaneously. A set of twelve input neurons receive information about the color distribution from the images captured from the panoramic camera.

Each neuron covers a band of the color variations in the image that is a bin value is assigned to each input. Each of the RGB color components of the image are divided into four bands. The activation of each neuron is scaled in the interval [0, 1] so that activation 0.5 corresponds to zero torque applied in the wheels. Activation values above and below 0.5 stands for forward and backward rotational speeds, respectively. The two output neurons act also as proprioceptive information about the speed of each wheel. A set of short term memory units stores the values of the output neurons at the previous sensor-motor state and sends them back to the output units through a set of recurrent connections [Floreano et al., 2005]. All other neurons in the hidden layer have recurrent connections to store previous activity.

$$f(x) = 1/(1 + \exp(-x)) \qquad (9.2)$$

Neurons use the sigmoid activation function in the range [0,1], where *x* is the weighted sum of all inputs (equation 2). For each discrete time interval they encode both the sensorial information and the motor commands passed to the wheels.

9.8. Evolving Controllers

Algorithms in Evolutionary Robotics (ER) frequently operate on populations of candidate controllers, initially selected from some random initial population of controllers. This population is then repeatedly modified according to a fitness function, a particular type of objective function that is used to indicate the closeness of a given design solution to achieving a set of aims.

Evolutionary Robotics builds upon several aspects of artificial evolution. The Genetics aspect is about what goes into the artificial chromosomes and how these chromosomes are mapped into individuals. Genetic encoding and genotype-phenotype mappings are the key to the evolvability of a system. In our case the genotype represents the architecture of a controller in a form of a binary string and the phenotype represents the possible solution space. The population of robot controllers is also referred to as genomes.

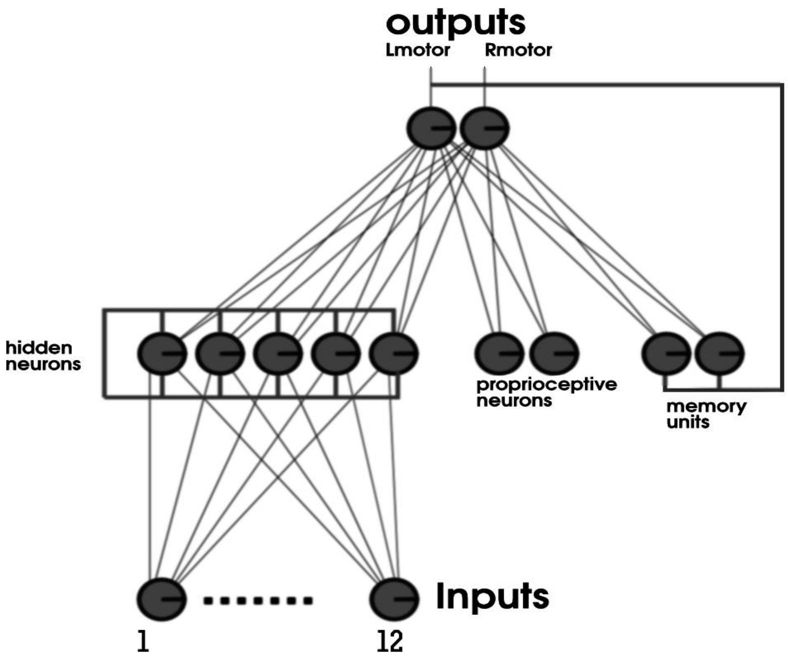


Figure. 9.3. Discrete-time recurrent neural network

Evolutionary algorithms have been widely used to design cognitive architectures for robots with emergent behaviors (see [Nolfi, 2002],[Harvey et al., 2005] for an overview). The main strength is their ability to cope well with high complexity problems using only a high-level reward function. Best candidates are rewarded only for their global efficiency because of the impossibility of foreseeing every sub-goal the robot has to solve. If the global objective is very hard then initial performance may be so poor that the evolutionary process is hard to initiate. Another problem is local minima in which the evolutionary process may become trapped. A fitness function must be simplistic yet descriptive enough for targeting specific goals. Designing a fitness function is essential to the successful use of a genetic algorithm. If the fitness function is poorly designed, the algorithm will either converge on an inappropriate solution, or will have difficulties in converging at all.

For a successful incremental evolution process the system requires an accurate knowledge of the problem to be solved so as to lead the evolutionary algorithm to perfect convergence. For graph based robot navigation the global task can be divided into smaller tasks. Both global task and subtasks are self-similar, i.e. the goal is to transfer the robot from one point to another. Since in our case the different sub-tasks are in nature exact copies of the main task, by just dividing the path that the robot needs to traverse, the only requirement is to determine when to switch from one sub-task to another. Fitness function is an objective function used as a metric to calculate the distance of each individual from a set of goals.

The success of evolutionary algorithms depends on the fitness function design. A good function design must guarantee that a collection of solutions exists, differentiating enough, with values that changes neither too rapidly nor too slowly with the given parameters of the optimization problem. The fitness function was designed to select robots for their ability to arrive at the goal zone. The neural network has a set of evolvable connections that are individually encoded in the genome. A population of 100 individuals is randomly initialized and each individual genome is decoded into the synapses of the neural network. The twenty percent of the population with the highest values are used for reproduction and the rest discarded. The new genomes have a crossover value of 0.1 per pair and mutation probability of 0.01. The meaning of crossover is swapping a pair of genetic strings around a randomly chosen point. Mutation consists of toggling the value of a random bit in the genetic sequence. The best two genomes from the previous generation are inserted to the current generation, unaltered, to improve the stability of the process. This strategy is known as elitist selection.

9.8.1. The Evolution Process

The fitness function was designed to select the best robots to arrive at the goal node and is described as follows. The fitness value is the percentage of the distance the robot covered between two adjacent nodes in the path. Every time the robot reaches a node in the node sequence, as extracted from the path planning phase, it is rewarded with a value of 1. Since it is extremely difficult for the robot to match the current perceived histogram with the target node, the assumption was made that 90% of the covered distance corresponds to successful goal reaching.

The robot must traverse the nodes in the specific order as dictated from the outcome of the Dijkstra's algorithm. If the robot arrives at a goal node that is not successive in order, the robot is not awarded for this sub-goal. Successful individuals have to arrive at all sub-goal nodes through this specific order. The running fitness value for every agent in the population is the summation of extra value gained for each successive step plus the current percentage of the distance between currently arrived at node and the next one in the sequence.

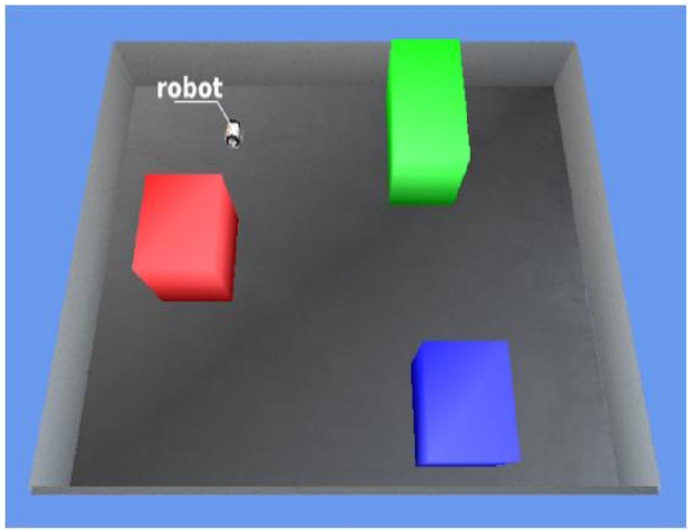


Figure 9.4. The simulated environment and the robot used in our experiments

9.9.Experimental Setup

9.9.1. The Robot

The simulated robot can be seen in figure 9.4. The omnidirectional camera is widely used in visual based robot navigation and localization, which is due to the large field of view. Images are obtained by placing a convex mirror a short distance from a camera. The main advantage that led us to promote this solution is the large field of view compared to orthographic or standard cameras. The system provides a 360^o view of the robot's environment around the vertical axis when it is mounted on top of the robot. Landmarks are always in the field of view except for occasional occlusion and therefore have increased reliability. This is advantageous when utilizing topological representations as the more information the image contains the more stable it is. Another advantage is the orientation independency when employed with statistical methods such as color histograms.

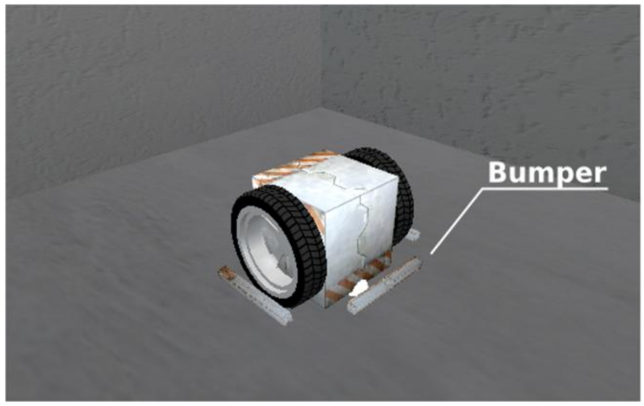


Figure 9.5. Simulated differential drive robot. The robot has four bumpers to detect collisions with walls and obstacles.

The robot is cubical in shape with two independent drive wheels attached in the middle of the chassis and two trailing casters, front and rear. This is a typical differential drive setup and the robot can change its direction by varying only the relative rotation speed of its wheels and hence does not require an additional steering mechanism. The robot is equipped with two, one bit, horizontal axis bumper bars. The purpose of the tactile sensors, when a reaction to a collision occurs, is to reposition the next individual to initial conditions and start a new simulation trial.

9.9.2. The Environment

For the experiments a simple 3D world is used, a closed rectangular arena with colored obstacles, dark gray walls and no ceiling (figure 9.5). This environment is not as visually complex as a typical real life environment. The primary goal was to demonstrate the plausibility of evolving agents that could use cognitive maps and behaviors based on visual information which otherwise would be very difficult if not impossible to employ.

9.9.3. Experimental Procedure

Initially a robot is allowed to freely navigate in the environment in order to build a collection of 500 panoramic snapshots of the environment from different perspective views. A GNG algorithm, performing off-line, formed a grid with topological relations between these visual cues. The grid starts with only two nodes and grows until the criterion of 20 nodes is met. Based on this grid, a shortest path is extracted to indicate optimal route from a starting position to a global target position in the arena. A genetic algorithm evolved a neuro-controller to allow a robot to successively follow the six nodes the optimal path consists of. Each individual robot tested for a period of time lasting for 10 seconds or 1000 simulation cycles. Trials were truncated earlier if collisions detected from the bumpers.

