Learning Symbolic Maps from Robot Navigation

Bachelor Thesis in Cognitive Science

Christopher Lörken
Institute of Cognitive Science
University of Osnabrück
cloerken@uos.de

Supervisors:
Prof. Dr. Ute Schmid, University of Bamberg
Prof. Dr. Martin Riedmiller, University of Osnabrück

July 26, 2004

ABSTRACT

Navigation in real life inspires powerful approaches used in robotics. Giving an overview of various biological navigation systems and their influence on the field of robotics, this thesis will present an extension of such an approach by introducing a knowledge-based map learning simulation. The presented system will be enlarged to mimic the biological correlates even more realistically by simulating differently skilled agents. The focus will lie on the agent's knowledge representation and therefore on the acquisition and maintenance of symbolic maps representing the environment the agent is operating in. The different skills of the agents will mainly be simulated by distorting their sensory perception that is equivalent to the implementation of measurement noise from the technical point of view. The presented system will be able to deal with some of the problems concerning map building that occur when distorted sensory input is integrated in the simulation. Moreover the performances of diversely skilled agents will be extensively tested and evaluated showing the robustness of the approach.

CONTENTS

1. INTRODUCTION AND OVERVIEW

Navigational tasks are encountered in many situations of our everyday life. Proper navigation thereby implies some sort of device or technique allowing us to orient ourselves. This becomes especially interesting when acquiring knowledge about new and unknown environments.

If you e.g. find yourself in an unknown area and you try to find your way it adds up to a combination of keeping track of your position while moving as well as trying to find familiar places to get an idea where you are. These two strategies can be seen as referring to two different kinds of navigational knowledge. The first one is *local* and the latter *global* knowledge. To extend the example above: Imagine you travel by bus through a fairly well-known city and you miss your bus stop. You might have to get out at a place where you have not been before. *Local navigation knowledge* means that you are able to find your way back to the bus stop while exploring that area whereas *globally* you try to find some familiar location that you can recognize from a previous visit to get a more general idea of where you are.

Dealing with the most diverse organisms, like humans, desert ants or pigeons, this is where different skills and techniques come into play that allow us to navigate. This familiar location could e.g. mean 'town hall' to a human, some 'special alignment of earth's magnetic field' to a migratory bird or a 'familiar looking laser range scan' to a robot.

As nature often is a good teacher, it is quite evident to try to copy those real-life techniques to implement them in the field of mobile robotics. Since a robot trying to find its way from the coffee machine back to its owner's office has to solve a simple navigational task that is comparable e.g. to a desert ant returning with some food to its den. The field dedicated to mimic nature's basics in robots is called *biomimetic robotics*[1].

This thesis will cover one aspect of biomimetic robotics namely biologically inspired navigation or, to be more precise, biological plausible representations of spatial knowledge in robot navigation. Therefore selected techniques of animal and human navigation will be introduced from an interdisciplinary point of view comparing their functionality and biological structures to state-of-the-art methods and devices used in robot navigation.

Hereby, the focus will mainly lie on map building and knowledge representation leading to the introduction of a simulation environment, developed by Baumann (2003), that is capable of simulating knowledge-based map learning by an agent navigating through a graph-like labyrinth. Hereby the agent builds and maintains a symbolic map representation of of its environment using memory like structures that allow it to recognize previously visited places and therefore to localize itself in the global context of previous navigation attempts.

[1] Biomimesis: to mimic life, to imitate biological systems (Cutkosky, 2004)

The system will be extended to gain an even more biologically plausible approach leading to the implementation of agents that are differently skilled mainly in terms of their sensory perception. From the technical perspective, this can be compared to enlarging the system to deal with measurement noise.

However, this brings along some problems concerning map building since representations based solely on distorted data get inconsistent. Addressing this problem an algorithm will be presented to deal with at least one type of the encountered errors while furthermore the whole system will be extensively tested.

The results gained from the experiments conducted will be evaluated to estimate the agents' performances under the influence of different abilities concerning their senses as well as the general capabilities of the extended simulation regarding its robustness.

The next section will describe the structure of this thesis.

1.1 Outline

Chapter 2 - From Rodents to Robots. An overview of biological navigation techniques found in a wide variety of animals as well as in humans will be given pointing out the important biological structures and functions involved relating and comparing them to approaches of contemporary real-world robotics.

Chapter 3 - The Simulation Environment. The program developed by Baumann (2003) will be described focusing on its biological plausibility as well as its knowledge representation system.

Chapter 4 - Extending the Simulation. This chapter will present the extensions made to the introduced simulation system relating them to the concepts presented in chapter 2. These extensions consist mainly of implementing different skills concerning the agents sensory perception introducing furthermore an algorithm based on linear relaxation that takes care of some of the spatial representation problems resulting from the first part of this chapter.

Chapter 5 - Experiments. The experimental design to test the extended agents' performances will be described yielding to an evaluation of the results and therefore of the overall system robustness and performance.

Chapter 6 - Conclusion. The results of the thesis are summarized discussing some possible further extensions of the presented system.

2. FROM RODENTS TO ROBOTS

Modelling biological plausible navigation and knowledge representation in robots implies a good understanding of the basic principles found in real life navigation. Therefore this chapter will introduce a selection of navigation techniques found in a diverse variety of organisms. Hereby, both the navigation methods as well as the biological instruments necessary to fulfil these methods (i.e. the different senses) will be presented and finally compared to the methods and instruments of contemporary robotics. This survey will focus in particular on the encountered internal representations of spatial knowledge, since this will be of major concern for the following chapters[1].

But first of all, there is the need to give some basic definitions about navigation as the defined concepts will be used to describe and evaluate the different techniques found in real and artificial life.

2.1 Navigation - Basics

Route-Following vs. Way-Finding - In general, navigational tasks can be split up into the two categories of route-following and way-finding.

Route-following relies on the so-called *route knowledge*, which consists of *sequences* of locations or *landmarks* (i.e. uniquely identifiable locations (see below)) as well as actions that lead from one location to another. Route-following corresponds to the task of following a familiar route like ones way to work every morning. Whereby *route* is formally defined as a concatenation of connected places (cf. Werner et al., 2000). Route-following is sometimes referred to as low-level navigation since it nearly may be performed unconsciously (cf. McNamara and Shelton, 2003).

Way-finding on the contrary relies on the so-called *survey knowledge* and is considered to be a somewhat higher process than route-following as it is related to the task of planning a novel, previously unknown, route from a source location to a target location. It is regarded to be deliberate and consciously controlled (cf. McNamara and Shelton, 2003). Therefore, survey knowledge contains a more detailed representation of the environment than route knowledge. It contains the real *spatial layout* of the area being a metric representation of the locations and their relations to each other. Hereby, this layout is aligned with respect to some kind of *reference frame* (see below). Both route and survey knowledge are commonly represented in a *map*.

[1] A good general overview applying to many of the following topics is also given in (Mallot and Franz, 1999)

Maps - A map is a (usually two dimensional) representation of space, containing a set of connected places which are spatially related to each other by transformations (cf. O'Keefe and Nadel, 1978, page 86). Hereby, places and relations have to be represented explicitly with regard to the reference frame.

Reference Frames - Reference frames can be defined as follows:

> "Put simply, a *reference frame* is a means of representing the locations of entities in space"[2].

This becomes clear with explaining the two most prominent reference frames, the *egocentric* reference frame and the *allocentric* reference frame (cf. Klatzky, 1998)[3].

Egocentric puts the navigating actor or *agent* itself into the center. All other positions within the environment are described and, more importantly, *change* relatively to that agent. I.e. if an agent changes its *heading* (the orientation it has), the whole map of the world rotates with respect to that change while the agent itself remains fixed.

Allocentric on the contrary assumes a fixed, global and aligned reference point. This is some kind of origin of a global coordinate frame acting as directed reference point. Therefore, a change of the agent's heading would only rotate the agent and not the whole system.

Viewpoint - The agent's viewpoint or *view* is the direction of its heading, thus its line of sight. Therefore, an agent might experience one and the same point as different viewpoints by entering a location coming from different directions. This might lead to place recognition problems (see section 2.3).

Landmark - Landmarks are some special places in the environment, that are (normally) easy to recognize and are frequently considered as being unique. Landmarks can be e.g. the "town hall" for a human, or a "rock" and a "tree" for an ant. Some experiments concern the learning of landmark configurations, i.e. different landmark alignments in space. In many cases this happens with artificial landmarks like for instance "a red L-shaped piece of wood" that has, amongst others, been used to train pigeons to find a food source by Blaisdell and Cook (2004).

Cognitive Map - The term cognitive map was firstly introduced by Edward C. Tolman in 1948 (see Tolman, 1948) (cf. section 2.2.1). It is a biologically inspired expression that can be interpreted as a *neural representation* of the environmental layout that is believed to be located in the hippocampus[4]. And even though some people argue, that this term has been defined in too many controversial ways and therefore should not be used (cf. Bennett, 1996), I feel it is spread widely enough, especially in most recent publications, to adopt it. Nevertheless it will hereby be restricted to its most common feature namely the encoding of an allocentric

[2] (Klatzky, 1998, page 1)

[3] Sometimes, beneath quite a big amount of other expressions, they are referred to as protagonist's and antagonist's view, idiothetic and allothetic or for instance in linguistics, as intrinsic and deictic reference frame (cf. e.g. Claus et al., 1998).

[4] See section 2.2.4 for a brief description of the hippocampus and its functions in humans.

representation of objects and places in space (Blaisdell and Cook, 2004). Pertaining to this definition cognitive maps are strongly related to the above mentioned survey knowledge, since they provide the necessary biological structure to represent it as well as the functionality to extrapolate on that representation to build up novel routes.

Forming the most important structure related to spatial navigation in organisms, the subsections of 2.2 will especially account to activities related to cognitive maps.

Keeping these definitions in mind, the next section will give a biological view of navigational methods and skills found in some animals and humans focusing on knowledge acquisition and spatial representations.

2.2 Navigation in Real Life

2.2.1 Rodents

Traditionally, rodents and rats respectively are very well studied animals since they showed supreme skills in navigational tasks especially in the field of maze-learning. Actually it were rats which inspired Edward C. Tolman to come up with the expression of a cognitive map as he was the first to point out that the rat's maze behavior is not only a matter of mere stimulus response. Instead, he stated that the rat's stimuli input is elaborated into a temporary "cognitive-like map" indicating routes and relations of the environment (cf. Tolman, 1948). A theory that has many times been approved since then (e.g. O'Keefe and Nadel, 1978).

Today we know that the cognitive map in rats and therefore their surpassing navigational performance is highly supported by the fact that rats have some special cells in the hippocampus namely the so-called *place cells* and *head-direction cells* (see e.g. O'Keefe and Nadel, 1978; McNamara and Shelton, 2003).

Place cells - The place cells provide the rat with a unique place recognition ability as they have location dependent response patterns. I.e. the neurons fire in the same way if the rat returns to a known location. The spatial area where place cells fire is called the place (or firing) field. However, this response pattern can only be triggered if the rat knows where it is in a *global* meaning (cf. O'Keefe and Nadel, 1978). I.e. the place cells do not fire when the rat is put somewhere in its firing field without having a general clue where it is. So it seems that they use at least some environmental knowledge like for instance the location of adjacent landmarks to localize themselves. This actually implies an allocentric representation of the rat's position (cf. Werner et al., 2000).

Interestingly, Save et al. (1998) found this place cell behavior even in blind rats. So, although they are used when present, visual cues seem to play not such an important role after all. Instead it was stated that dynamic, motion-related information in conjunction with stimulus recognition should be sufficient to develop the place recognition ability (page 1818).

Even more recently, O'Keefe and Burgess (2004) particularized these results by stating that the cells respond to the distance of the rat from the walls of the environment whereas each cell prefers different combinations of walls. They found, that the cell's response was triggered through a combination of visual cues and movement information.

Head-direction cells - Furthermore, the hippocampal head-direction cells have been found to fire whensoever the rat orients itself to a particular heading. Thus actually performing the work of a compass and providing the rat with an allocentric heading information (Klatzky, 1998).

Representation - Klatzky (1998) states, that these special skills of the rats allow them to integrate their ego- and allocentric locational information (the rat perceives distances egocentrically), in combination with an allocentric heading component, "to produce higher-level representations and support functioning in space" (page 9). This *higher-level representation* can be interpreted as a *cognitive map* (O'Keefe and Nadel, 1978).