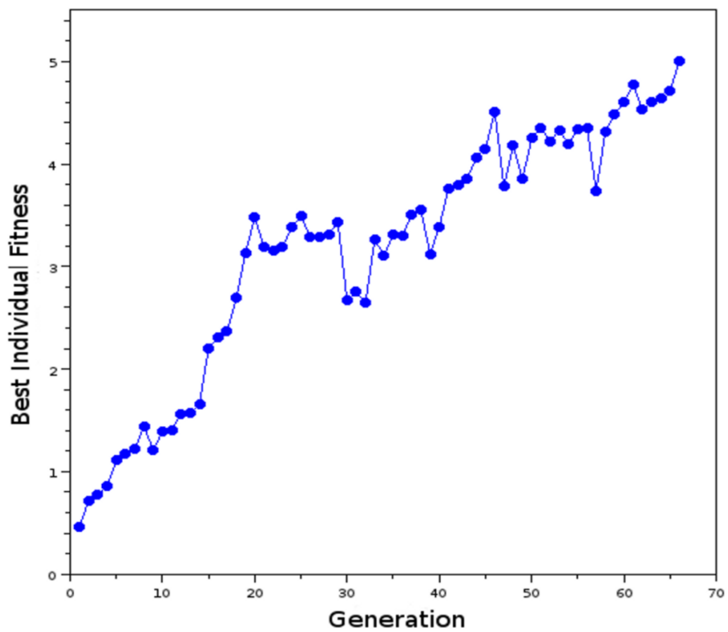


Figure 9.6. Best individual fitness value from each generation.

9.10. Results

This section shows experimental results of the proposed method. Several sets of experiments were performed with varying parameters relating with the GNG algorithm, the neural network architecture and the genetic algorithm. Something worth mentioning is the fact that the dark shades of the environment gave better results than other colorations. This is simple to explain since black color interprets as absence of color and does not interfere with the three other landmarks, the discrete nature of which is being enhanced.

Figure 9.6 depicts a record of the best individual score for each generation to evaluate the solution domain. As can be seen, a navigation controller evolved after 66 generations. The robot that used this controller, managed to pass through all the intermediate points until the final objective. The path followed by the robot is shown in Figure 9.7. The gray and red points correspond to intermediate sub-goals and final goal respectively. The optimal path planning computed with Dijkstra's algorithm between an initial and final position in the graph that was generated by the GNG algorithm.

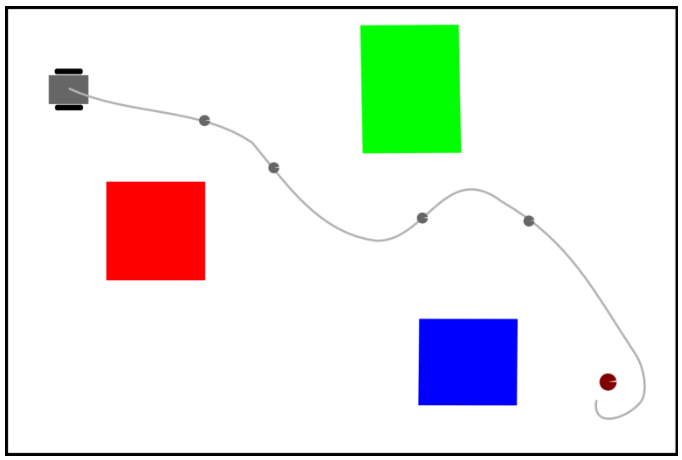


Figure 9.7. The robot successfully followed the sequence of nodes. The small gray dots are the positions that the robot encountered the threshold of 90% of the distance covered between two adjacent nodes. (A successful controller is always awarded with a fitness value of 5). The difference between the actual position and the robot position is due to the error in the calculation of the best matching unit and the 90% accuracy threshold.

9.11. Summary

This chapter explores the advantages of evolutionary sub-goal robot navigation with a cognitive map architecture. All methods used have been tested using a simulated environment. The GNG algorithm has been previously shown to be effective in forming topological maps through an appearance based framework. Evolutionary strategies have also been applied successfully in solving complex problems such as visual navigation. However these algorithms may take some time to converge to an optimal solution. Feature selection is a particularly important step for building robust learning models. The method based on global only image properties and may suffer from the problem of perceptual aliasing [Angeli etal.,2008], the fact that different physical locations correspond to similar sensory perceptions.

However the purpose of this study was to demonstrate the efficiency of simple algorithms to solve complex systems. After verification of the aforementioned algorithms using simulations, these need to be evaluated on actual robots and modify as necessary to ensure acceptable real life robot navigation.

10. Online Topology Preservation

This chapter aims to explore a rehearsal mechanism as a countermeasure to prevent catastrophic forgetting in unsupervised learning connectionist networks. A comparative study have been carried out to evaluate the effectiveness of applying the learning procedure to the three well known algorithms, SOM, NG and GNG. Both advantages and disadvantages are highlighted in terms of performance and reliability.

10.1. Semantic-Temporal Memory Representation

Unsupervised learning networks are attractive for topology mapping tasks since they are able to discover hidden structure in unlabeled data. In practice, as dictated from experimentation work on chapter 8, successful application of topology preservation to a particular problem requires the presence of the whole data set. Implausible forgetting have been identified as one of the main problems on neural networks. Temporal sequence learning is arguably more important than static pattern learning in real world problems. Training a network incrementally from just one or a small set of input data is highly desirable for online robot mapping. As have been stated before [Dayoub et al., 2008], due to the dynamic nature of real world environments, a robot need to continuously update it's internal representation. In their work presented a method for creating an adaptive map for appearance-based localization using long-term and short-term memory concepts.

As have been proposed by [Ans et al., 2000] the basic principle, in order to avoid catastrophic forgetting, is to learn new external patterns interleaved with internally generated 'pseudo-patterns'. The basis of a self refreshing mechanism can be analyzed as follows. Initially a network learns a given distribution. After learning is complete, a short-term memory stores the weight vectors as the codebook of the input data. The short-term memory represents what the network has learned so far. Finally, the short-term memory elements are being concatenated, with a new input distribution, into a new training set. The system then learns both input distributions

[Rousset et al., 2004], although the previous one may be represented more sparsely than before (figure 10.1).

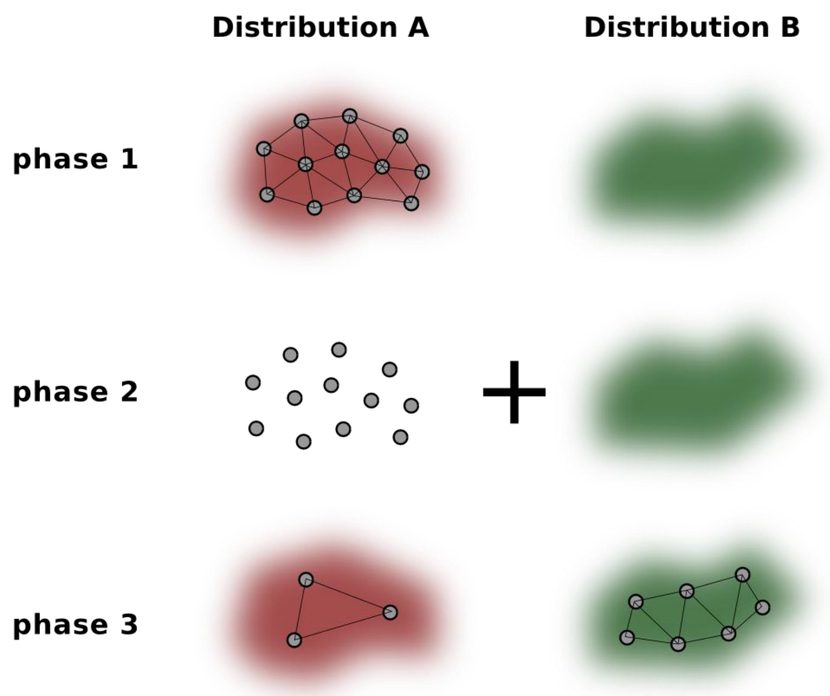


Figure 10.1. Rehearsal learning procedure. During phase 1 a network is trained with a distribution A. During phase 2, the sparsely learned representation of distribution A is being merged with distribution B. During phase 3, both distributions are being learned. The procedure could be viewed as a neurobiologically plausible model for long term memory consolidation [Ans et al., 2004].

In the remainder of this chapter, the aim is to provide an understanding of the properties of a single network with a self-refreshing mechanism which learns sequentially the content of a memory buffer. This simple buffer acts as a sort term memory which stores an episode that consists of a small sequence of *t* events. Then the content of this knowledge buffer is transferred to the target network by means of pseudo-patterns.

10.2. Pseudo-rehearsal architecture

The proposed architecture consisted of a short and a long term memory structures. Applying a pseudo-rehearsal technique to a self organizing network is straightforward since weight vectors and connections are always accessible. Apart from memory requirements for the aforementioned algorithms a new buffer was necessary to temporarily store all previous weight vectors. This buffer, which updates on every learning cycle, reflects the abstract history of events learned so far. When the network needs to learn a new pattern, the network would be trained on the new pattern and the set of previously stored weight vectors that reflect the history of events so far. That way, new patterns are interleaved with patterns that, even though they were not the originally learned patterns, nonetheless reflected the original function learned. An algorithmic schematic of the technique can be seen in figure 10.2.

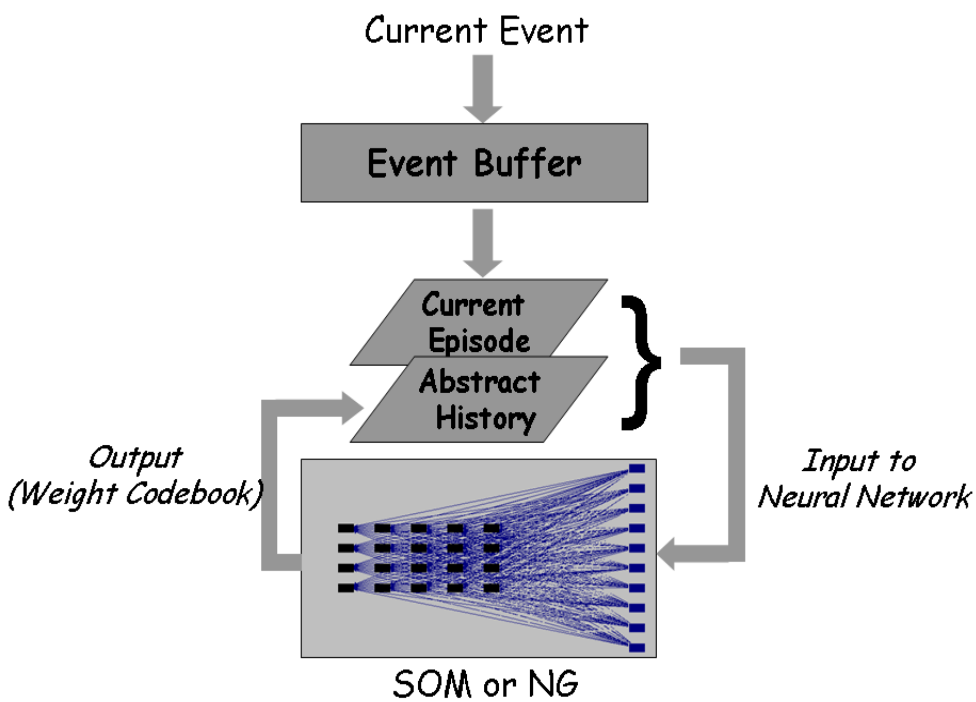
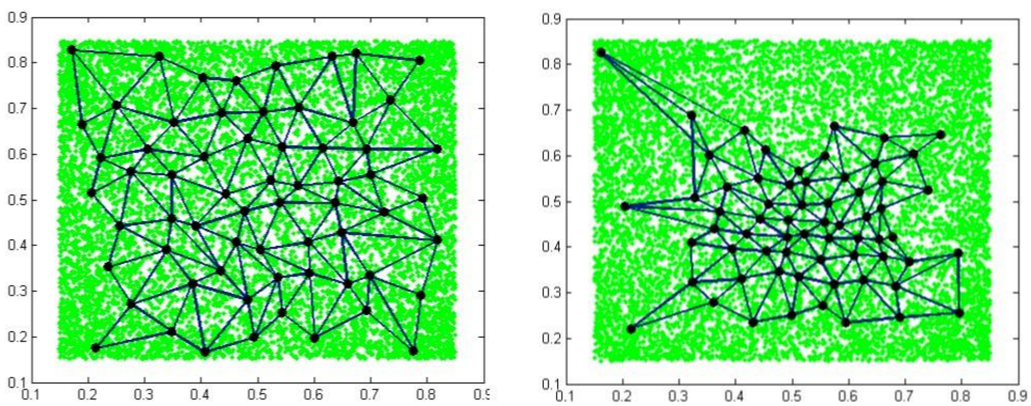


Figure 10.2. Self-refreshing mechanism with event buffer.