Recapitulating this section, rats owe their extraordinary performance in maze navigation to some special type of cells within their hippocampal formation. These cells seem to provide them with a compass-like sense, as well as a unique place-recognition system that can be integrated into an allocentric high-level representation, a cognitive map, of their environment and their location inside it.

2.2.2 Pigeons and Migratory Birds

Similarly to rats, pigeons are known as excellent navigators, this time even in a larger scale. They are able to return to their roost over vast distances without any knowledge whatsoever of where they have been released. Generally, the task to return to ones home is called *homing*. To build up their environmental representation, pigeons mainly use three different senses that will be explained here before taking a closer look on their representation.

Olfaction - The pigeon's homing ability is largely based in their *olfactory* sense. Whilst in their pigeonry, they associate odors with wind directions and assessing the odor of their release place, they build a spatial map representation with regard to their previous knowledge. Experiments have shown severe homing problems in pigeons with disabled olfactory senses, and as well in pigeons with air-filtered roosts that have not been able to collect any olfactory impressions within their home environment (cf. Breed, 2001).

Compass sense - Additional to their highly specialized olfactory sense pigeons have, as well as rats, a compass sense. But as opposed to the rodents whose head-direction cells were also functional in the dark pigeons rely on the sun to judge the direction of north, regarding their feeling of time. Again, pigeons show strong homing problems in experiments, this time, if their internal clock is manipulated. For instance by setting them out to artificial light, pretending an unscheduled sunrise (cf. Breed, 2001).

However, pigeons as well as many migratory birds show some additional sense of earth's magnetic field. Being still topic of research, the theory arose that they are equipped with a vision-based compass. The assumption is that radical pair processes influenced by the magnetic field affect the sensitivity of the light receptors in their eyes (see Ritz et al., 2000).

Landmarks - The third homing technique of pigeons is again a well known concept, namely landmark-based navigation. Blaisdell and Cook (2004) successfully set up an experiment where pigeons where trained to find a food source in an small-scale environment with some artificial landmarks positioned in there. Rearrangement of those landmarks resulted in a different search behavior by those previously trained pigeons.

Representation - In their landmark experiments, Blaisdell and Cook (2004) found direct evidence for what pigeons are actually able to encode in their cognitive maps. They discovered that pigeons are able to encode landmark - goal vectors, thus spatial relations from a landmark to a goal (in this case to the hidden food source). Moreover the pigeons were able to learn landmark - landmark vectors. Thus they can encode spatial relations without the need of a direct stimulus as a kind of reinforcement learning. Instead, the pigeons were even able to infer the goal location from landmarks that have previously not been associated with the goal at all, showing that they have been able to somehow compute novel relations between learned locations. Blaisdell and Cook (2004) explained this ability by suggesting that the pigeons have to be able to integrate the different spatial relations into a larger, higher-order map representation of their surroundings and hence a fully functional cognitive map. One could assume that pigeons might also be able to integrate the information of their other senses into this representation what would explain their supreme homing abilities.

Subsuming this section, the navigational skill of pigeons is most likely based on integrating olfactory impressions with an allocentric sensation of north and a landmark recognition system into some form of representation that allows them to perform their unique homing abilities. This integration into a higher-order map representation has already been proven against their landmark-based navigation while this representation fulfils the criteria of a cognitive map.

2.2.3 Cataglyphis - The Desert Ant

Normal ants navigate by secreting pheromone trails leading to a food source and back to their den. So they simply follow their olfactory sense without involving any special skills in their homing behavior. But in contrast to normal ants, the Cataglyphis, a desert ant, that is primarily found in semi-arid regions of Tunisia and Greece, has developed a quite different and far more interesting kind of navigation which will be explained here.

This special ant is capable of performing *odometric path-integration* a navigation style that is frequently used in robotic navigation as well and will now be explained.

Odometric path-integration - means that the ant integrates

> "all directional changes and distances travelled into a homing vector, representing the exact direction and distance of the nest

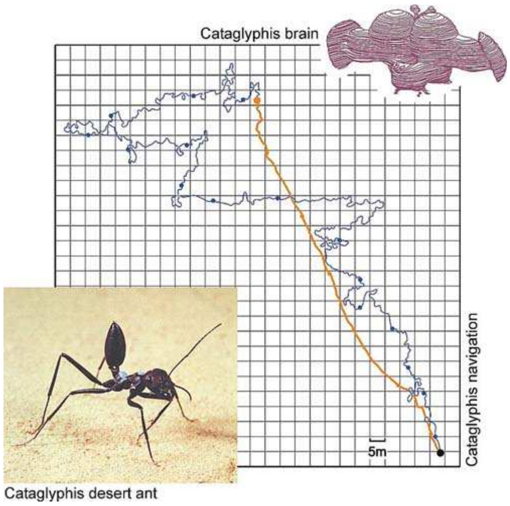

Fig. 2.1: Homing behavior of the Cataglyphis. The blue line represents the foraging behavior of the ant. The yellow line shows the way it took back to its den (i.e. the homing vector). (With kind permission of Rüdiger Wehner, Zürich.)

entrance"[5],

and to its starting point respectively (see also Wehner et al., 1996). These proprioceptive (i.e. by the ant experienced) changes in heading and distance are called *odometric information* while the integration of those changes into a representation of spatial position changes within the environment is called *path-integration*. Navigation based on odometry is prevalently referred to as *dead-reckoning*.

Figure 2.1 shows the Cataglyphis and an exemplary way it took looking for food as well as the integrated *homing vector* visualized by the direct path it took back to its den.

In robotics path-integration is frequently used to keep track of the robot's position. It works similarly to that found in the ant, as the robots movements are proprioceptively measured for instance by its wheel encoders that usually supply data like the current wheel speed in x and y axis of an egocentric reference frame. This information can be integrated over time to extrapolate the robot's position within an allocentric reference frame with respect to the starting point of the robot. Since this is actually the navigation principle used in the simulation, robotic path-integration will again be topic in chapter 3.

At this point, however, it should be sufficient to point out an *accumulative error* in the estimation of the global position. This error grows, as measurement faults add up in each step, that is integrated into the path. It is one of the major problems of path-integration (see section 4.2), especially when dealing with error-prone direction estimates. To face this shortcoming, the ant is equipped with a device insusceptible to at least orientation

[5] (Werner et al., 2000, page 300)

faults. As suggested by the skills of rats and pigeons this device is a compass sense.

Compass sense - The desert ant's compass works again differently compared to those found in birds and rodents. It is to say an innate sky-light compass (cf. Wehner et al., 1996; Peper and Tolani, 2001). Cataglyphis is able to see the specific pattern of the spectral component as well as the polarization that light leaves when entering earth's atmosphere. Therefore it can localize the general direction of its den, as it can judge directions by these visual cues (cf. Ronacher et al., 2000), whereas visual cues of landmarks are only used, if the ant is already close to its home.

Summarizing this section, Cataglyphis' navigation is mainly based on a dead-reckoning path-integration system, that is supported by a compass. This relatively simple system that is also commonly used in robot navigation allows the ant to accomplish high performance homing while working reliably enough to support foraging speeds of approximately 15m/s and covering distances of a couple of hundred meters.

Obviously this kind of navigation does not really require the previously mentioned high-order representation of a cognitive map, as the ant is not able of computing novel routes based on its knowledge. The only information it tracks over time are no spatial relations between places but a simple information of the relative, and therefore egocentric, distance and direction to its den.

2.2.4 Humans

Lacking these specialized navigation devices of the various presented species, it is no surprise that people sometimes, even usually, get lost in unknown and untrained areas. Unless, of course, they use some navigational tools to compensate for their impairment. Like e.g. a compass, a sextant, the mileage counter in a car, a GPS device or simply a map.

Nevertheless, humans show great skills in acquiring knowledge about new environments, by simply navigating through them (Rothkegel et al., 1998). They especially have got the ability of explicitly learning spatial relations between distinct locations, thus generating survey knowledge.

Therefore, in particular since the onset of non-invasive brain imagery in the form of *PET*[6], *MRI*[7] and particularly *fMRI* (i.e. *functional MRI*, thus working with subjects, that do specific tasks) scanning techniques, the human brain has been intensively studied.

After giving some substantiating arguments for the existence of a spatial cognitive map in humans the important role the hippocampus plays in the acquisition of spatial memory will be indicated.

[6] PET stands for Positron Emission Tomography. A radioactive tracer is injected into the subject's blood circuit, leaving a track, that can be scanned by the PET scanner. This technique can be used to highlight brain regions that are involved in the subjects current task.

[7] MRI stands for Magnetic Resonance Imaging. The nuclei of tissue molecules are aligned through a strong magnetic field. After the field has been turned off, the nuclei change back to their original configuration, releasing a detectable energy that can be transferred into a quite detailed image of the scanned tissue.

Structured representation - Doing some experiments concerning spatial learning of new environments, Werner and Schmidt (1998) for instance discovered that people often err systematically when remembering spatial locations. They argue that this is evidence of some kind of basic underlying structure of spatial memory.

Integration - This concurs e.g. with the results of Werner et al. (2000) who state, that

> "learning between landmarks is incidental and largely irrelevant. With more experience, however, information about different routes is integrated into a network-like structure and distance information is added."[8].

This means that humans are capable of integrating their environmental knowledge over time and to refine it to gain a more consistent and detailed version of their spatial representation. One could assume that this more detailed representation would allow a human to produce novel routes like shortcuts more successfully than on a sparse representation.

Allocentric-based computation - Additionally Rothkegel et al. (1998) did an experiment where subjects had to judge the distances between objects in their environment. Using euclidian distances between distinct objects they guaranteed that the subjects were using an allocentric reference frame, which is computationally more difficult than judging with respect to ones egocentric reference frame.

They found that it took the subjects increasingly longer to estimate the distance between two objects with an increasing number of other objects lying between them. This shows, that only adjacent locations are stored with explicit distance information and that no shortcuts (in the sense of skipping intermediate places) are explicitly stored.

Instead, the subjects had to calculate the overall distance by summing up the individual distances between all the locations (cf. Rothkegel et al., 1998). This refers to the the concept of survey knowledge, as introduced in section 2.1.

Subsuming these findings leads to the conclusion of humans having a cognitive spatial map. Since it would be conform to the definition of the cognitive map as providing the necessary representation of space as well as the functionality to extrapolate on that representation to develop novel routes within a known environment. These are clearly tasks a human can accomplish.

For reasons of completeness the following paragraph will conclude this overview of basic real life navigation principles and spatial knowledge representation by introducing the brain structure that is supposed to contain those neural structures accountable for the functionality of the cognitive map, the hippocampus.

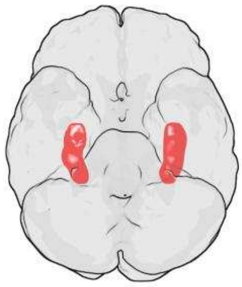

Fig. 2.2: The hippocampus in the human brain. The figure displays a horizontal view of the brain (i.e. from above). The head's front is at the top (from Wikipedia, 2004).

The Hippocampus

Being part of the limbic system, the hippocampus is located in the temporal lobe of the human brain[9] (see figure 2.2).

It plays an important role in the acquisition of new memories as well as in navigational tasks. As a complete description of the functionality and versatility of the hippocampus would be much too complex, I can only give a brief overview and have to refer to the literature for a more precise version (see Kandel et al., 2000; Hartley et al., 2003).

Researchers from the field of cognitive psychology came to the conclusion of the hippocampus playing an important role in the formation of new memories about personally experienced events. Yielding the so-called *episodic memory* that can be interpreted as memory for things that happened in a particular place and time. This process of combining personal experiences with events is also referred to as *associative learning*. The assumption that the hippocampus is the structure accounting for this kind of learning and memory formation is primarily based on the fact that patients, who suffer on lesions of the hippocampal area, show severe cognitive deficits resulting from the hippocampal dysfunction, in this case the disability to form new memories. A disease that is called *anterograde amnesia*.

Concerning the hippocampus' role in navigation another very well known disease related to hippocampal degeneration is *Alzheimer's disease* with spatial disorientation being one of its very early symptoms.

The two main questions demanding clarification apply to the location and the structure of the cognitive map.

Location - The hippocampus' role in navigation has been surveyed for instance by O'Keefe and Burgess (2004) who have studied which parts of the human brain are activated while performing tasks on spatial navigation. Doing PET and fMRI scans on subjects navigating through virtual environments, they have found that the hippocampus and the adjacent parahippocampal

[8] (Werner et al., 2000, page 297)

[9] To be a little bit more precise: Superior of the brain stem within the ventral temporal cortex.

cortex showed intense signs of activation. A result that can even be reproduced if the subjects only imagine to navigate through their environment.