10.3. Online Learning Experiments

A set of experiments have been conducted in order to explore the performance for different input distributions and event buffer sizes. First, a SOM used to learn a 2D random distribution with different sizes for the event buffer. The results of the learning procedure can be viewed in figure 10.3. As can be seen, when the buffer size increased from one event to five events, topology and density preservations were more accurate. The experiment has been repeated with a NG algorithm and the results were slighty better, in terms of expandability. For comparison purposes, a standard NG algorithm trained with the same distribution (figure 10.4). In that case, the only learned point was the last input where all nodes have been focused (figure 10.4).

A notable problem with pseudo-rehearsal SOM was the challenging search for the optimal learning parameters, size and structure of the grid. The parameters of the self refreshing NG for the 2D random distribution defined as: n =64 the number of neurons, epochs = 1000 for the number of training epochs or the training steps, α_0 = 0.5 initial step size and λ_0 = n/2 the initial decay constant. Learning parameters for the SOM network defined as follows: Two dimensional rectangle shaped lattice of size 8x8, initial training radius = 2, final training radius = 0.1 were the neighborhood areas of influence, and Gaussian was the neighborhood function of influence. For all experiments, training continued until adaptation parameters, which decreased according to fixed schedules, reached the predefined values.



a. buffer size =5 b. buffer size =1

Figure 10.3. Pseudo-Rehearsal SOM results for two different sizes of short-term memory buffers

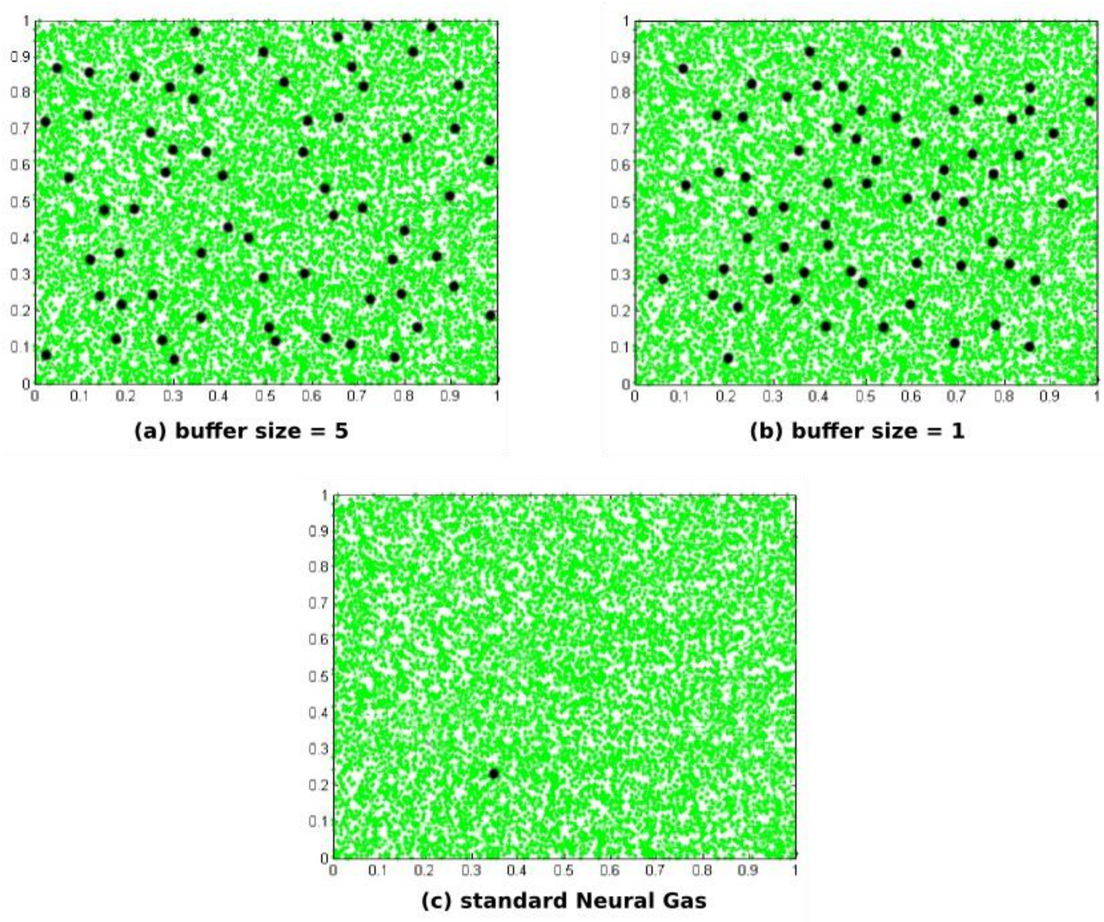


Figure 10.4. pseudo-rehearsal NG versus standard NG. Image (a) shows learning results with a buffer size = 5. Image (b) shows learning results with buffer size = 1. Image (c) the standard NG only learned the last input vector.

10.3.1 Learning a ring shaped distribution

To better understand the properties of the self-refreshing mechanism, a series of experiments were performed with a two dimensional ring shaped distribution that contained 1000 points. As can be seen in (figure 10.5) the performance of an one dimensional SOM proved to be quite satisfactory. Temporal information recorded well and the spatial representation was accurate. As it was expected, nearby neurons appeared adjust in time and space. This is true since an one dimensional SOM has the freedom to rearrange their nodes to catch the input space with higher topographic precision than an equivalent two dimensional (figure sinus).

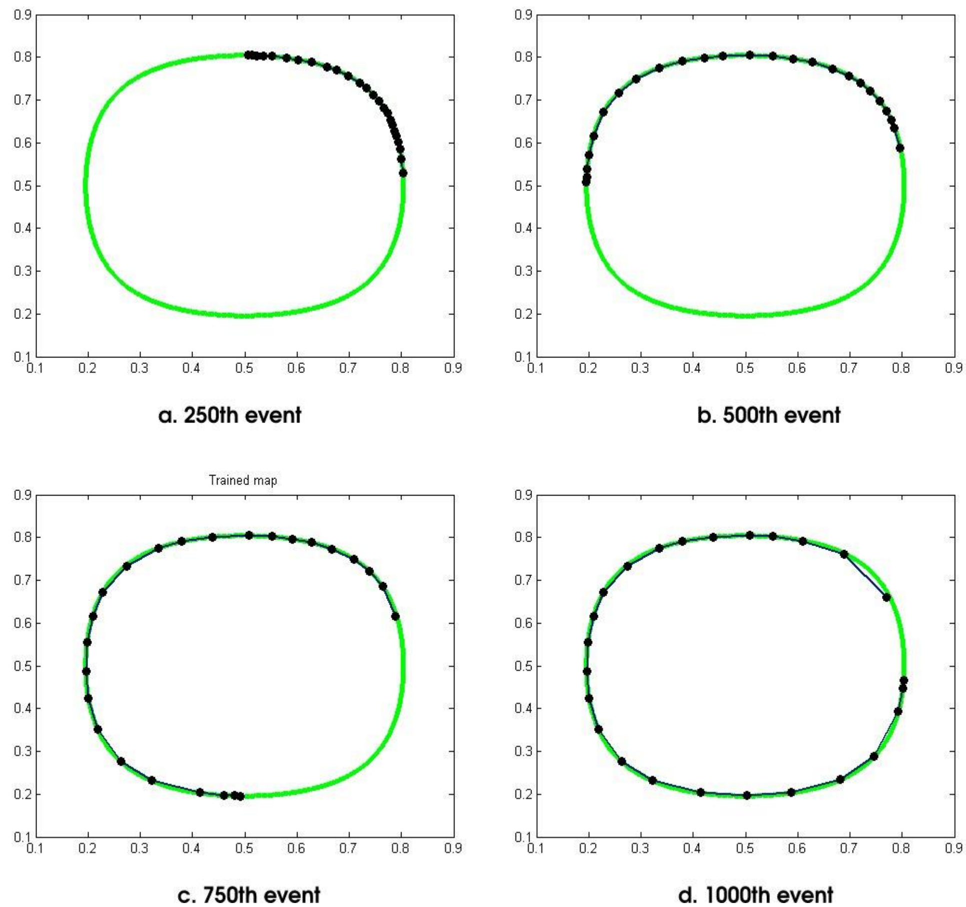


Figure 10.5. Episodic learning for self-refreshing SOM with buffer size = 10. the results show good spatial representation.

The procedure repeated for the NG algorithm, for the same ring shaped distribution, and the online learning capabilities of the scheme were reflected in figure 10.6. As can be seen, the performance of self-refreshing NG on capturing the input space was not as effective as the SOM counterpart. The nodes progressively moved off the trajectory, defined by the distribution, with a tendency towards the center of this. This was ascribed to the fact that, whenever a new input vector was applied, the algorithm adapted all nodes towards this specific input, unlike SOM where only the nodes between a neighbor area from the best matching unit, are being affected for each time step. For common reference, the same parameters have been applied to both SOM and NG, as in the previous series of experiments, except that the number of neurons was set to 25 for NG while the structure of the SOM network was one dimensional with a size 1x25. The buffer size determined to be 10 element wide.