These results have been confirmed and specified even more precisely by Hartley et al. (2003) (cf. also McNamara and Shelton, 2003). They have been able to show that route and survey knowledge are indeed subject to unequal brain structures.

Testing human subjects in navigating through virtual environments, Hartley et al. (2003) found a major activation within the area of the posterior hippocampus and regions of the medial temporal lobe when doing wayfinding tasks, thus using survey knowledge. Combined with activation in those areas related to sensory perception, as e.g. the occipital lobe for visual perception, they assumed some kind of *perceptual-spatial processing* involved in navigating via novel routes symbolizing the usage of a cognitive map.

On the contrary, studying route-following tasks, thus route knowledge, they found primary activation of the *caudate nucleus* that is a structure of the basal ganglions and known to be responsible for voluntary movements. Hereby those areas identified in wayfinding tasks and furthermore those needed for sensory perception almost showed no activation at all. They explained the nearly complete absence of this activation by assuming an action-based route representation that does not need perceptual-spatial processing and instead allows acting like an autopilot (cf. Hartley et al., 2003, page 885).

Another very interesting survey, involving London's taxi drivers, has been carried out by Maguire et al. (2000). To achieve a licence, a taxi driver in London has to pass a test involving the memorization of about 25.000 streets plus many landmarks in a 6 mile radius of London. The mean training time to pass this test is about two years. Maguire et al. (2000) discovered a clearly identifiable volume increase of the posterior hippocampal grey matter accompanied by a volume decrease of the anterior region again pointing out the great importance of the hippocampus concerning survey knowledge and therefore the cognitive map in humans.

Structure - Concerning the structure of the cognitive map, O'Keefe and Nadel (1978) expressed that

> "the map system is assumed to have two basic components: a mapping space and a locative mechanism for building and changing maps"[10].

An assumption confirmed e.g. by Nadel et al. (1998) who have transferred the famous water maze experiment to humans producing some remarkable results. Usually, the water maze experiment contains some kind of water basin filled with a cloudy, non-transparent fluid. A set of landmarks is arranged surrounding this basin and a rat is trained to find an invisible platform beneath the fluid's surface. This test was extended to humans who were trained using a virtual environment. The really interesting aspect of their study is, that *elderly* people performed significantly worse in

[10] (O'Keefe and Nadel, 1978, page 102)

this experiment than middle aged adults or juveniles. The elder subjects performed well on completing a puzzle reconstructing the maze arena and its landmarks but failed in placing the target into that puzzle after the training. They performed so badly, that Nadel et al. (1998) were able to say, that none of them found the target consistently.

This impairment of locating a position in contrast to maintaining a fully functional representation of the environment suggests that the assumption of O'Keefe and Nadel (1978) is true and that the cognitive map really provides two separated neural structures concerning representation and assessment of the map.

To sum up this section, the hippocampus seems to be strongly involved within the process of forming and representing spatial memory, especially concerning survey knowledge, as well as to access it and to compute on that representation. Hippocampal areas have even been proven to grow as a result of extreme usage. Whereas hippocampal dysfunction and degeneration leads to severe problems concerning, amongst other things, navigation.

The cognitive map itself, as defined in this context, seems to be mainly located in the posterior hippocampus and also shows signs of a two-part architecture related to mapping and processing.

Subsuming this section about real life navigation, a quite diverse selection of nature's navigation methods and instruments has been introduced. May this be the rat's place cells, the pigeon's olfactory sense or the path-integration and sky-light compass of the desert ants. Moreover the term of a *cognitive map* has been established and evidence of its existence, location and functionality as a processable higher-order map representation of spatial relations within the environment has been given.

Therefore the next chapter will finally come to the technical side of navigation describing a selection of the most important methods and devices used in contemporary robotics to model the introduced navigational behaviors of biological systems.

2.3 Navigation in Artificial Life

2.3.1 Sensory Perception

Being able to navigate in an environment naturally implies the ability to sense it somehow. A robot's *sensors* are those input devices that allow the robot to perceive its environment, to locate itself in it and to navigate through it. Hence sensors play an as important role in robotics as senses in nature. Therefore the most important sensors used in robotics will now be briefly introduced before discussing their limitations compared to the biological senses that inspired them.

Laser range finder - Laser range finders send out a laser beam to measure distances. Their measurements are robust and accurate but without being able to detect transparent obstacles like glass.

Ultrasonic Transducers - Ultrasonic transducers, also known as *sonar sensors*, are found in nature e.g. in bats. They send out short pulses of ultrasound usually at 40 kHz measuring the time it takes the sound to bounce from an obstacle and return to the robot giving an estimate of the distance from that object. The advantage of sonar sensors is that they can detect "invisible" objects like glass. Nevertheless they have got problems with sound absorbing obstacles as well as special object alignments. In room corners for instance, a sound signal is reflected with different time delays from both walls that form the corner. The result would be two distinct sound signals, measured by the robot, yielding a wrong distance estimation.

Infrared - Infrared (IrDA[11]) are very cheap short distance sensors. They do a good job in detecting nearby obstacles.

Tactile - Tactile sensors are touch sensors that are normally used to prevent the robot from keeping to steer into the direction where it just hit an obstacle.

Camera - Camera sensors count to the most important sensors in robotics as a picture of the surrounding area usually contains a lot of relevant data for short to intermediate distance navigation. The main camera systems are mono-, stereo- and omni-vision. As the first two are probably self-explanatory the latter usually consists of a camera mounted vertically onto the robot and a parabolic mirror that is formed in a way to provide a 360 degree view of the area. The advantage of camera systems is that they really contain a lot of valuable information but the bad part is that it is computationally expensive and, indeed, non-trivial to extract the relevant information.

Compass and Global Positioning System (GPS) - These sensors provide an exact global orientation and in the case of GPS a global satellite guided positioning but are unfortunately inapplicable under some circumstances like for instance in indoor environments.

Proprioception - The *self-measurement sensors*. They mostly measure the movements of the wheels by so called *wheel encoders* to provide odometric measurements. Unfortunately these proprioceptive sensors are quite error-prone leading to the problems discussed in section 4.2.

The relation of most of these sensors to equivalent senses in nature should be quite obvious. Nevertheless, there are a couple of points that should be regarded explicitly as all of those sensors share major disadvantages when comparing them to their biological archetypes, namely their efficiency and accuracy.

Resolution and data rate - For instance a modern camera picture, as it can be computationally handled by now in the field of robotics, has got a resolution of 640x480 pixels, what is far less than e.g. the human eye can perceive. Furthermore the data-rate to transfer and evaluate sensory perception is much higher in living creatures than in robots. E.g. it is up to now impossible to build a robot that would be able to fly through a tree at full speed as the needed *frame-rate* (i.e. the time needed to process 1 image of a camera) would be significantly higher than technically possible so far.

[11] I.e. Infrared Data Association sensors.

Distortion - Additionally to being slow, artificial senses provide only low-resolution output being furthermore highly inaccurate as they are limited to their physical implementations which are unfortunately quite error-prone. As already mentioned some of them, like compass or GPS, fail to operate in certain environments completely. Others, like the sonar sensor and the odometric devices, have to deal partially with high amounts of measurement errors, a corruption of sensor readings known as *measurement noise* or *distortion*.

Integration - As every sensor has its shortcomings, consistent sensor data provided by just one kind of sensor are usually very sparse. The common way to deal with this is to combine the sensors and integrate their readings to form a more complete and more consistent representation. As for instance a laser range finder cannot detect transparent objects, it can be combined with a sonar sensor that can detect obstacles like glass but has its problems with sound absorbing objects that are, on the other hand, no problem to the laser range finder.

An example for a highly efficient integration of different sensor systems can be found e.g. in Burgard et al. (1999), who equipped a robot for navigating within a crowded museum with a combination of sonar sensors, tactile sensors, laser range finders, infrared and a stereo camera system.

Biological plausibility - Nevertheless, noisy measured data have a biological equivalence as well. For example a cheap and old camera probably yields a stronger distorted picture than a brand new one with a higher resolution. The biological equivalent would be that elderly people can usually see noticeably worse compared to younger people. Or to put it more general, varying agents of a species might have different special skills or diversely elaborated competencies. This means it is normal to have subjects in a population that for instance can judge distances better than others.

Furthermore, biological agents as well as robots show a fluctuating competence in terms of their skills and sensory perception. As, on the one hand, a human who might be just tired performs most likely worse in certain tasks than when being completely awaken. A robot, on the other hand might encounter these fluctuations because of some temporary technical difficulties like battery drain or simply dirt on the lens of the camera.

Accordingly, modelling agents with diverse sensory skills from a biological point of view can be compared to set up biomimetic robots that are equipped with sensors of different quality or with sensors that are distorted differently.

An approach that will indeed be used later on to model diversely competent agents that will be analyzed in their navigation performances (see 4).

Having given an overview of the input devices a robot might have, and therefore its sensory input to perceive its environment, the next step will be to introduce the way it deals with all this input namely its *knowledge representation*.

2.3.2 Knowledge Representation

Somehow the robot has to represent all its sensory information. Inspired by the above introduced theory of a cognitive spatial map (cf. section 2.2), *maps* form the most common kind of spatial robotic knowledge representation as they are highly biologically plausible and moreover have the benefit of providing a good visualization for humans of what the robot has learned.

Although man-made maps, like e.g. CAD[12] maps, are used as well in mobile robotics, they will be skipped here as this thesis is concerned in autonomous map learning hence in maps that are acquired by the robot itself.

The remaining map types can be roughly divided into the two categories of *grid-based* and *topological* maps (cf. Thrun, 1998). Usually, both of these map classes are represented using allocentric reference frames. That is because each point within an egocentric maps has to be separately rotated and translated with every movement the robot performs. Therefore they are computationally much more expensive than allocentric maps in which only the robot itself has to be rotated and translated.

The following two sections will now describe these two major categories of mapping.

Grid-Based Mapping

Represented information and layout - Grid-based maps have obtained their name because of splitting up the environment into commensurate metric grids cells. These grid cells project some data connected to an area of points in reality to the two dimensional Euclidean space of the map.

In the majority of cases, this projected data are the information whether or not that particular place is regarded to be *occupied* and the *probability* of this grid being occupied respectively. This broadly used type of grid-based maps is therefore called *occupancy grids* (e.g. Thrun, 1998, 2003).

Figure 2.3 exemplarily shows a couple of occupancy grid maps that are, in this case, restricted to a small local area. Hereby, the grid cells are the actual points of the maps whereas dark points represent space with a low probability of being occupied and brighter areas reflect a high occupation probability.

The accuracy of this approach is highly determined by the actual size of the grid cells. Large grid cells code for a large space in reality whereas smaller cells provide a higher resolution sometimes opening new paths for the navigation that would habe been considered blocked by a low resolution occupancy grid. E.g. a grid size of 1×1 m might not be able to represent a doorway that would be easily recognizable in maps with a 10×10 cm resolution.

Advantages - Subsumed shortly, the advantages of grid based approaches are mainly as follows:

[12] CAD stands for Computer-Aided Design. It refers to an area of engineering based tools to visualize some design like maps for instance.

- They are capable of producing accurate metric maps, even in *large-scale environments*[13] as a sufficiently high resolution would be able to acquire the needed accuracy.
- Occupancy grids can be updated *online* (i.e. while the robot explores the area) endowing the ability of coping with dynamic environments.
- While being rather easy to implement, they frequently yield optimal results in computing a *shortest path* to a target location (provided a sufficiently high resolution).
- The grid-based place recognition system is prevalently *viewpoint independent* rarely leading to similar input for distinct locations.

Disadvantages - The disadvantages of grid-based maps on the contrary are mainly:

- They are susceptible to measurement noise. Only partially wrong position estimates can already mark unoccupied areas as being the opposite thus leading to an erroneous map. Nevertheless, this problem looses importance when dealing with the online approach. In this case the error would vanish after some time since the approach is able of mapping dynamic environments.
- Moreover grid-based maps bear some problems in combination with *symbolic problem solvers* (cf. Thrun, 1998) as associating a symbolic meaning to each grid cell would be computationally expensive. I.e. an operator normally wants to state something like "Go to the coffee machine in the kitchen" instead of "Go to grid location 224, 487".
- The probably most important drawback, however, is that these maps are extremely space consuming. The grid resolution has to be chosen rather high as the system is, unfortunately, independent from the environment's complexity (cf. Thrun, 1998). A grid size of 2×2 m would be somewhat space saving but it is just inappropriate for a robot of the dimensions 20×20 cm. This problem might, however, be diminished by using an online approach capable of changing the grid cell resolution during runtime.
- The high complexity is also disadvantageous for computing e.g. a shortest path on the map[14]. Though grid-based navigation may yield the optimal solution the computational costs to achieve this solution may be too high.