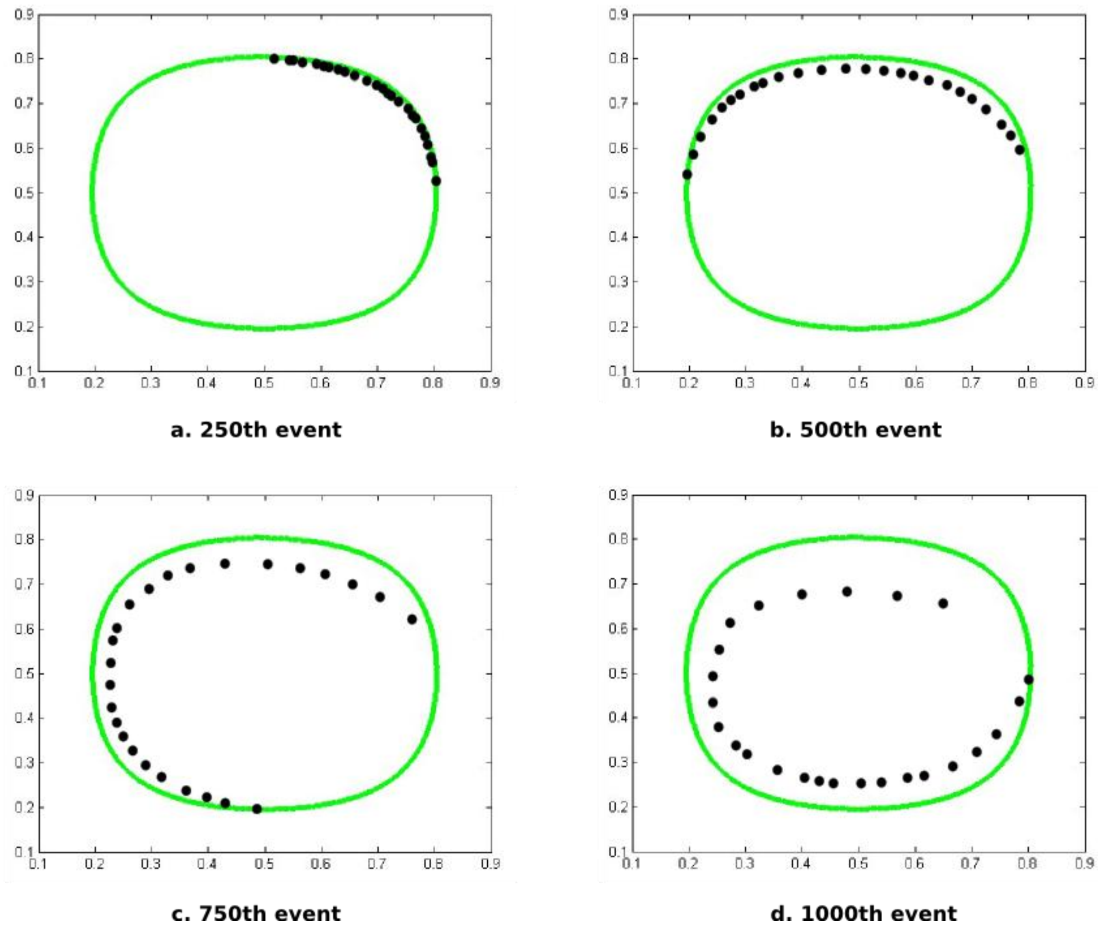


Figure 10.6. Episodic learning for self-refreshing NG with buffer size = 10. The nodes progressively moved off the track defined by the distribution.

Another variation of the last series of experiments have been conducted for only the one dimensional SOM, with the ring shaped distribution convoluted with noise, and the results showed good accuracy on learning the average trajectory and filtering out noisy fluctuations (figure 10.7). It is worth noting that in order to achieve good convergence rate and stable models, both networks, especially the SOM, may require fine tuning of the learning parameters.

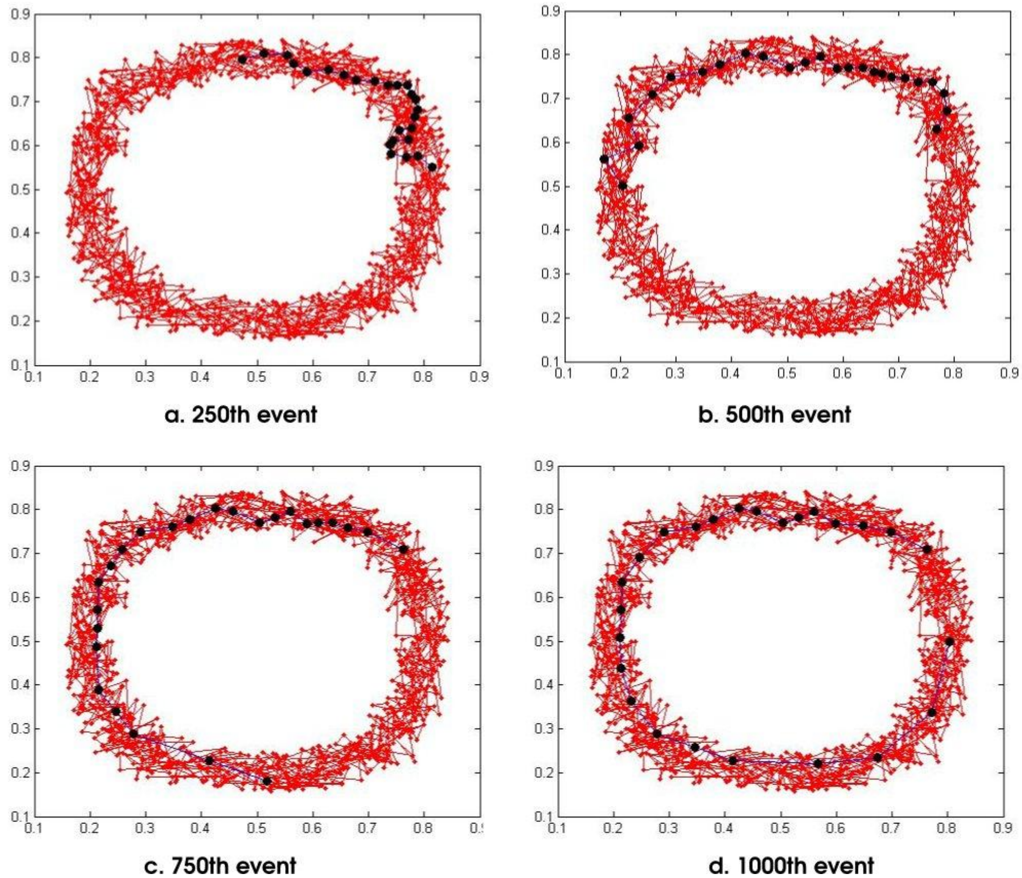


Figure 10.7. One dimensional self-refreshing SOM learning a ring shaped distribution convoluted with noise.

Considering optimal dimensions of a SOM network, and particularly in the case of a two dimensional SOM, satisfactory results have been obtained only for convex shaped distributions without holes. Learning a trajectory as depicted in (figure 10.8) may lead the network to learn both the trajectory and the enclosed area. Although, the quantization error indicated good results, many nodes were located outside of the distribution.

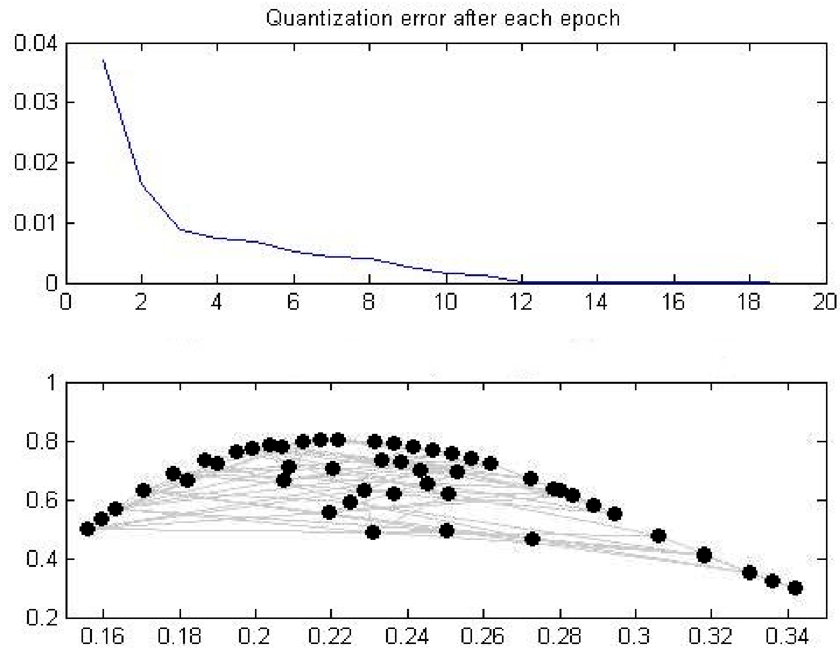


Figure 10.8. A 2D SOM learning a sinus like wave signal. The network captured both the signal and the enclosed surface defined by the signal line. It also appeared warped with irrelevant nearby nodes.

10.4. Learning a robot trajectory

With an aim to study the performance of the self-refreshing method applied to a robot mapping process, a set of experiments were conducted with a simulated robot that followed a pathway within an unknown environment. The robot, equipped with an omnidirectional camera, allowed to follow the pathway, during which images obtained at regular space intervals using relative odometry. All the images have a size of 256x256 pixels. In order to extract comparative results, the robot built the data-base offline and the same content used for all environment mapping scenarios. For image content representation, color visual descriptors have been extracted in the form of color histograms. The images were transformed from RGB to Lab color space and the histograms determined to span along 12 bins for each color component a* and b*. A feature vector was then formed by concatenating the two channel histograms into one vector with a size of 24 values. All values normalized by dividing each bin value by the total number of pixels in the image.

Three systems designed to simulate sequential learning, one for every neural network. Because SOM grid dimensions must be pre-assigned, the self-refreshing learning model applied to only one dimensional SOM grid since this outperformed every other grid structure on learning a curve. For both the NG and GNG algorithms, it is not a prerequisite to determine the grid dimensions since NG does not form a grid and GNG dynamically change it's grid formation during training. To compare the performance of every algorithm, the equivalent standard algorithms have also been trained with the same data distribution. For commencing sequential learning of SOM while adaptation, it is necessary to use small neighbouring area of influence combined with an adequate number of nodes. Training takes place in episodes which consists of five events each (buffer size). Training of GNG continues until the predefined number of 25 nodes is reached.

10.5. Results

The performance of self-refreshing SOM and NG , proved to be effective for online topology preservation when applied to the given input distribution. However, a potential problem in practical applications may be to determine *a priori* the number of nodes that are required for an effective application. Depending on the complexity of the distribution to be modeled, different node numbers may be appropriate. Both SOM and NG algorithms, requires this number to be known in advance. The GNG algorithm successively add new nodes, starting from a network with only two nodes, by evaluating local statistical measures extracted during previous learning steps. However, for comparison purposes, the maximum number of GNG nodes considered to be 25. Since, for both feature and codebook vectors is difficult to visualize their multidimensional data directly, only the first element of each color component and the corresponding codebook vector elements have been used to depict graphical representations of results. The simulation results demonstrated the general behavior of the three models.

10.5.1 Self-Refreshing Self Organizing Map

The self-refreshing SOM adapted sequentially to input space for the given scenario (figure 10.9), although, some forgetting of the first signals was obvious. Gradual forgetting is not a totally undesirable effect. Forgetting can be interpreted as making space for new knowledge. In the case of a robot moving in a dynamic environment, where allocation of objects takes place and may interrupt free space, unlearning parts of the map is highly desirable in order to allocate nodes into new unexplored areas. Learning parameters for the SOM network defined as follows: One dimensional with size 1x8, initial training radius = 2, final training radius = 0.1 the neighborhood area of influence, and Gaussian the neighborhood function of influence.

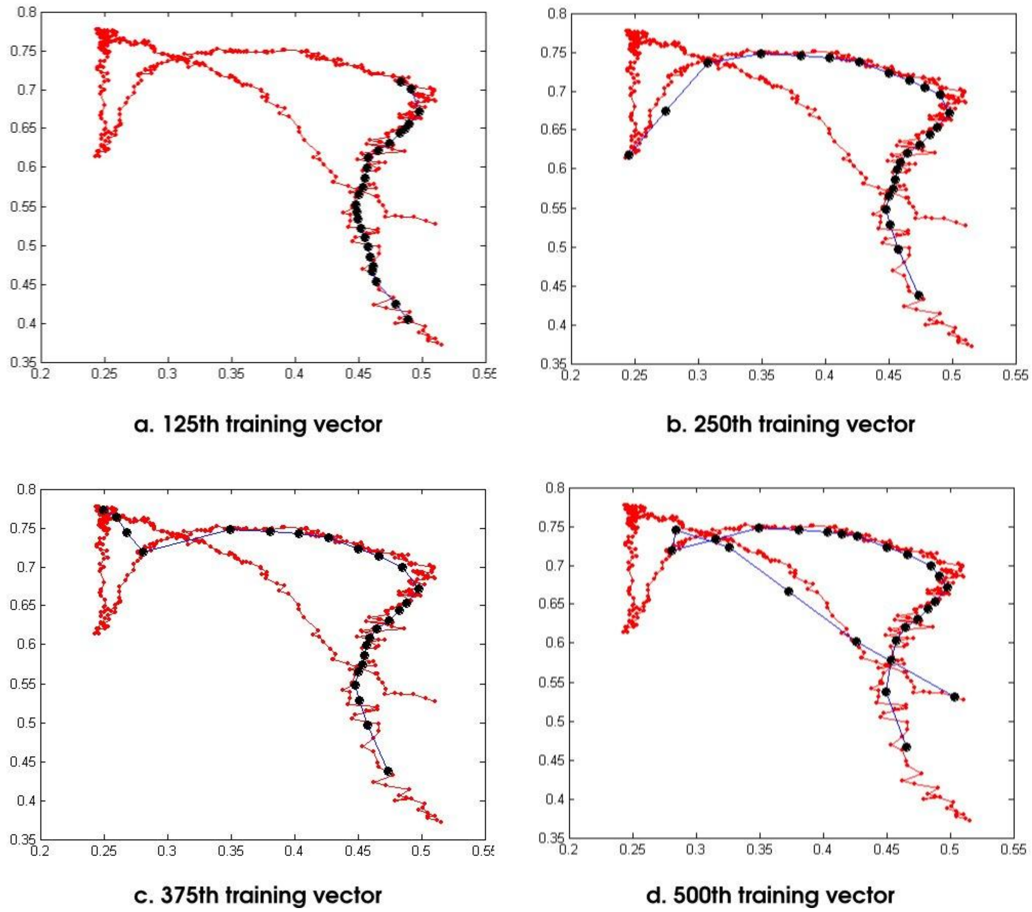


figure 10.9. Training an one-dimensional self-refreshing SOM on a robot trajectory.

10.5.2. Self-Refreshing Neural Gas

The NG algorithm also performed well but some node shifting out of the pathway was obvious (figure 10.10). The reason for this is explained by the global influence of each input signal on all the nodes. It appears that some forgetting took place, although the effectiveness of the model could be rated as similar to that of the self-refreshing SOM model. The parameters of the self refreshing NG defined as follows: n =25, the number of neurons, epochs = 1000 the number of training epochs , $\alpha_0 = 0.5$ the initial step size and $\lambda_0 = n/2$ the initial decay constant.

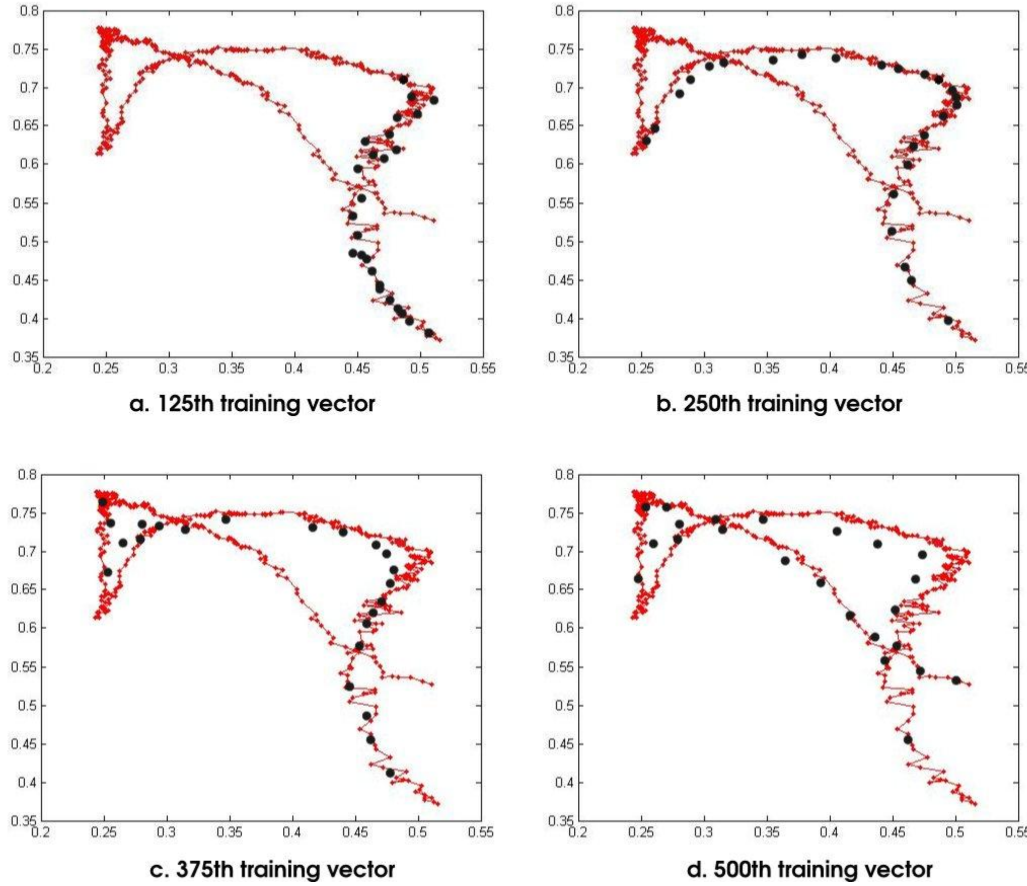


Figure 10.10. Tracking of the sequential learning process of NG.

10.5.3. Self-Refreshing Growing Neural Gas

In the case of GNG many discontinuities have been noted on the network (figure 10.11). Since the algorithm tries to model the input space, unaffected nodes from input signals cause them to delete their edges. Although, it is able to build a topology without discontinuities [Palamas et al., 2006]. this could be possible only by setting extreme values to the age variables of each edge. The parameters that used were: lambda = 100 which determines how often to add a new vertex, max-edge-age = 400, alpha = 0.05, beta = 0.0005 and maximum number of nodes = 25.

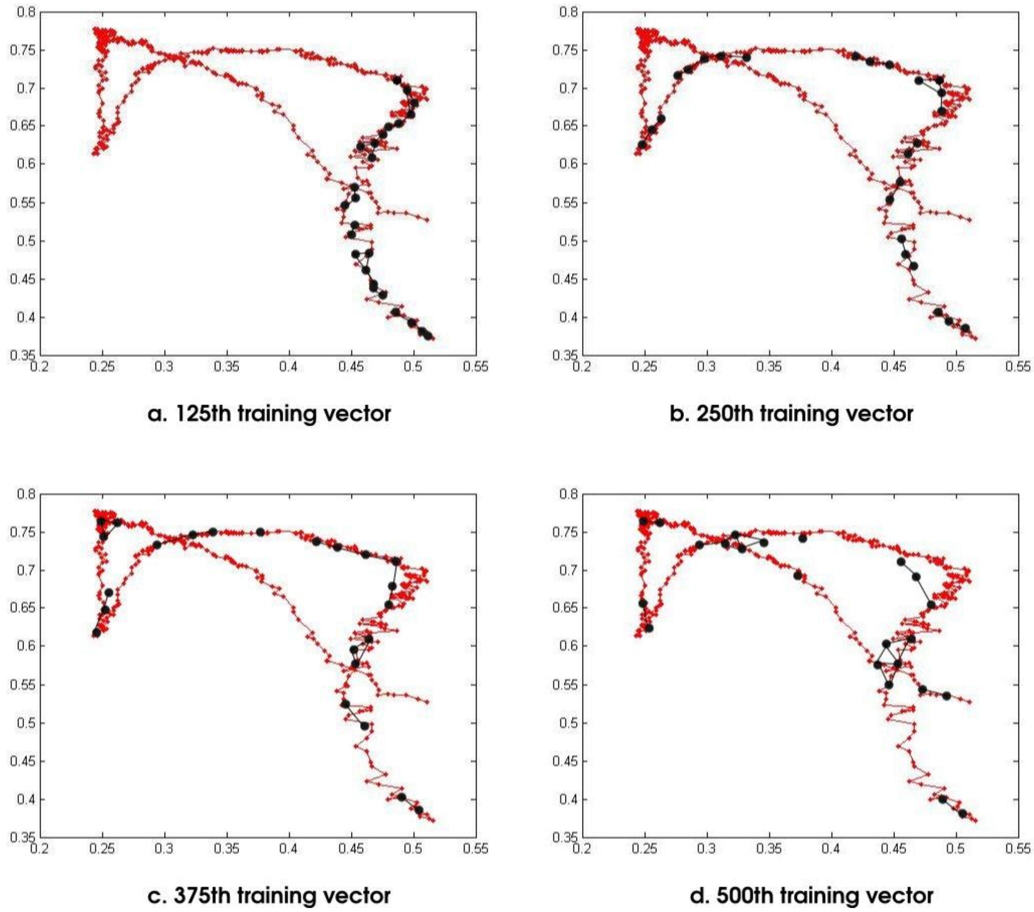


Figure 10.11. Tracking of the sequential learning process for GNG.

10.5.4. A Closer look on Self-Refreshing Model

A closer look on the results, for the case of a self-refreshing SOM, reveals the potential to model the average trajectory and filter out occasional fluctuations (figure 10.12). In real mobile robot situations, fluctuations that may be caused by faulty perceptions are common, caused by electronic or sensor noise, or artefacts such as reflections and illuminations when vision sensors are employed. During learning of a new location a faulty perception would lead to a ‘ghost’ place in memory, i.e., a node in the graph with no corresponding place in reality. When online learning is employed, occasional perceptions are being correlated with ‘history’ of events which may lead to memory systems less prone to 'ghost' perceptions.