In conclusion, grid-based maps, or occupancy grids, are capable of providing exact (dynamic) map representations of an environment while unfortunately bringing along rather high demands in matters of space and computational complexity. This complexity prohibits, amongst other things, the functionality of assigning symbolic meanings to places and complicates the finding of a shortest path.

[13] An environment is considered a large-scale environment, if it reaches far beyond the sensor range of a static robot (Kuipers and Byun, 1991).

[14] For a short description of how path-planning may be accomplished in grid-based maps refer to section 2.3.4.

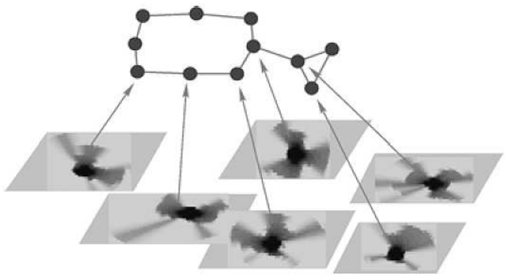

Fig. 2.3: An example for topological and grid-based maps. A topologically connected set of places is obtained by associating and evaluating local occupancy grids. (With kind permission of Tom Duckett, Örebro.)

Topological Mapping

> "Topological maps correspond to the minimal models of an axiomatic theory describing the relationships between the different sources of information explained by a map."[15]

Represented information and layout - Representing such a *minimal model*, topological maps usually consist of *graphs*. I.e. they consist of some *nodes* and some *arcs* (or *edges*) connecting these nodes[16]. The actual appearance of a graph forming a topological map can be seen in figure 2.3.

Hereby, the nodes represent distinct places within the environment while the arcs, relating them with each other, can represent a sequence of actions the robot has to perform to travel from one node to another. But an even more common way to assign meaning to the arcs is to integrate environmental metric information into the topological map, defining the different nodes as distinct spatial locations that are related to each other. Thus an arc is prevalently the representation of an *angle* and a *distance* between two nodes of the topological map.

Acquiring the map - If not built online, topological maps are acquired by collecting sensor data, building a metric representation of the environment and afterwards selecting distinct points of interest to form the map. Thus topological maps are often built based on purely metric maps like e.g. occupancy grids (cf. figure 2.3). Hereby, the usage of topological maps avoids many of the problems concerned with grid-based maps by simply reducing the maps complexity.

The selection of the places that ought to become the topological nodes can be done in a variety of ways e.g. by *Voronoi diagrams*.

Put simply, the idea of a Voronoi diagram is to insert lines into the map that have the maximum distance to nearby surrounding obstacles. The intersections of these lines become the nodes of the topological map whereas the lines form the arcs. The resulting topological map depicts the most

[15] (Remolina and Kuipers, 2004, page 47)
[16] For a complete definition of graphs see section 3.2.2.

secure route for a robot to follow as it travels in a maximized distance to all obstacles in that area.

E.g. Thrun et al. (1998) are using Voronoi diagrams to select "critical points and lines" to divide grid-based maps into distinct regions. The centers of those regions became the nodes of a topological representation. In that example they were able to reduce the 27,280 occupied cells of their occupancy grid map that was built integrating sonar sensor readings as well as a stereo camera system, to only 67 nodes in a topological graph (Thrun et al., 1998, page 7).

Advantages - Topological maps have some advantages as compared to grid-based maps.

- The example given above points out one of the major benefits of topological maps. They are extremely space-efficient compared to full metric representations like occupancy grids. Such a map allows highly efficient path-planning algorithms like e.g. A* to show their strengths (used in this context e.g. in Bolduc et al., 1997; Ferguson et al., 2003).

  However, the result is normally suboptimal as a topological navigating robot travels from node to node within the graph being prohibited e.g. from cutting edges. Nevertheless the time and space savings clearly justify the non-exactly optimal route.
- Topological maps are furthermore quite tolerant concerning local positioning problems. If the robot's position is estimated wrongly for a few frames (i.e. update cycles of e.g. a camera) it is normally an error not severe enough to locate the robot e.g. in the room next door. Hence the computed path for the robot remains valid.
- Moreover topological maps support the mentioned usage of symbolic problem solvers since in principle a *symbolic map* is a topological map in which the nodes have a symbolic meaning.

  The symbolic meaning describes some features, like "entrance sign" or "shelve with cereals" in a grocery store, that are assigned to the different locations of the map (cf. e.g. Fu et al., 1996).

  Thus a symbolic knowledge representation actually easily supports the combination with symbolic problem solvers as it allows commands like "Go to the coffee machine in the kitchen".

Disadvantages - Together with being suboptimal concerning shortest paths, the major disadvantage of topological mapping concerns the online building of the map as this brings along the problem of integrating noisy sensor data into a consistent map representation. This will be described in section 4.2 in greater detail. Hereby, the problem is that for constructing topological maps locations that have been previously visited have to be recognized in order to merge these places with the according already known node. Otherwise, the graph becomes inconsistent, since new nodes are added instead of closing a loop. This problem gets eminently severe in large-scale environments as it gets harder the longer the way is back to the already known node. The problem can be described as a *global localization problem*, as the robot looses track of its global position.

Subsuming this section, topological maps provide an extremely efficient environmental representation in terms of computational and space complexity while being able of assigning symbolic meanings to places and therefore supporting symbolic problem solvers. Their disadvantage of not necessarily yielding optimal solutions for path-planning is more than compensated by their performance leaving the problems of online place recognition as the only disadvantage to be taken seriously.

The last two sections have introduced the two most commonly implemented kinds of knowledge representation, grid-based and topological maps, elucidating their benefits and disadvantages. Their relation to the biological psychological theory of the cognitive spatial map as presented in section 2.2 becomes clear by recalling the statement of Werner et al. (2000) on page 10. They argued that learning an environment takes place as learning distinct locations like landmarks and their relations to each other. This is exactly the principle basis of a topological maps with its nodes and the relating arcs.

Therefore, the simulation introduced in chapter 3 will use topological or to be more precise symbolic maps. The exact usage of this kind of map including its representation as a graph will be defined in that chapter.

For a consistent introduction to robotic navigation, however, there are two more topics that should be mentioned at least briefly. These are localization and path-planning.

2.3.3 Localization - Place Recognition

Obviously, map-based navigation is only possible if one knows one's current position on the map. As a robot does not have a built-in functionality of the rat's place cells, which could be interpreted as a localization utility, it has to locate itself in a different way. Hereby the problem of localizing a robot can be divided into two categories:

- The dynamic allocation of the robots position during runtime
- Localization under global uncertainty

The first problem may result for instance from noisy sensor data and usually results in relatively small position errors. To cope with this, a robot could use e.g. a triangulation algorithm to calculate its position relative to a couple of landmarks whose allocentric positions are known. The localization of these landmarks, for instance by a camera, can fluctuate e.g. due to blurred pictures that are actually a common phenomenon if the robot moves while taking that picture. This would result in jumps in the estimated robot position. One solution to deal with these problems is to bring in probabilistic models that estimate the robot's position on the basis of its previous position and the action it took during the last time step. Keywords concerning to this are e.g. *bayesian filtering*, *kalman filter* or *Markov localization* (cf. Thrun et al., 1998, 2000).

The problem of localization under global uncertainty is also known as the *kidnapped robot problem.* After having learned a map the robot is set somewhere without knowing its start location. The problem the robot has to face in this case is recognizing its current position by its sensor readings. For grid-based maps, probabilistic methods provide again good and widespread solutions. See *Monte Carlo Localization* (cf. e.g. Thrun et al., 2001, 2000) or *Particle Filters* (cf. e.g. Fox et al., 2000; Hähnel et al., 2003).

The general idea is to collect weighted samples where the robot might be and update weightings and samples according to the consistency of those samples regarding the current sensor readings. Therefore, improbable samples are sorted out whereas the most probable samples agglomerate roughly over time at the true robot location.

Dealing with topological maps the problem is a little different. The disambiguation of the position estimate can for instance be accomplished by taking the node's neighbors into account to identify the current one. An approach that was brought up by Gregory Dudek (cf. Dudek et al., 1993). As this is actually implemented in Moritz Baumann's simulation it is one of the topics in chapter 3.

2.3.4 Path-Planning

Going from one's current position to a target location in a known environment implies the planning of a path between those two points.

As already mentioned, grid-based approaches yield the problem of dealing with an extremely high number of points. As an example a method for grid-based navigation will be introduced here before coming to topological path-planning.

Grid-based navigation - As said in 2.3.2, the high number of locations (grids) leads to the problem of expensive computational costs to find an optimal or even a possible path.

Nevertheless there are some relatively efficient algorithms like e.g. the *dynamic programming* approach of *value iteration*. The main idea of such approaches is that they are so-called *any-time algorithms*. I.e. they provide a fast, but not necessarily optimal, solution improving them by iterations over time (cf. Thrun et al., 2000, 1998). Thus the robot is able to start moving into the direction of its goal location, refining its path-planning while moving there.

The basic idea is that to all grid cells costs are assigned: Whereas the goal has initial costs of zero all other fields have ∞ costs. These values are updated by iteration steps in the way that a (non-occupied) grid gets the costs of its cheapest neighbor plus the costs to get there. Thus, after enough iterations, every grid's cheapest neighbor is actually the right grid to go to in order to reach the target in optimal time (e.g. Thrun, 1998).

Besides providing an optimal solution this algorithm has the benefit of being robust to local localization problems. As value iteration computes the costs for all grids to go to the target location it does not matter if the robot's position estimate is partially incorrect and jumps a few grids.

Topological navigation - Path-Finding approaches for topological maps are computationally much cheaper than grid-based approaches. The easiest way would be a depth-first or breadth-first search on the graph. Containing only 67 points as in the example of section 2.3.2 this would actually be fast enough even for online navigation.

As the search for a node in a graph can actually be seen as a search e.g. in a tree, further explanations should not really be needed here.

This chapter tried to give an insight into biological as well as robotic navigation drawing some important parallels between the two fields while explaining the most important concepts to understand the problems of robotic navigation. The following chapters will put some of the encountered concepts into the context of a simulation forming a model for agent navigation in graph-like environments.

3. THE SIMULATION ENVIRONMENT

Dealing with real robots brings along some major disadvantages. One of them is that they are quite unsuitable for large-scale experiments as it both takes a lot of time to run them and furthermore brings severe limitations on the experimental design.

This especially becomes a problem when dealing with experiments that require different environmental conditions or, even worse, discriminatively skilled robots. Therefore one is left with only one choice namely to *simulate* such experiments.

In the following, the *simulation environment* forming the basis of the approaches evolved in chapter 4 will be introduced while establishing relations to the presented concepts of biological and robotic navigation[1].

3.1 Defining the Task

Section 2.1 pointed out two major categories of navigational tasks namely *path-following* and *way-finding*. Hereby, way-finding was considered to be the more complex process regarding it being deliberate and consciously controlled. It was stated that way-finding is based on so-called survey knowledge that is gained from a spatial layout representation of the agent's environment[2].

As way-finding tasks are the scope of this simulation, the program simulates an agent trying to find its way from a source location to a target location in a given *graph-like* environment as seen e.g. in Fig. 3.1.

Therefore, the agent has, at first, to try to localize itself in a global context regarding previously learned knowledge of that area. Afterwards, by knowing its position, the agent becomes capable of planning a route to its target location.

This task can be understood as the task of navigating through a labyrinth. It is e.g. comparable to the rat's maze learning tasks or, to pick an example from real world robotics, to Ferguson et al. (2003) or Hertzberg and Kirchner (1996) who used mobile robots to map abandoned mines and sewerage pipes respectively.

The focus of the following will lie on the ability of maintaining a symbolic cognitive map representation of the environment, allowing the agent to localize itself under global uncertainty within a previously traversed graph-like environment and labyrinth respectively.

[1] The original simulation was developed by Moritz Baumann who therefore provides a more detailed description in Baumann (2003).