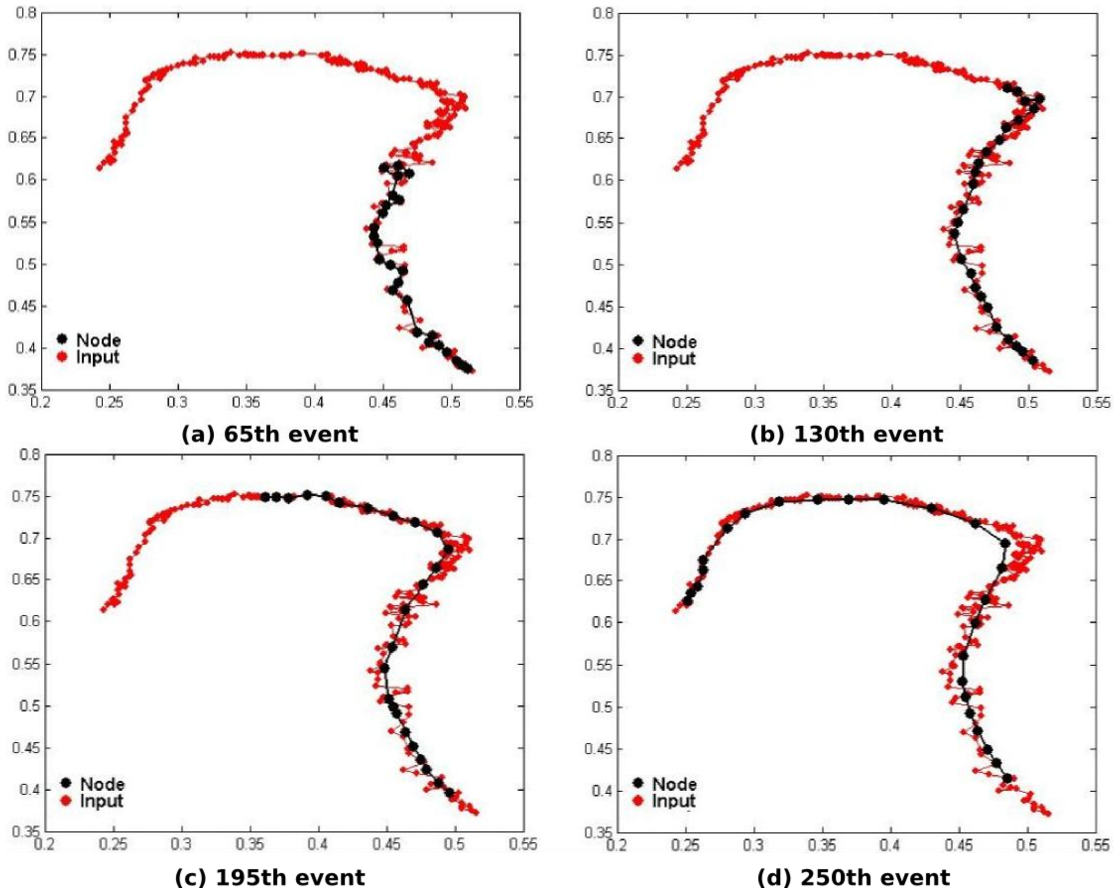


Figure 10.12. Closer look of the learning procedure.

10.5.5. Standard Algorithm Performance

In the case of standard SOM, NG and GNG algorithms, as it was expected, newly learned information completely destroyed previous learned information (figure 10.13). Since every algorithm has to learn only an input vector at a time, all corresponding nodes are being focused on the current input vector.

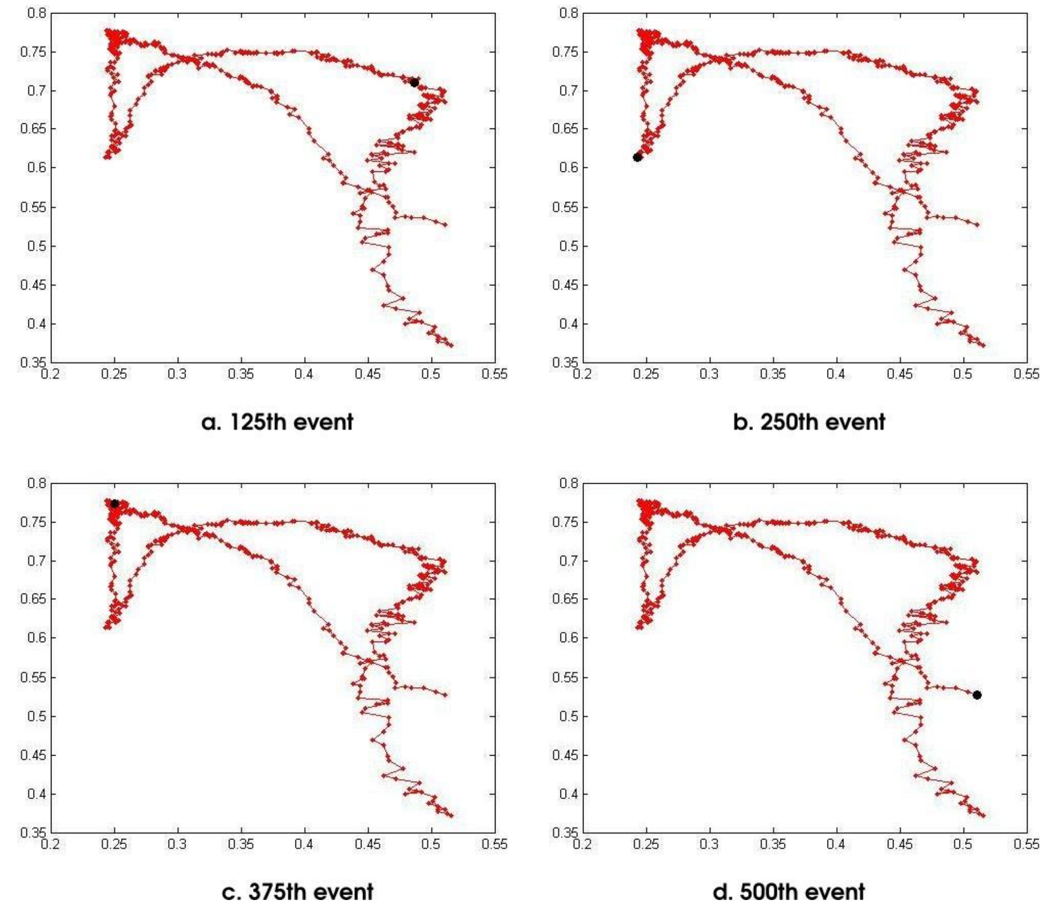


Figure 10.13. The original algorithms suffer from catastrophic forgetting. All nodes were being concentrated on every current input signal.

10.5.6. performance evaluation

The issue of SOM topographic quality is a complicated one. Depending on the relationship between input and output space, some information of the topological arrangement may be lost in the process. If the dimensions of the data set is higher than the dimensions of the map grid, a representation mismatch may be detected between input and output spaces. The topology preservation error then is additively accumulated in the sequential learning process. On the other hand, it is well known that NG and GNG adapts almost perfect to input manifolds. All three variants of self-refreshing learning appeared capable of overcoming the so called plasticity–elasticity dilemma. The effectiveness of the density matching property of the model can be seen on all results. A related question is, how topology-preserving can be measured from the input data space onto the network structure.

Several quantitative metrics have been proposed to evaluate topology preservation quality and error, like the topographic product [Bauer & Pawelzik, 1992] or the topographic function [Villmann & Martinetz, 1994]. However these measures cannot distinguish between folding of the map along nonlinearities in the data manifold and folding within a data manifold. Thus, at this stage, a visual inspection of the map to detect neighbourhood violations is at least trustworthy. A particular metric may serve as an optimization factor to improve overall performance but since they only return one value, is difficult to attribute quantitative accuracy.

10.6. Conclusions

A biologically plausible solution have been explored, to abolish the undesirable effect known as 'catastrophic forgetting' and these ideas extended to well known unsupervised learning neural networks. All of the aforementioned algorithms in the literature review, related with self organization, processes only static or offline data. The proposed memory model exhibited on-line knowledge retention and abstraction, that attenuates with time. A series of experiments demonstrated that SOM and NG were able to map the input space quite satisfactory, for different kind of data distributions. Temporal information recorded well and the spatial representation was accurate. As it was expected, nearby neurons appeared adjust in time and space. Furthermore, a simulated robot mapping situation have been modeled with self-refreshing variants of SOM, NG and GNG as the main learning algorithms. The results dictated that all three scenarios were able of online map building where both map building and localization run continuously and in parallel. All the algorithms faced some gradual forgetting of older data as new data were acquired. Gradually forgetting previous experience is not a totally undesirable effect because it enables the online or sequential learning of information in artificial neural networks. This attribute suggests a framework for analysis and modelling of lifelong learning and developmental effects in cognitive systems. Memory usage was limited and predefined, independent of spatial coverage dimensions or time of exploration.

11. Conclusions – Future Work

This chapter presents the primary contribution to knowledge and recommendations for further work.

11.1 Contribution to Knowledge

11.1.1 Appearance based map building

An appearance based neurocomputational architecture that emulates hippocampal place cells learning has been validated on a simulated mobile robot, and the results demonstrated the ability to efficiently map the surrounding environment. Low level, global image features, based on color and various texture features, have been proved particularly descriptive, in the context of a CBIR process. The proposed methodology was able to process position information, in order to build a topological map, without identifying special landmarks. This method, make use of visual only cues about the environment which were able to generate “concept patterns” where a robot can search upon to identify it's position. A series of experiments demonstrated that SOM and NG algorithms were able to map the input space quite satisfactory, for different kind of data distributions. Temporal information recorded well and the spatial representation was accurate. Further research might explore alternative visual scene interpretation methods.

11.1.2 Robot Navigation

A less explored area is the application of evolutionary methods to design developmental robots. This practice involves evolutionary processes to optimize parameters of developmental core algorithms, usually belonging to the field of self-organization. Both evolutionary and developmental adaptation are mutually constraint. Developmental processes referred to as evolutionary artefacts [Harvey et al., 2005] because each individual plasticity entails costs which must be offset by a higher level regularity in developmental outcome. Similarly, evolution exploits adaptation advantages but is always constraint by what is developmentally possible. By isolating adaptation in stages, focusing on the application for which each stage have been proved to be most appropriate, and have been shown that this strategy could be useful for capturing both dynamics of interaction between robots and their environment while exhibiting higher level cognitive behavior. In this thesis, an evolutionary cognitive architecture have been proposed to enable a mobile robot to cope with the task of visual navigation. Initially a graph based world representation is used to build a map, prior to navigation, through an appearance based scheme using only features associated with color information. During the next step, a genetic algorithm evolves a navigation controller that the robot uses for visual servoing, driving through a set of nodes on the topological map. Experiments in simulation show that an evolved robot, adapted to both exteroceptive and proprioceptive data, is able to successfully drive through a list of sub-goals minimizing the problem of local minima in which evolutionary process can sometimes get trapped. This approach was proven to be more expressive for defining a simplistic fitness formula yet descriptive enough for targeting specific goals.

11.1.3 Pseudo-Rehearsal Learning

A connectionist model, designed to avoid catastrophic interference, applied on popular unsupervised topology preservation networks. Experiments demonstrated that an unsupervised neural network can be benefited from the self-refreshing learning procedure in order to avoid the catastrophic forgetting phenomenon. Episodic mapping was feasible, and past memories decayed more naturally. All algorithms, under consideration, demonstrated sequential learning capabilities. This novel approach also applied to address the problems of continuous robot mapping in an artificial environment. The self-refreshing model proved to be an astonishing simple technique which can credit memory models with the ability of life long learning whenever required.

11.2 Future Work

The relationship between emotion and memory is complex, but generally, emotion tends to increase the likelihood that an event will be remembered later and that it will be remembered vividly. Some biologists believe that stressful incidents causes the formation of long lasting memories by the brain. An expansion to the episodic learning mechanism with emotional modulation, by replicating specific more vivid information, may help to simulate enhancement effect on memory through capturing of attention. That way, considering a topological framework only, a mobile robot may be capable of building an environment representation online, in which the nodes correspond to salient scenes with high emotional weight, and arcs correspond to known pathways.

Evolutionary adaptation need to be evaluated for different scenarios and for higher complexity environments while incorporating better visual descriptors extracted from views of the environment. Rehearsal learning may also been applied as the core developmental process instead of the off-line variations that have been used so far. After verification of the aforementioned algorithms using simulations, these need to be evaluated on actual robots and modify as necessary to ensure acceptable real life robot navigation.