[2] Since simulations do not deal with real robots, the term *agent*, or if convenient to the model *cognitive agent*, is usually preferred and therefore will here as well be used exclusively in the following.

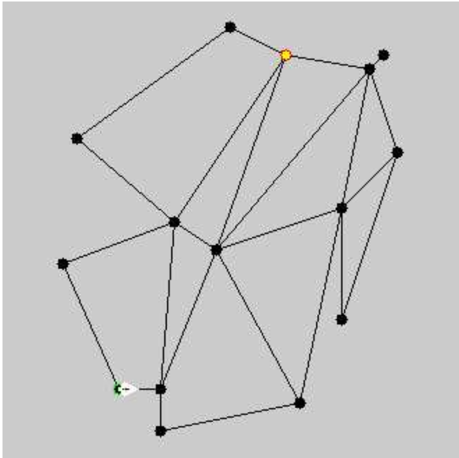

Fig. 3.1: The agent's environment. Yellow nodes represent landmarks whereas the agent is visualized as a white arrow. The green circle around the agent's position depicts the source location whereas the red circle around the landmark depicts the target location.

It remains to be pointed out which and how those biological fundamentals introduced during the last chapter are modelled in the simulation environment.

Therefore, the underlying knowledge representation will be introduced in a biological and technical as well as in a formal simulation related context within the next section followed by a description of the cognitive skills the agent possesses before illustrating the accomplished extensions in chapter 4.

3.2 Knowledge Representation and World Model

3.2.1 Biomimetic Perspective

Because of the biological plausibility of a spatial cognitive map, this concept will be integrated into a memory-based structure. The simulation provides the structures of a *long-term memory* (LTM) as well as with a *short-term memory* (STM). Hereby previously gained knowledge is stored in the LTM while the STM provides the full computational power and functionality of a cognitive map.

Short-term memory - The STM is normally related to the immediate processing of information without the possibility to store this information permanently (cf. Stillings, 1987)[3].

Hence an agent operating in one of the simulated labyrinths will use its STM to store information of the locations it has just visited, thereby acquiring a spatial representation, a cognitive map, of its environment. The locational data stored consist of the spatial relations between distinct points as well as of some symbolic meaning that is associated with the location leading to a topological, *symbolic map*.

[3] In the context of this thesis the term of the STM will be restricted to this very basic definition as it is sufficient to point out its affinity to the cognitive map.

This map is built by navigating through the environment. Simply by judging distances and directions between distinct locations a structure evolves representing the path that the agent took from its starting point to its actual location.

This methodology of integrating proprioceptive movement data into a spatial representation should meanwhile be quite familiar from examples like the homing behavior of the Cataglyphis (section 2.2.3) or the dead-reckoning, path-integration system based on odometric data, e.g. from wheel encoders, in mobile robotics.

As the STM is only capable of storing information temporarily it is only used to build up the map of *one* search attempt in a labyrinth that will later on be stored in the LTM for permanent reference. Searching the same labyrinth again will produce a completely new map that has to be matched with those maps stored in the LTM as will be explained in the next paragraph.

The operations needed for way-finding, route-following and matching the current search locations to previously known ones are processed on this model of a cognitive map and in the STM respectively.

Long-term memory - On the contrary to the STM, the long-term memory is capable of storing information over time but cannot operate on it directly (cf. Stillings, 1987). It holds the knowledge of all previous searches within an environment or labyrinth to provide global context to the search the agent is currently performing.

As you would try to find a familiar place around an unknown bus stop in a familiar city (see section 1) the agent tries to recognize places it has previously visited by trying to match its current position to locations within its LTM by comparing their features.

If this match succeeds, the agent is able to merge the map connected to that location in its STM with the connected map in its LTM which leads to a larger integrated map. The new integrated map is generated in the agents STM and even if it does not already show the current search target the search space would be highly reduced as many paths and locations of that labyrinth might already have been visited.

This might not lead to the optimal solution of the shortest path, but as Baumann (2003) has shown the resulting search time is drastically reduced compared to a depth-first search on the environment.

As chapter 4 will describe some extensions of the original simulation, chapter 5 will evaluate the encountered performances on a wide variety of labyrinths approving the results found by Baumann for even these extended agents.

In conclusion, the agent will be equipped with two connected main structures that can be seen as a model of long- and short-term memory. While the LTM takes care of storing navigational knowledge permanently, the STM takes over the functionality of a cognitive map, hence acquisition of the map including its matching against the LTM as well as the necessary processing to solve way-finding and route-following tasks.

The following section will give a more formal overview of how these models are represented.

3.2.2 Simulation Perspective

The simulation has to take care of modelling and therefore simulating the over all environment, the sensors of a robot and furthermore the memory architecture described in section 3.2.1. Hence a formal description of the used techniques will be given here.

Environment Model - As already mentioned the environments dealt with consist of different graph-like labyrinths. Therefore they are represented as *undirected, non-reflexive, planar graphs*[4].

- Hereby, a *graph* G is defined as

$$G = (V, E), \tag{3.1}$$

with V being a set of vertices or *nodes* of the form that each node $v \in V$ consists of a tuple of coordinates $(x, y) \in \mathbb{R}^2$ describing a location in space. The set V of all nodes is denoted by

$$V = \{v_1, \ldots, v_N\}. \tag{3.2}$$

Therefore, graph G has exactly N nodes.
E is the set of *edges* with $E \subseteq [V]^2$, thus each edge $e_{ij} \in E$, with $e_{ij} = (v_i, v_j)$, connects two nodes v_i and v_j, with $0 \leq i, j < N$ and $i \neq j$.

- Being *undirected* implies:

$$\forall e_{ij} \in E \rightarrow e_{ji} \in E \quad \text{and} \quad e_{ij} = e_{ji}. \tag{3.3}$$

- Since Graph G is *non-reflexive*, it follows that

$$\forall v \in V \rightarrow (v, v) \notin E, \tag{3.4}$$

i.e. the current node cannot be its own successor.
This restriction of G is extended to not contain any loops ≤ 2 following that

$$\exists e \in E : e = (v_i, v_j) \rightarrow \nexists e' \in E : e' = (v_i, v_j), e \neq e', \tag{3.5}$$

thus there exists only one edge between two nodes.

- Being *planar* follows that the graph is embedded in a plane[5]. I.e. edges cannot intersect with each other. They can merely touch in their ends.

[4] This restricted form of a graph will be sufficient for the presented approach. For a more formal definition beyond these features see e.g. Diestel (2000).

[5] The graphs dealt with here are actually not only planar but more precisely even *plane*, as *planar* literally only indicates an abstract graph that can be drawn as a plane (cf. Diestel, 2000, pages 76 et seqq.).

Furthermore, the *degree* of a node specifies the number of edges leading from that node to another and according to this the grade of *connectivity* describes how dense the nodes are connected to each other. In the following, a graph G that has a connectivity of 100% is a graph in which a node e_i is *fully connected* to all its neighbors e_j that can be reached with respect to the graph's planarity. A connectivity of 0%, however, describes a *spanning tree*, i.e. a graph in which a node is connected to the least number of adjacent nodes possible.

Sensor model - The sensor model formalizes the *sensory perception* of the environment by an agent.

In the context of this simulation, agents will be able to perceive mainly three features namely *distances*, *directions* and the number of edges leading away from one node (i.e. the degree of the node). A distance perception, or *measurement*, δ_{ij} codes for the relative distance between two connected nodes, v_i and v_j in the graph that can as well be described as the length of the way, or edge, e_{ij} connecting those two nodes:

$$\delta_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} = |e_{ij}|. \tag{3.6}$$

As the number of edges leading away from a node should be quite clear, the directions the agent can travel are those angles θ_{ij} in which these edges are oriented. This angle is measured relative to the agent's heading at the start of the current search that is taken as a global reference direction. Hence a node contains a list of angles that describe the directions in which it is connected to other nodes. This leads to a second definition of edges as an edge $e_{ij} = (v_i, v_j)$ can as well be described as

$$e_{ij} = (\delta_{ij}, \theta_{ij}) \quad \text{and} \quad e_{ji} = (\delta_{ji}, \theta_{ji} - \pi). \tag{3.7}$$

I.e. the relative distance and direction from one node to another. This leads to a location description, similar to polar coordinates, describing *relations* between locations in space by distance and direction values of the form that a vertex v_j being a successor of v_i can be described as

$$v_j = v_i + \delta_{ij} \begin{pmatrix} \cos\theta_{ij} \\ \sin\theta_{ij} \end{pmatrix}. \tag{3.8}$$

The evaluation of the perceived data leads to a *system state s* describing the agent's current *configuration* namely its coordinates within a local coordinate frame as well as a heading component:

$$s = (x, y, \theta). \tag{3.9}$$

Both values are represented in an allocentric reference frame related to the agents start position and alignment that has been defined as $(0, 0, 0)$.

Furthermore, graphs may contain *landmarks* as described in section 2.1. The set of all landmarks is described as $L \subseteq V$ with a node $v_i \in L$ symbolizing a landmark and $|L| = l$ representing the number of landmarks.

As a final remark concerning the sensor model and agent configuration it is left to mention that the way between two distinct locations, is not explicitly modelled. It is assumed that the agent gets from one location to another in *one* step instead of summing up small values of spatial alterations between two places. One could regard this as taking already preprocessed odometric information consisting of locally path-integrated data leading to a consistent one step path.

Memory model - As postulated in the biomimetic section, the memory model consists of two parts that model short- and long-term memory.

The *STM* consists of an online constructed graph representation of the environment that is aligned to an allocentric reference frame relative to the agents start configuration (as described above). By traversing links within the labyrinth, leading from one location to another, the agent perceives the directions and distances travelled and integrates them into the graph, hence building a map representation of its environment.

Hereby, the assumption is made that locations, and nodes respectively are uniquely identifiable in a local meaning. Being a feature of symbolic maps this means that the place recognition system of the agent is able to recognize locations that it has already been at during that particular search attempt. This is due to the fact that a node, once visited, is assigned *symbolic meaning* of local relevance. This symbolic meaning is independent from the other nodes of the graph and may therefore be recognized at each visit to that node regardless of the direction one approaches the node.

This assumption of unique local identifiability can e.g. be compared to Dudek et al. (1991) who achieved a reliable place recognition system by equipping a robot with one or multiple markers that could have been left at prominent places and afterwards recognized. The approach used here can be regarded as a Dudek model with a sufficient number of markers. Franz et al. (1998) on the other hand were even able to show that a relatively small extraction from a panoramic 360° snapshot of a location was sufficient to uniquely identify the same location in successive visits[6].

Dealing with uniquely identifiable nodes implies that the STM's graph representation does not contain the same node twice at different locations.

The *LTM* on the contrary consists of a set of disjunct graphs that have been constructed during previous search attempts. These graphs, though being consistent within themselves, may actually contain the same node more than ones, though not within the same partial graph. This is due to the fact, that the LTM contains graphs that have been formed with respect to different starting points and orientations. Therefore the reference frames, although allocentric, do not have the same reference point and therefore are not equal as they may have been shifted and rotated differently.

The main function of the LTM is actually being a storage device accessible from the STM. So the STM can try to find nodes in the LTM *congruent* to nodes of the current search. I.e. nodes that may be rotated and translated in a way that they have the same *signature* (orientation of exits) as just

[6] As already mentioned, this behavior, called *visual homing*, can e.g. be found in honey bees.

visited nodes. This is done as introduced in Dudek et al. (1993), hence taking not only one node into account, but sets of adjacent connected nodes comparing their relations to those found in graphs of the LTM. This comparison is done up to a certain amount of neighbors where one failed comparison results in a rejection of the compared node.

Finding such congruent nodes means to localize oneself within the global context of previous searches leading to the possibility of merging the STM with the identified map of the LTM.

Note that the special case of the node being a landmark implies that it is immediately recognized in the LTM, if present, leading to the forthwith merging of the LTM and STM maps.

This process of merging two maps reduces, indeed, the number of unvisited nodes and edges and therefore the search space, fulfilling the functionality postulates of section 3.2.1.

Strategy - In the simulation, the only exploration strategy of an agent trying to find its goal within the labyrinth is a simple *depth-first search*, i.e. the agent takes the exit to its right until it gets to a node where it has already been at. From this node, the agent backtracks to a previously visited node that contains unvisited exits, following them in the same manner until it finds its goal.

If, however, the agent succeeds in recognizing a location of its search in the LTM and in merging the two graphs, the agent will in case that the goal location lies within the newly extended map perform a breadth-first search within its memory to find the shortest known path to that target location. Otherwise, the depth-first search is continued on the reduced search space.