References

Akers E. L. and Agah A., "Topological Localization Using Appearance-Based Recognition," Journal of Automation Mobile Robotics Intelligent Systems, vol. 4, no. 1, pp. 68–84, 2010.

Angeli A., Doncieux S., Meyer J. A., and Filliat D., Incremental vision-based topological SLAM. Ieee, 2008, pp. 22–26.

Ans B. and Rousset S., "Neural networks with a self-refreshing memory: Knowledge transfer in sequential learning tasks without catastrophic forgetting," Connection Science, vol. 12, no. 1, pp. 1–19, 2000.

Ans B., Rousset S., French R. M., and Musca S., "Self-refreshing memory in artificial neural networks: learning temporal sequences without catastrophic forgetting," Connection Science, vol. 16, no. 2, pp. 71–99, 2004.

Argyros A. A., Bekris K. E., and Orphanoudakis S. C., Robot homing based on corner tracking in a sequence of panoramic images, vol. 2. IEEE Comput. Soc, 2001, p. II–3–II–10.

Ascani A., Frontoni E., Mancini A., and Zingaretti P., Robot localization using omnidirectional vision in large and dynamic outdoor environments. Ieee, 2008, pp. 576–581.

Ascani A., Frontoni E., Mancini A., and Zingaretti P., "Feature group matching for appearance-based localization," in IEEERSJ International Conference on Intelligent Robots and Systems, 2008.

Bais A., Sablatnig R., and Gu J., Single landmark based self-localization of mobile robots. IEEE Computer Society, 2006, p. 67.

Baker S. and Nayar S. K., A theory of catadioptric image formation, vol. pp, no. January. Narosa Publishing House, 1998, pp. 35–42.

Baldassarri P., Puliti P., Montesanto P., and Tascini G., "Self-Organizing Maps versus Growing

Neural Gas in a Robotic Application," in Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks: Part II: Artificial Neural Nets Problem Solving Methods, 2003, pp. 201–208.

Barrera A. and Weitzenfeld A., "Biologically-inspired robot spatial cognition based on rat neurophysiological studies," Autonomous Robots, vol. 25, no. 1, 2008, pp. 147–169.

Bauer H. U. and Pawelzik K. R., "Quantifying the neighborhood preservation of self-organizing feature maps," IEEE Transactions on Neural Networks, vol. 3, no. 4, pp. 570–579, 1992.

Booij O., Terwijn B., Zivkovic Z., and Krose B., "Navigation using an appearance based topological map," Proceedings 2007 IEEE International Conference on Robotics and Automation, no. April, pp. 3927–3932, 2007.

Brooks R. A., "Artificial Life and Real Robots," Artificial Intelligence, vol. 48, no. 2, pp. 3–10, 1992.

Burgard W., Fox D., Hennig D., and Schmidt T., "Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids," in Artificial Intelligence, 1996, vol. 2, pp. 896–901.

Butz M. V., Shirinov E., and Reif K. L., "Self-Organizing Sensorimotor Maps Plus Internal Motivations Yield Animal-Like Behavior," Adaptive Behavior, vol. 18, no. 3–4, pp. 315–337, 2010.

Campbell N. W. and Thomas B. T., "Automatic Selection of Gabor Filters for Pixel Classification," 1997.

Carpenter G. A. and Grossberg S., "The ART of adaptive pattern recognition by a self-organizing neural network," Computer, vol. 21, no. 3, pp. 77–88, 1988.

Cartwright B. A. and Collett T. S., "How honey bees use landmarks to guide their return to a food source," Nature, vol. 295, no. 5850, pp. 560–564, 1982.

Casarrubias-Vargas H., Petrilli-Barceló A., and Bayro-Corrochano E., EKF-SLAM and Machine Learning Techniques for Visual Robot Navigation, vol. 0. Ieee, 2010, pp. 396–399.

Chui, Charles K. (1992). *An Introduction to Wavelets*. San Diego: Academic Press. ISBN 0-12-174584-8.

Clausi D. A. and Jernigan M. E., "Designing Gabor filters for optimal texture separability," Pattern Recognition, vol. 33, no. 11, pp. 1835–1849, 2000.

Collett M., "How desert ants use a visual landmark for guidance along a habitual route.," Proceedings of the National Academy of Sciences of the United States of America, vol. 107, no. 25, pp. 11638–11643, 2010.

Collett T. S. and Judd S. P. D., "Multiple stored views and landmark guidance in ants," Nature, vol. 392, no. 6677, pp. 710–714, 1998.

Cosens D. and Toussaint N., "An experimental study of the foraging strategy of the wood ant Formica aquilonia," Animal Behavior, vol. 33, pp. 541–552, 1985.

Cummins M. and Newman P., "Probabilistic Appearance Based Navigation and Loop Closing," Proceedings 2007 IEEE International Conference on Robotics and Automation, no. April, pp. 2042–2048, 2007.

Datta R., Joshi D., Li J., and Wang J. Z., "Image retrieval: Ideas, influences, and trends of the new age," ACM Computing Surveys, vol. 40, no. 2, pp. 1–60, 2008.

Dayoub F. and Duckett T., An adaptive appearance-based map for long-term topological localization of mobile robots. IEEE, 2008, pp. 3364–3369.

Dijkstra E. W., "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

Dill M., Wolf R., and Heisenberg M., "Visual pattern recognition in Drosophila involves retinotopic matching.," Nature, vol. 365, no. 6448, pp. 751–753, 1993.

Dissanayake M. W. M. G., Newman P., Durrant-Whyte H. F., Clark S., and Csorba M., "An Experimental and Theoretical Investigation into Simultaneous Localisation and Map Building," The Sixth International Symposium on Experimental Robotics VI, vol. 250, pp. 265–274, 2000.

Dyer F. C., "Bees acquire route-based memories but not cognitive maps in a familiar landscape," Animal Behavior, vol. 41, no. 2, pp. 239–246, 1991.

Elman J. L., "Finding Structure in Time," Cognitive Science, vol. 14, no. 2, pp. 179–211, 1990.

Engelson S. P. and McDermott D. V., Error correction in mobile robot map learning, vol. 3. IEEE, 1992, pp. 2555–2560.

Fiala M. and Basu A., "Robot navigation using panoramic tracking," Pattern Recognition, vol. 37, no. 11, pp. 2195–2215, 2004.

Field D. J., "Relations between the statistics of natural images and the response properties of cortical cells.," Journal of the Optical Society of America A Optics and image science, vol. 4, no. 12, pp. 2379–2394, 1987.

Floreano D., Kato T., Marocco D., and Sauser E., "Coevolution of active vision and feature selection.," Biological Cybernetics, vol. 90, no. 3, pp. 218–228, 2004.

Floreano D., Suzuki M., and Mattiussi C., "Active Vision and Receptive Field Development in Evolutionary Robots," Evolutionary Computation, vol. 13, no. 4. pp. 527–544, 2005.

Fox, D., Burgard, W., & Thrun, S. (1999). Markov Localization for Mobile Robots in Dynamic Environments. Journal of Artificial Intelligence Research, 11, 391–427.

Fox D., Burgard W., Dellaert F., and Thrun S., "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," in Proceedings Of The National Conference On Artificial

Intelligence, 1999, no. Handschin 1970, pp. 343–349.

Franz M. O. and Mallot H. A., “Biomimetic robot navigation,” Robotics and Autonomous Systems, vol. 30, no. 1–2, pp. 133–153, 2000.

Franz M. O., Schölkopf B., Mallot H. A., and Bülthoff H. H., “Learning view graphs for robot navigation,” Autonomous Robots, vol. 5, no. 1, pp. 111–125, 1998.

Fraundorfer F., Engels C., and Nister D., Topological mapping, localization and navigation using image collections. Ieee, 2007, pp. 3872–3877.

French R., “Catastrophic forgetting in connectionist networks.,” Trends in Cognitive Sciences, vol. 3, no. 4, pp. 128–135, 1999.

French R. M., “Catastrophic Forgetting in Connectionist Networks : Causes , Consequences and Solutions,” Trends in Cognitive Sciences, vol. 3, no. 4, pp. 128–135, 1999.

Fritzke B., “A Growing Neural Gas Network Learns Topologies,” Advances in Neural Information Processing Systems 7, vol. 7, no. 1, pp. 625–632, 1995.

Gabor D., “Theory of communication,” Communication Theory, vol. 93, no. 26, pp. 429–457, 1946.

Gerecke U. and Sharkey N., Quick and dirty localization for a lost robot, vol. 1. IEEE Service Center, 1999, pp. 262–267.

Gerecke U., Sharkey N. E., and Sharkey A. J. C., “Common evidence vectors for self-organized ensemble localization,” Neurocomputing, vol. 55, no. 3–4, pp. 499–519, 2003.

Gonzalez-Barbosa J. J. and Lacroix S., Rover localization in natural environments by indexing panoramic images, vol. 2, no. May. Ieee, 2002, pp. 1365–1370.

Gould J. L., “The locale map of honey bees: do insects have cognitive maps?,” Science, vol. 232, no. 4752, pp. 861–863, 1986.

Greene N., Environment Mapping and Other Applications of World Projections, vol. 6, no. 11. IEEE, 1986, pp. 21–29.

Hafner V. V., “Learning Places in Newly Explored Environments,” in Meyer Berthoz Floreano Roitblat and Wilson Eds SAB2000 Proceedings Supplement Book Publication of the International Society for Adaptive Behavior. 2000.

Harvey I., Paolo E. D., Wood R., Quinn M., and Tuci E., “Evolutionary Robotics: A New Scientific Tool for Studying Cognition,” Artificial life, vol. 11, no. 1–2, pp. 79–98, 2005.

Howarth P. and Rüger S., “Robust texture features for still-image retrieval,” Image Rochester NY, vol. 152, no. 6, pp. 868–874, 2005.

Ishii K., Nishida S., and Ura T., A self-organizing map based navigation system for an underwater robot, vol. 5. IEEE, Piscataway, NJ, USA, 2004, p. 5726.

Jain A. K., “Data clustering: 50 years beyond K-means,” Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010.

Jensfelt P. and Kristensen S., Active global localization for a mobile robot using multiple hypothesis tracking, vol. 17, no. 5. IEEE, 2001, pp. 748–760.

Knapek M., Oropeza R. S., and Kriegman D. J., Selecting promising landmarks, vol. 4, no. April. Ieee, 2000, pp. 3771–3777.

Kohonen T., “Self-organized formation of topologically correct feature maps,” Biological Cybernetics, vol. 43, no. 1, pp. 59–69, 1982.

Kohonen T., Self-Organizing Maps, vol. 30. Springer, 1995, p. 521.

Koskela T., Varsta M., Heikkonen J., and Kaski K., Temporal sequence processing using recurrent SOM, vol. 1, no. April. Ieee, 1998, pp. 290–297.

Kröse B. J. A., Vlassis N., and Bunschoten R., "Omnidirectional Vision for Appearance-based Robot Localization," in Int Workshop on Sensor Based Intelligent Robots, 2002.

Kurz A., "Constructing maps for mobile robot navigation based on ultrasonic range data.," IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society, vol. 26, no. 2, pp. 233–242, 1996.

Lee T. S., Image representation using 2D Gabor wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(10):959–971, Oct. 1996.