This path found may, however, not be globally optimal, as such a path would only be found if all the edges, that the optimal path consists of, would have been merged into the current STM map representation.

To summarize this section the simulation has been designed as a biomimetic inspired approach of modelling the functionality of a cognitive map in a robotic context providing models of the necessary structures and senses within an agent as well as an environment model that supports a wide variety of different environmental settings.

The next chapter will now introduce the extensions to the presented model made to gain a even more realistic and biological as well as technical plausible solutions.

4. EXTENDING THE SIMULATION

One point in which the the program developed by Baumann (2003) is very well extendable concerns its sensor model. It is assumed that an agent is actually capable of performing *perfect* measurements of its surroundings, hence having perfect senses. Obviously, this is neither biological nor technical plausible.

Biologic agents are well known to have differently well developed skills, affecting their performance. As e.g. an experienced carpenter might judge distances in the scope of a few meters or centimeters most probably better than an accountant who has never used a measuring tape before in his whole life. As well as subjects of different age usually perform differently well on many sorts of experiments (cf. section 2.2.4 or Nadel et al., 1998) some people have impairments interfering with their actions prohibiting the accomplishment of certain tasks. That is why statistical relevant statements can only be made after conducting experiments with, the larger the better, populations of subjects whose skills and characteristics form a good mean of the overall population of that species.

As stated in 2.3.1, biological agents show as well different performances regarding the context of the tasks e.g. while being tired or exposed to extreme heat. This shows that even the processing strategies of their perception, i.e. their cognitive way to handle the task, might be temporarily affected.

In the field of robotics, these different skills or impairments become quite obvious, as different robots really are equipped with different kinds of sensors, or at last with sensors of different quality. A cheap laser range sensor might give distance estimates that differ greatly from the actual distances whereas a better sensor might lead to much more accurate measurements. Moreover the same sensors mostly perform differently well under different environmental conditions, like a camera is of nearly no use if one has to deal with frequent light changes because it lacks the ability of adaptation. These fluctuations in measurements are also referred to as *distortion* or *measurement noise* as already stated in section 2.3.

Generally speaking, different agents show to have differently well developed skills or different processing strategies when coping with their sensory perception that affect their performances in solving certain tasks.

This points out the necessity to extend the simulation to somehow model these different abilities of agents in order to get a program being able of providing a more realistic and more generally valid solution.

It furthermore leads to the question of how well the extended program will be able to handle different grades of distortion as most living creatures are at least somehow capable of dealing with such insufficiencies.

4.1 Adjustment of the Sensor Model

To be consistent to biological agents and to provide a better simulation of the technical aspects of mobile robotics, the sensor model has to be extended to support differently skilled agents. As their senses allow them to perceive odometric related changes in distance and orientation, these proprioceptive abilities have to be adapted and integrated into the simulation for getting closer to reality.

Since the agents are already capable of performing perfect estimations of measured data, they have to be somehow restricted in this ability. Therefore a distortion of these particular senses to judge distances and angles has been implemented, forming a simulation of measurement noise and somehow impaired and less competent biological agents.

Distance noise - The distance measurements performed by the agent are as defined in equation 3.6 the length of the way between two places and the length of the edge connecting two nodes in the graph respectively. Achieving this value by *dead-reckoning*, both in real life and robotics, agents should be more or less error-prone in performing these judgements. Therefore a noise function has been implemented to distort the initial correct estimated distance:

$$\delta_{ij} = \delta_{ij} \pm \varepsilon_{dist} * \delta_{ij}. \tag{4.1}$$

Hereby, ε_{dist} is a value with $0 \leq \varepsilon_{dist} \leq 1$ that describes a *maximal distortion* of the distance measurement δ_{ij} in percent. I.e. the measured distance is being falsified by a randomly chosen percentage $\leq \varepsilon_{dist}$ corresponding to the length of the measured distance. This is due to the assumption, that relatively small distances are more easily judged correctly than long distances. E.g. a distance of one meter with a ε_{dist} of 10% will be perceived by the agent as a distance of one meter $\pm\ 0 - 10$ cm. A distance of 10 m, however, would be distorted by a value of up to 1 m.

Figure 4.1 illustrates how these noisy distance measurement impair the agent's map representation.

Degree noise - A distortion being related to the size of the measured angle would lead to unreasonably different distortions as an angle of 10° would be distorted by a much smaller value than e.g. 350° though both of them describe the same angle in relation to 0°. Therefore the distortion function for angle measurements is not related to the value of θ_{ij} but is defined as

$$\theta_{ij} = \theta_{ij} \pm \varepsilon_{deg} * 2\pi. \tag{4.2}$$

Where ε_{deg}, again, stands for the upper bound of a randomly chosen distortion percentage that this time is related to 2π and therefore will distort every angle in the same way E.g. for $\varepsilon_{deg} = 10\%$ this would mean a degree noise $\leq 36°$.

Tolerance - Dealing with noisy data, an agent has to be *tolerant* in comparing previously visited nodes and edges with the current ones, as it cannot expect a perfect match. The distance tolerance range has been chosen as

$$\tau_{dist} = \varepsilon_{dist} * \delta_{ij} + \varepsilon_{dist} * \delta'_{ij}. \tag{4.3}$$

Where δ_{ij} is the just measured distance and δ'_{ij} is the recalled distance value that has to be checked for equality. The both values are considered equal if δ'_{ij} lies within $\pm\tau$ range of δ_{ij} and rejected otherwise.

Note at this point that the simulation contains an algorithm to address distorted distance measurements as will be presented later on.

The angle tolerance is analogical defined as

$$\tau_{deg} = \varepsilon_{deg} * 2\pi. \tag{4.4}$$

Figure 4.2 again visualizes an agent's map representation under the influence of noisy angle readings.

Note that these tolerance values proofed to be crucial for the agent's performance leading to the necessary assumption of a *minimum noise* to get τ_{dist} and τ_{deg}. As this is strongly connected to the experiments, it will will be discussed in chapter 5 in greater detail.

General impairment - As a third way of influencing the skills of an agent, a parameter ε_{rec} has been implemented depicting an *overall failure possibility* resulting in difficulties for the agent to recognize nodes. I.e. ε_{rec} is a percentage value according to which the agent will fail to compare two angles or distances leading not necessarily to a *fail* result but to a judgement with the different outcome of the agent's actual intension. This means that both *false positives* and *false negatives* are possible.

ε_{rec} can be regarded as being related to the cognitive skills of an agent instead of being solely based on sensory deficits as it affects the error-proneness of the agent's judgements and is not based on mere sensor insufficiency. In this sense the error can also be interpreted as a *retrieval error* describing the difficulty of recognizing something that may have been perceived correctly.

This kind of error is biologically inspired e.g. by the studies of Nadel et al. (1998) who found distinct performances by comparing elderly people with juveniles what can be modelled by this error.

These alterations form the extensions integrated into the sensor model leading to a biological and technical more relevant simulation of mobile agent navigation as they actually provide a way of simulating diverse agents with different sensory and cognitive skills allowing experiments on their performance on navigational tasks.

But as the integration of noise into the sensor model leads to a major problem of dead-reckoning path-integration systems this will be topic of the following section.

4.2 Problems of Path-Integration

As already mentioned in section 2.3.2, path-integration bears the difficulty of recognizing previously visited places solely based on odometric data. Therefore it is to say that the figures presented in fig. 4.1 and 4.2 are not completely correct. The actual problem is that the integration of distorted measurements into a graph leads to inconsistencies. Figure 4.3 visualizes that measurement

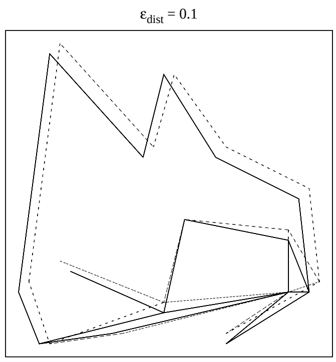

Fig. 4.1: Distance noise. The figure shows an agent's STM with $\varepsilon_{dist} = 0.1$. The drawn through line depicts the correct layout of the graph while the dashed line visualizes the environment as it has been perceived by the agent.

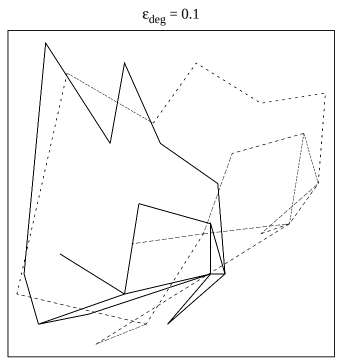

Fig. 4.2: Degree noise. The figure shows the same labyrinth only this time experienced by an agent with $\varepsilon_{deg} = 0.1$. One can see that errors in angle measurement may lead to more severely wrong representations then compared to distance errors.

noise leads to uncertainty concerning one's actual correct position. The greyed area in this figure shows the possibility distribution for the real position of the node, as it should be somewhere near the actually taken, but distorted, measurement.

Another problem is depicted in figure 4.4, showing that measurement errors usually accumulate over time which may lead to severe localization faults. In this case the agent traversed a cyclic graph going from $A \rightarrow B \rightarrow C \rightarrow A$ but because of its noisy sensor readings the agent assumes to be in location D instead of A after finishing the loop. The graph got inconsistent, as it actually contains an invalid extra node D and lacking the right edge leading from $C \rightarrow A$.

However, the here simulated agents are capable of identifying each node they have visited during the current search. Therefore, an agent does not have the problem to believe being at another node than the actual one as long as it has

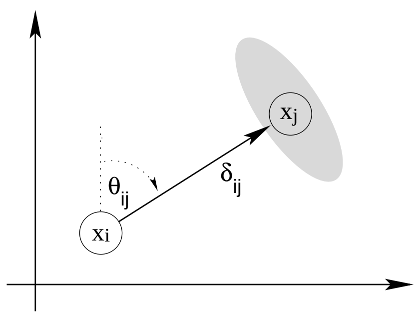

Fig. 4.3: Uncertainty. Distorted measurements lead to uncertainty concerning the target node's position. In this case the node could be anywhere within the greyed area.

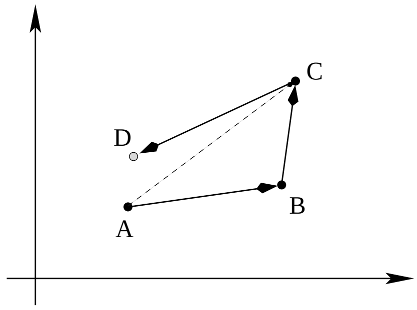

Fig. 4.4: Inconsistency. Returning to previously visited nodes results in inconsistent node locations. Regarding to its odometric data the agent would think to be in a novel node D.

visited that node already, at least once, during that search.

On the contrary, it is able to close those gaps resulted from wrong measurements by just inserting links between nodes it has traversed without paying attention to the distorted measurements. However, this leads to edges e_{ij} that are inconsistent with the actually taken measurements, a problem that will be topic of the next section.

To give an example of how severe these problems are in real robotics, e.g. Thrun et al. (1998) mentioned that they have tried to build maps with their museum tour-guide robots relying purely on odometric data from their wheel encoders. The resulting maps have been highly inaccurate and strongly deformed which has resulted in not being usable at all.

Summarizing this section shows that relying on proprioceptive odometric data, while having to deal with measurement noise, usually results in displaced, skewed or even inconsistent maps. Thus the environmental representation constructed by the agent gets highly inconsistent and might even result in a contradictory representation. The next section will therefore introduce a way to handle such erroneous measurements and will describe how they can be integrated into a map making it as consistent as possible.

4.3 Dealing with Inconsistency

Lacking correct measurements of distances and angles, there is obviously no way to come up with a completely correct map representation. Nevertheless there are ways do construct a representation that lead to a consistent map based on the distorted measurements in which the errors are integrated yielding the most plausible solution.

A good way to integrate such uncertainty is to model probabilistic algorithms trying to minimize inconsistency errors on the measured data. Therefore an approach will be introduced that stands out by excellent results in association with highly efficient time complexity that enables the algorithm to be run in online map building systems, namely a relaxation algorithm developed by Duckett et al. (2002) for SLAM[1].

[1] *Simultaneous Localization and Mapping* is the name of the research area dedicated to those navigational tasks of localization and map building.

This section will introduce this approach as it has been implemented restricted to deal with noisy distance measurements. The approach can be extended to deal with errors on angle measurements as proposed by Frese and Duckett (2003) that will be shortly described afterwards[2].

Relaxation

The principle idea of relaxation may be described in form of a spring analogy (cf. Duckett et al., 2002; Golfarelli et al., 1998). The measured distorted relations between the nodes of a graph are regarded to be springs that are connected to each other. The distance between two adjacent places is the desired length of the spring but as each spring has its own desired length and they are connected at their ends they will expand and contract and eventually reach an equilibrium representing the minimum of an energy function of the graph's springs.

This energy function corresponds to a probabilistic error function which is obtained by evaluating the differences in the location prediction of different relations coming from different nodes. I.e. in figure 4.4 (page 34) node A has different positions regarding to edge e_{BA} ($B \rightarrow A$) compared to e_{CA} ($C \rightarrow A$), resulting in different assumptions concerning A's position. The function describing the overall graph error is:

$$\chi^2(x) = \sum_{e_{ji} \in E} r^{e_{ji}T}(C_{ji}^{-1})r^{e_{ji}}, \tag{4.5}$$

i.e. a sum over all relations $e_{ji} \in E$ with e_{ji} being the edge $x_j \rightarrow x_i : \forall x_i, x_j \in V$ thus a quadratic function of all edges leading to a particular node x_i summed for all the nodes of a graph.
Hereby,

$$r^{e_{ji}} = x_i - x_j - \mu_{ji}, \tag{4.6}$$

and

$$\mu_{ji} = \delta_{ji} \begin{pmatrix} \cos\theta_{ji} \\ \sin\theta_{ji} \end{pmatrix}. \tag{4.7}$$

This means that μ_{ji} is the actually *measured relation* between two points x_i and x_j, as defined in equation 3.8 (page 27), while $x_i - x_j$ corresponds to the relation of these two points obtained from their coordinates in the graph.

This leads to the interpretation of $r^{e_{ji}}$ as the difference of the actual displacement of the two nodes according to their coordinates minus the just measured displacement when traversing the edge $x_j \rightarrow x_i$. Dealing with a perfect measurement would result in $r^{e_{ji}} = 0$ whereas $r^{e_{ji}} \neq 0$ indicates a distorted measurement μ_{ji}.

The C_{ji} of equation 4.5 describes a covariance matrix of the characteristics of an edge forming a model of the actual measurement uncertainty as the measurement error is assumed to be representable as a Gaussian distribution around the measured value μ_{ji} with the covariance of:

$$C_{ji} = \begin{pmatrix} \Delta_{\delta\delta} & \delta_{ji}\Delta_{\alpha\delta} \\ \delta_{ji}\Delta_{\alpha\delta} & \delta_{ji}^2\Delta_{\alpha\alpha} \end{pmatrix}. \tag{4.8}$$

[2] A second approach of dealing with noisy distance data has been implemented that was inspired by hebbian learning. It will, however, not be described hare as the results gained from that approach were not as good as of the relaxation algorithm.

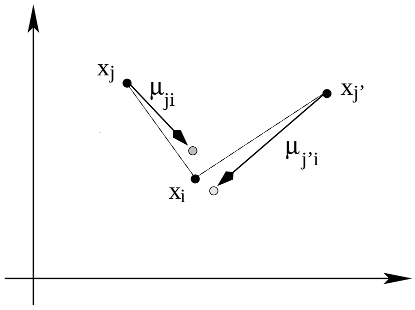

Fig. 4.5: Relaxation. Relaxation of node x_i means to find that position of x_i most consistent to the measured relations μ_{ji} and $\mu j'i$. The grey points represent the position estimates of x_i according to its neighbors while the dotted line shows the edge of the STM graph connecting the nodes.

As one can see, each edge e_{ji} has its own assigned covariance matrix C_{ji} that is dependent on the edge's length δ_{ji} and describes the actual faith in the correctness of this particular edge. The explicit computation of the values $\Delta\delta\delta$, $\Delta\alpha\delta$ and $\Delta\alpha\alpha$ will be described below.

The actual relaxation algorithm now searches for the particular coordinates of a single node x'_i that minimizes the function

$$x'_i = \left(\sum_{e_{ji}} C_{ji}^{-1} \right)^{-1} \sum_{e_{ji}} C_{ji}^{-1}(x_j + \mu_{ji}) \tag{4.9}$$

that can be gained by transforming equation 4.5. As a minimization of this function will lead to those coordinates that are most consistent to the measurements of the current node's position x_i according to all its adjacent nodes x_j and their measured relations μ_{ji}. Put simply, the algorithm tries to "move each node to where it neighbors think it should be"[3] and therefore moving, or *relaxing*, *one* node x_i while considering all its neighbor points x_j as fixed. For a visualization of this see figure 4.5.

To achieve this, the algorithms iteratively performs the following steps:

1. Initialize the covariance matrices C_{ji} for all edges $e_{ji} \in E$. Initially this can be the identity matrix or any other previously acquired matrix for this edge.

2. For all nodes x_i, get estimates x'_{ij} for the position of node x_i dependent on the measured relations from all its neighbor nodes x_j. I.e. calculate the set of points where its neighbors think node x_i should be.

3. Calculate the most probable position for all nodes x_i by this equation:

$$x'_i = \left(\sum_{e_{ji}} C_{ji}^{-1} \right)^{-1} \sum_{e_{ji}} (C_{ji}^{-1} x'_{ij}). \tag{4.10}$$

[3] (cf Duckett et al., 2002)

I.e. the weighted sum of all estimated positions x'_{ij} of x_i yielding the most probable position estimate x'_i.

This is actually equation 4.9 as $x_j + \mu_{ji}$ really is the estimate x'_{ij} as computed in step 2.

4. After having calculated the new estimates for all node positions, update the covariance matrix for all edges appropriately to the used sensor model. In this case the used equations to gain the three different parameters were:

$$\Delta_{\delta\delta} = \sum_i^N \sum_j^{N_j} \frac{1}{NN_j}(\delta'_{ji} - \delta_{ji})^2 \tag{4.11}$$

$$\Delta_{\alpha\delta} = \sum_i^N \sum_j^{N_j} \frac{1}{NN_j}(\delta'_{ji} - \delta_{ji})(\theta'_{ji} - \theta_{ji}) \tag{4.12}$$

$$\Delta_{\alpha\alpha} = \sum_i^N \sum_j^{N_j} \frac{1}{NN_j}(\theta'_{ji} - \theta_{ji})^2 \tag{4.13}$$

Where N_j refers to the number of neighbors of node x_i (i.e. its degree) and δ'_{ji} and θ'_{ji} are those values that represent the relations between the new estimated coordinates x'_i and x'_j while δ_{ji} and θ_{ji} refer to the old values. Therefore one can see that the values of the covariance matrix are composed of the differences gained from the estimated node positions x'_i calculated in equation 4.10 and the really measured relations $(\delta_{ij}, \theta_{ij})$ (see equation 3.7). Remember that C_{ji} is adapted for each edge e_{ji} by being dependent in the distance measurements of that particular edge (see equation 4.8).

5. Resume from step 2 for a certain number of iterations or until the error falls below some pre-defined threshold.

The overall complexity of this algorithm is approximately $O(N)$, with N being the number of nodes, as has been shown by Duckett et al. (2002). Note that this algorithm converges to an optimal solution as the error function 4.5 is quadratic and has therefore only one global minimum. For a complete proof of convergence refer to Duckett et al. (2002).

The results of the relaxation algorithm are exemplary shown in figures 4.6 and 4.7. Remember that the unprocessed original graph representation as acquired by the agent solely based on its odometric sensors, would contain gaps due to the inconsistency of measurements that are not visualized. Nevertheless the figures show the excellent adaptation of the graph compared to the actual environmental layout. Note that the number of iterations to achieve this solution can actually be chosen rather small ($\sim$ 5 iterations have been sufficient for the experiments presented in chapter 5).

Embedding angle distortion - As stated above, the introduced relaxation algorithm is only capable of dealing with distorted distance measurements. The problem with an additional integration of angle distortion is that the error

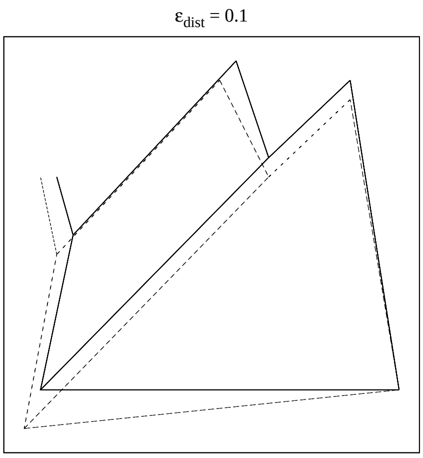

Fig. 4.6: Unrelaxed STM. STM of an agent with $\varepsilon_{dist} = 0.1$. The drawn through line represents the actual layout of the environment whereas the dashed line represents the agents belief of how its surroundings look like.

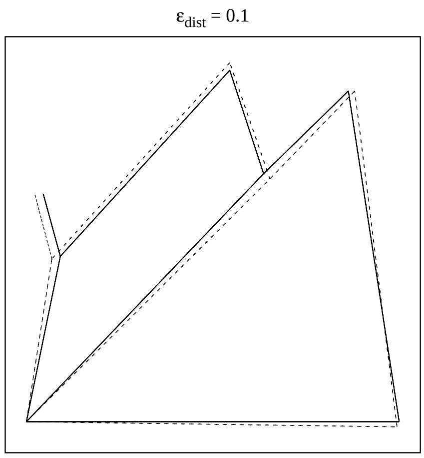

Fig. 4.7: Relaxed STM. STM of an agent with $\varepsilon_{dist} = 0.1$. The figure shows the same STM representation as in figure 4.6 after 5 iterations of relaxation.

function gets non-linear because of the use of rotation matrices resulting in a fairly more complex approach.

The following will contain just a very short overview while referring to the work of Udo Frese and Tom Duckett for further information and an actual implementation of the algorithm.

The main difference to the above algorithm is the extension of a location's coordinates by a third component, describing its global orientation, leading to the definition of a location a as $a = (a_x, a_y, a_\phi)^T$. This leads to an extension of the covariance matrix resulting in a 3×3 matrix for associating probability values to the angle component of a location. Unfortunately the measurement function yielding the relations between globally oriented points are non-linear as they contain *sin* and *cos* functions. Therefore a linearized equation system has to be developed leading to the error function

$$\chi^2(x) = x^T A x - 2x^T b, \tag{4.14}$$

whereas matrix A and vector b result from an integration of the jacobian derivatives of the linearized measurement function and x is a vector containing all nodes of the graph. The resulting minimization problem can be solved by finding an appropriate configuration of x to get a 0 gradient of the partial derivative[4]

$$0 = \frac{\partial(\chi^2(x))}{2\partial x} = \frac{\partial(x^T A x - 2x^T b)}{2\partial x} = Ax - b \tag{4.15}$$

[4] Because of the linearization yielding a quadratic function there is again only a global minimum.

and therefore to find a solution for

$$Ax = b. \tag{4.16}$$

This is done similar to the approach presented above by minimizing this equation system iteratively by so-called *Gauss-Seidel relaxation* yielding the optimal graph-layout most consistent to the taken measurements.

5. EXPERIMENTS

Chapter 4 has introduced several extensions of the approach introduced by Baumann (2003). These extensions have been made to gain a simulation that is meant to deliver realistic results in terms of biological and technical plausibility.

This chapter presents the experiments of how the adjustments that were made fit into the introduced model of knowledge-based learning and online map-building and the conclusions which can be drawn out of the experiments.

As already mentioned before, simulations bear the powerful advantage of being able to test experiments with comparatively low time and money demands. Therefore a couple of diversely designed experiments have been run, investigating the agent's performance on map learning.

The next section will describe the experimental design containing the different setups of environments and agents as well as the realization of the experiments before presenting their actual outcome.

5.1 Experimental Design

As introduced in section 3.2.2, graphs, and therefore the graph-like labyrinth environments, are distinguishable mainly in their *size*, *connectivity* and, in this case, their number of *landmarks*.

The agents themselves may vary in their embedded skills namely their abilities to perceive distances and angles as well as their general handicaps (see chapter 4).

Diverse combinations of all these values were set up to extensively analyze the agent's performances yielding a total of 16 different types of labyrinths for testing 29 different types of agents. The settings chosen will now be described in detail[1].