Leonard J. J. and Durrant-Whyte H. F., Mobile robot localization by tracking geometric beacons, vol. 7, no. 3. IEEE, 1991, pp. 376–382.

Leonard J. J. and Durrant-Whyte H. F., Simultaneous map building and localization for an autonomous mobile robot, vol. 3, no. 91. Ieee, 1991, pp. 1442–1447.

Loevsky I. and Shimshoni I., "Reliable and efficient landmark-based localization for mobile robots," Robotics and Autonomous Systems, vol. 58, no. 5, pp. 520–528, 2010.

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, *15*(4), 151-190. Taylor & Francis

Lynch K., The Image of the City, vol. 21, no. October. MIT Press, 1960, p. 194.

Lynch K., What time is this place?, vol. 65, no. 1. MIT Press, 1972, p. 288.

MacQueen J. B., "Kmeans Some Methods for classification and Analysis of Multivariate Observations," 5th Berkeley Symposium on Mathematical Statistics and Probability 1967, vol. 1, no. 233, pp. 281–297, 1967.

Manjunath B. S. and Ma W. Y., "Texture features for browsing and retrieval of image data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 837–842, 1996.

Marimpis A., Papadourakis G., Palamas G., “Uniform Color Spaces Clustering in an Unsupervised Manner”, 13th IEEE International Symposium on Computational Intelligence and Informatics, 2012.

Martinetz T. and Schulten K., “A ‘Neural-Gas’ Network Learns Topologies,” Artificial Neural Networks, vol. 1. University of Illinois at Urbana-Champaign, pp. 397–402, 1991.

Martin M. C. and Moravec H. P., “Robot Evidence Grids,” Robotics, no. March, 1996.

Ma W. Y. and Manjunath B. S., A comparison of wavelet transform features for texture image annotation, vol. 2. IEEE Comput. Soc. Press, 1995, pp. 256–259.

McCloskey M. and Cohen N. J., “Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem,” Psychology of Learning and Motivation, vol. 24, pp. 109–165, 1989.

McClelland J. L., McNaughton B. L. & O'Reilly R. C., “Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Psychological Review, 102, 419–457, 1995.

Miller J. and Carlson L., “Selecting landmarks in novel environments.,” Psychonomic bulletin review, vol. 18, no. 1, pp. 184–191, 2011.

Minsky M. L. and Papert S. A. 1969. *Perceptrons*. Cambridge, MA: MIT Press

Mouret, J.-B., & Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. Evolutionary Computation, 20(1), 91–133

Nehmzow U., “Map building through self-organisation for robot navigation,” in Advances in Robot Learning 8th European Workshop on Learning Robots EWLR8 Proceedings Lecture Notes in Artificial Intelligence Vol1812 SpringerVerlag Berlin Germany, 2000, vol. 1812, pp. 1–22.

Nehmzow U. and Smithers T., “Map building using Self-Organising Networks in Really Useful Robots,” in Proceedings of the First International Conference on Simulation of Adaptive Behavior, 1991, pp. 152–159.

Nehmzow U., “‘Meaning’ through Clustering by Self-Organisation of Spatial and Temporal Information,” vol. 1562, C. Nehaniv, Ed. Springer Berlin / Heidelberg, 1999, pp. 209–229.

Nelson A. L., Barlow G. J., and Doitsidis L., “Fitness functions in evolutionary robotics: A survey and analysis,” Robot. Auton. Syst., vol. 57, no. 4, pp. 345–370, Apr. 2009.

Nolfi S., “Evolving robots able to self-localize in the environment: the importance of viewing cognition as the result of processes occurring at different time-scales,” Connection Science, vol. 14, no. 3, pp. 231–244, 2002.

O’Keefe J. and Dostrovsky J., “The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat.,” Brain Research, vol. 34, no. 1, pp. 171–175, 1971.

O’Keefe J. and Nadel L., The Hippocampus as a Cognitive Map, vol. 3, no. 3. Oxford University Press, 1978, p. 570.

Oliva, A., & Torralba, A. (2006). Building the gist of a scene: the role of global image features in recognition. In Progress in Brain Research (p. 2006).

Palamas, G.Houzard, J.-F.; Kavoussanos, M.; “Relative Position Estimation of a Mobile Robot in a Greenhouse Pathway”, IEEE International Conference on Industrial Technology, 2006, pp. 2298-2302.

Palamas G., Ware A., “Sub-goal based Robot Visual Navigation through Sensorial Space Tesselation” International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No.11, 2013.

Palamas G., Papadourakis G., Kavoussanos M., Ware A., “Unsupervised Topology Preserving

Networks that Learns Sequentially", International Journal of Computational Intelligence Research, Vol.2, No. 1 (2006), pp. 66-71

Pitas I. and Venetsanopoulos A.N., Nonlinear Digital Filters: Principles and Applications, Kluwer Academic Publisher, 1990.

Robins A. and McCallum S., "The consolidation of learning during sleep: comparing the pseudorehearsal and unlearning accounts.," Neural Networks, vol. 12, no. 7–8, pp. 1191–1206, 1999.

Robins A., "Catastrophic forgetting, rehearsal and pseudorehearsal," Connection Science, vol. 7, no. 2, pp. 123–146, 1995.

Robins A. and McCallum S., "Catastrophic forgetting and the pseudorehearsal solution in Hopfield-type networks," Connection Science, vol. 10, no. 2, pp. 121–135, 1998.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.

Se S., Lowe D. G., and Little J. J., Vision-based global localization and mapping for mobile robots, vol. 21, no. 3. IEEE, 2005, pp. 364–375.

Siagian C. and Itti L., Biologically Inspired Mobile Robot Vision Localization, vol. 25, no. 4. IEEE, 2009, pp. 861–873.

Sim R. and Dudek G., "Comparing image-based localization methods," in Pattern Recognition, 2003, vol. 18, pp. 1560–1562.

Smith J. R. and Chang S., "Tools and techniques for color image retrieval," Image Rochester NY, vol. 2670, no. 3, pp. 426–437, 1996.

Smith R. C. and Cheeseman P., "On the representation and estimation of spatial uncertainty," The International Journal of Robotics Research, vol. 5, no. 4, pp. 56–68, 1987.

Strösslin T., Sheynikhovich D., Chavarriaga R., and Gerstner W., “Robust self-localisation and navigation based on hippocampal place cells.,” Neural Networks, vol. 18, no. 9, pp. 1125–1140, 2005.

Suzuki M. and Floreano D., “Evolutionary Active Vision Toward Three Dimensional Landmark-Navigation,” in SAB, vol. 4095, Springer-Verlag, 2006, pp. 263–273.

Suzuki M. and Floreano D., “Enactive Robot Vision ,” Adaptive Behavior , vol. 16 , no. 2–3 , pp. 122–128, Apr. 2008.

Tamimi H. and Zell A., Global visual localization of mobile robots using kernel principal component analysis, vol. 2. Ieee, 2004, pp. 1896–1901.

Tarutoko Y., Kobayashi K., and Watanabe K., Topological Map Generation based on Delaunay Triangulation for Mobile Robot. Ieee, 2006, pp. 492–496.

Thrun S., “Simultaneous Localization and Mapping,” The International Journal of Robotics Research, vol. 23, no. 7–8, pp. 693–716, 2008.

Thrun S., “Robotic Mapping: A Survey,” Science, vol. 298, no. February, pp. 1–35, 2002.

Tolman E., “Cognitive maps in rats & in men,” Psychological Review, vol. 55, no. 4, pp. 189–208, 1948.

Topp, E. A., Human-Robot Interaction and Mapping with a Service Robot : Human Augmented Mapping. - Stockholm : KTH, 2008. - (Trita-CSC-A ; 2008:12)

Trullier O., Wiener S. I., Berthoz A., and Meyer J. A., “Biologically based artificial navigation systems: review and prospects.,” Progress in Neurobiology, vol. 51, no. 5, pp. 483–544, 1997.

Ulrich I. And Nourbakhsh I., Appearance-based place recognition for topological localization, vol. 2, no. April. Ieee, 2000, pp. 1023–1029.

Varsta M., Heikkonen J., and Lampinen J., “Temporal Kohonen Map and the Recurrent Self-Organizing Map : Analytical and Experimental Comparison,” Neural Processing Letters, vol. 13, no. 3, pp. 237–251, 2001.

Villmann T., Der R., and Martinetz T., A new quantitative measure of topology preservation in Kohonen’s feature maps, vol. 2, no. Martinetz. IEEE Service Center, 1994, pp. 645–648.

Vyas V. S. and Rege P., “Automated Texture Analysis with Gabor filter,” Transform, vol. 6, no. 1, pp. 35–41, 2006.

Wang Y. W. Y., Chen D. C. D. , and Shi C. S. C. , Autonomous Navigation Based on a Novel Topological Map, vol. 1. Ieee, 2009, pp. 254–257.

Wehner R., Bleuler S., Nievergelt C., and Shah D., “Bees navigate by using vectors and routes rather than maps,” Naturwissenschatten, vol. 77, pp. 479–82 ST – Bees Navigate by Using Vectors and Ro, 1990.

Welch G., Bishop G., 1995. *An Introduction to the Kalman Filter*. Technical Report. University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

Werner F., Sitte j., and Maire F., “Automatic Place Determination using Color Histograms and Self-Organising Maps,” in International Conference on Advanced Robotics, 2007, pp. 111–116.

Wichert G. V., Mobile robot localization using a selforganized visual environment representation, vol. 25, no. 3–4. Elsevier, 1997, pp. 29–36.

Yamada S., “Recognizing environments from action sequences using self-organizing maps,” Applied Soft Computing, vol. 4, no. 1, pp. 35–47, 2004.

Yamada S., “Evolutionary behavior learning for action-based environment modeling by a mobile robot,” Applied Soft Computing, vol. 5, no. 2, pp. 245–257, 2005.

Yamada S. and Murota M., Applying self-organizing networks to recognizing rooms with behavior sequences of a mobile robot, vol. 3. IEEE, 1996, pp. 1790–1794.

Zhang J. Z. J. , Tan T. T. T., and Ma L. M. L., Invariant texture segmentation via circular Gabor filters, vol. 2, no. 2. IEEE Comput. Soc, 2002, pp. 901–904.

Zagal, J. C., Ruiz-del-Solar, J., & Vallejos, P. (2004). Back to reality: Crossing the reality gap in evolutionary robotics, Juan Cristóbal Zagal, Javier Ruiz-del-Solar and Paul Vallejos. In Notes. Elsevier Science.

APPENDIX A – Published Papers

(In descending chronological order)

Palamas G., Ware A., “Sub-goal based Robot Visual Navigation through Sensorial Space Tesselation” International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No.11, 2013.

Marimpis A., Papadourakis G., Palamas G., “Uniform Color Spaces Clustering in an Unsupervised Manner”, 13th IEEE International Symposium on Computational Intelligence and Informatics, 2012.

Palamas, G.Houzard, J.-F.; Kavoussanos, M.; “Relative Position Estimation of a Mobile Robot in a Greenhouse Pathway”, IEEE International Conference on Industrial Technology, 2006, pp. 2298-2302.

Palamas G., Papadourakis G., Kavoussanos M., Ware A., “Unsupervised Topology Preserving Networks that Learns Sequentially”, International Journal of Computational Intelligence Research, Vol.2, No. 1 (2006), pp. 66-71

APPENDIX B – Data CD