5.1.1 Environmental Settings

The actual chosen environmental conditions and therefore labyrinth layouts chosen consist of all permutations of the following parameter settings:

Number of nodes:	$N = 20$ and $N = 40$.
Number of landmarks:	$l = 1$ and $l = 5$.
Connectivity:	0%, 25%, 50%, 100%,

where a connectivity of 0% describes a spanning tree and 100% a fully connected graph.

[1] For the complete list of all settings refer to appendix A.1.

Note that a number of landmarks with $l = 1$ in fact equals a labyrinth without landmarks as the target location is implemented to be a landmark since it requires to be globally identifiable. Therefore $l = 1$ is the smallest number of landmarks possible.

5.1.2 *Agent Settings*

To guarantee generalizable results, diverse agents with the settings described in this section have been tested.

The perceptional angle error ε_{deg}, the perceptional distance error ε_{dist} as well as the general perceptional error ε_{rec} have each been set to the values

$$\varepsilon = 0\%, 2.5\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%.$$

Besides agents that have senses with a perceptional error of one type, agents were tested whose sensors produce noise values on both ε_{deg} and ε_{dist}. This yields to a total of 29 different agents.

Furthermore, these agents have been tested using different strategies for comparing nodes of their STM and LTM. The functions that have been used will be described in detail in section 5.3.2 as they show a high impact on the agent's actual performance.

5.1.3 ***Measured Data***

The experiments are meant to deliver an evaluation of the performance of different agents in a representative selection of labyrinths to get an insight of the agent's actual abilities and its requirements to perform the task of knowledge-based map-learning. For an evaluation of the experiments the following context measurements have been applied:

- *Number of steps* - Information of the agent's performance can be gained by counting the number of steps made until the goal is reached. A step is hereby defined as the traverse of one edge e_{ij}. Traversing the same edge more than once, or in different directions, will be each time counted as a step.

- *Shortest path* - The shortest path through the labyrinth will be measured for the purpose of comparing the agent's step number to the least needed number of steps.

- *Failed search attempts* - An agent fails to find its target due to a falsely identified location, i.e. an agent believes to be at a place where it is not, prohibiting the continuation of the search.

- *Time of merging* - This time is the actual number of steps before the first *merge attempt*. Hereby, a merge attempt means that the agent has recognized a node and combines the local graph of its STM wit a previously acquired graph stored in its LTM.

- *Grade of exploration* - The exploration describes how much of the actual labyrinth has, at a whole, already been explored and therefore should be known to the agent from this or previous search attempts.

5.2 Experimental Realization

Each of the above defined agents was run in each of the different environments. Or to be more precise, each agent is tested in 40 equivalent labyrinths of one class performing five searches in all of them before continuing with the next labyrinth class. Being an *equivalent labyrinth* means having the same parameters for number of nodes, connectivity and landmarks but different topological layouts.

The agents have to perform the task specified in section 3.1 i.e. to find a goal in the environment. As the number of steps needed to find such a goal is highly dependent on the labyrinth's layout, even for labyrinths with the same settings, and to achieve a statistically relevant statement the agents have been tested in the following manner:

1. Generate the next specified agent.
2. Generate the next specified labyrinth.
3. Do five trials on that particular labyrinth. I.e. perform five searches.
4. Reset the agent's memory and repeat step 3 with 40 labyrinths of equivalent settings.
5. If there are more labyrinths specified continue with step 2.
6. If there are more agents specified continue with step 1.

Note, that for step 3 two different setups have been tested. The first was to perform the five search trials with fixed starting and goal points, the second with altering source and target.

The complete set of experiments has been conducted employing the different comparison strategies mentioned in section 5.1.2.

The measured data as described in section 5.1.3 is hereby summed for each agent and afterwards averaged over the number of equivalent labyrinths and labyrinth classes and stored for the different trials.

Therefore, the simulation has been equipped with a batch system capable of automatically running wide varieties of tests with adjustable parameters for the needed environmental settings and different agents.

5.3 Results

As stated in section 5.2, each agent was tested in 640 labyrinths of 16 different classes performing a total of 3200 search trials. These have been performed in two main runs, one with altering initial points and targets the second one with fixed start and goal locations while employing different strategies concerning the comparison of nodes.

This section will present the results gained from the experiments differentiating them by the different fields of errors, ε_{deg}, ε_{dist} and ε_{rec}, and therefore by the different types of agents, but firstly presenting the results for the perfectly skilled agent, i.e. the agent with undistorted sensors.

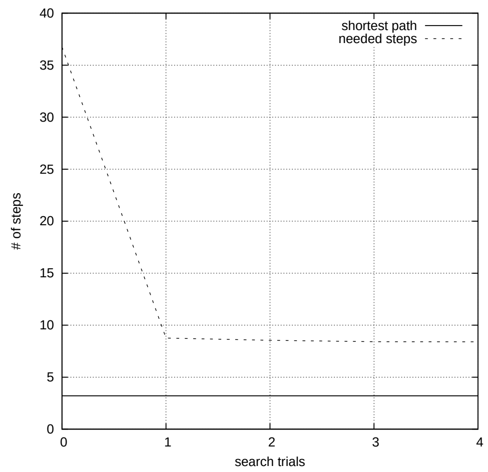

Fig. 5.1: Perfectly skilled agent. Fixed source and target locations.

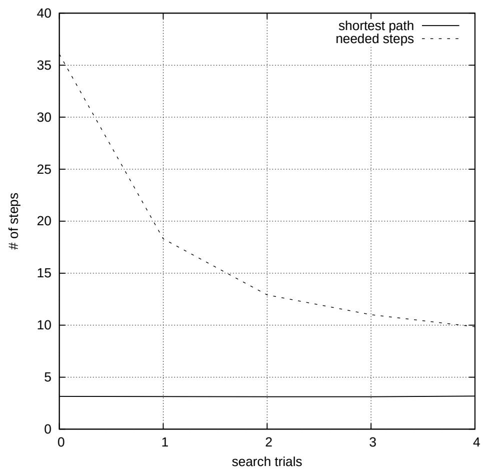

Fig. 5.2: Perfectly skilled agent. Altering start and goal locations.

5.3.1 Perfect Measurements

The experiments run with perfectly skilled agents approved the results of Baumann (2003). Figures 5.1 and 5.2 show the agent's average performance on all the labyrinths. The first trial, with the number 0, does always consist of a depth-first search on the graph without the availability of prior knowledge. The trials 1 - 4 on the contrary are those in which the agent was able to use the knowledge acquired in previous search attempts. As one can see in the figure, the mean number of steps needed by the perfectly skilled agent drops drastically, by approximately 2/3 to 3/4, compared to the depth-first search.

One should keep in mind that the way found to the goal is not likely to be the shortest path. This is because it would imply a correct path, being randomly chosen by the depth-first search strategy at the beginning, that will later on be recognized and followed by the agent to the goal location.

As the shortest path repeatedly turned out to be in the area of 2.5 to 3.5 steps and will therefore not be mentioned explicitly in the following.

For the different kinds of runs the following results have been achieved:

Fixed start and goal location - In figure 5.1 one can see that the run with the same start and goal location shows faster learning results without improving significantly in successive trials.

The performance gain encountered is approximately 75% compared to a depth-first search yielding an average path length of only ~ 5 steps more than the shortest path would need. A depth-first search on the contrary would need an additional 37 steps compared to the shortest path.

This can be explained by taking into account that a fixed source location has only a limited number of adjacent nodes yielding only a small set of nodes that form the possible start of the search and therefore simplifying the early self-localization by the agent's place recognition ability.

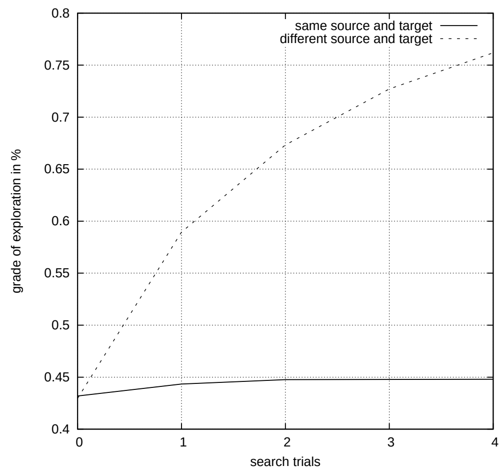

Fig. 5.3: Perfectly skilled agent. Percentage of exploration as measured in the two different kinds of runs.

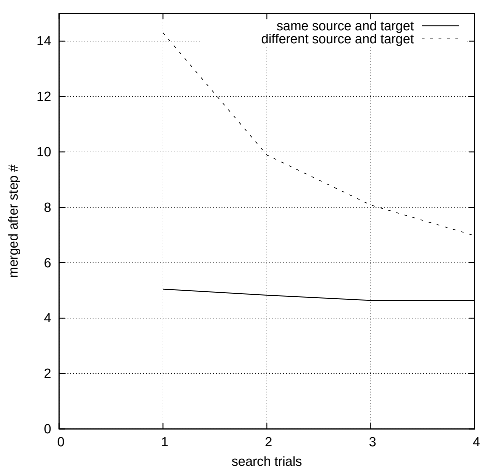

Fig. 5.4: Perfectly skilled agent. Amount of steps made until merging of STM and LTM in the different runs. Note that as trail 0 represents the trial without prior knowledge nothing could have been merged.

Altering start and goal location - The run with altering source and target points on the contrary shows rising performance for each trial as seen in figure 5.2. This is presumably because the increasing exploration of the environment yields a more and more reliable place recognition.

Conclusion - The results are confirmed by the measurement data showing the grade of exploration and the number of steps needed to merge the memory for the first time. Figure 5.3 shows the labyrinth exploration in the two runs with and without altering source and target whereas figure 5.4 shows the number of steps needed until the current node has been recognized in the agent's LTM for the first time.

One can see the direct correlation between the agent's performance of needed steps (figures 5.1 and 5.2) and the increasing exploration of the labyrinths. The agent with altering source and target succeeds the earlier the more of the labyrinth it has explored. On the other side the agent with the fixed start and goal locations that does not show any increase in labyrinth exploration is bounded to a relatively static performance over all successive trials once it manages to localize itself.

Though showing a seemingly better performance in terms of needed steps, it is most likely that the agent with fixed goal and target locations would proof severely worse when encountering a new source point as compared to the globally experienced agent.

Note that the different number of landmarks did not show any unexpected results as the merging just takes place the earlier the more landmarks are available.

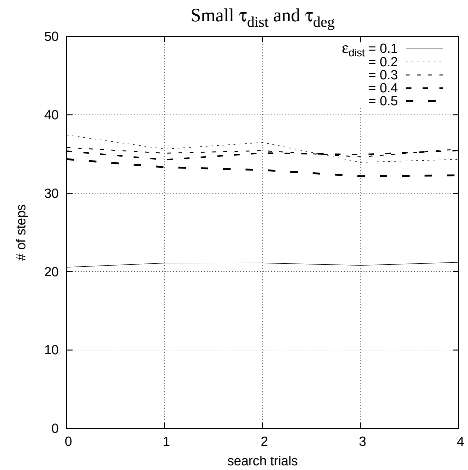

Fig. 5.5: Distance distortion. Agent with distorted distance perception connected with small tolerance values.

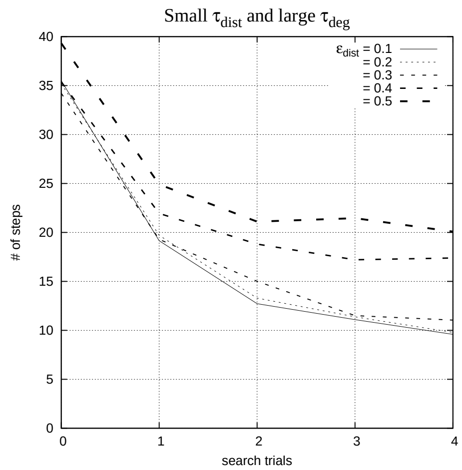

Fig. 5.6: Distance distortion. Agent with distorted distance perception provided a more tolerant angle comparison.

As a final remark, the failure rate of the agents was extremely low with less than 6% in the case of the agent with changing sources and targets and no failures at all in the other case.

Since the correlation between labyrinth exploration and the time of the first merge attempt was consistent with all types of agents this will not be shown explicitly for these different types.

The results in comparing the two different kinds of altering and non-altering source and target points showed as well to be consistent with all classes of agents dealt with. The following sections will therefore not explicitly concern this difference but instead will focus on the results obtained by running experiments with the differently skilled agents regarding the trials of changing start and target locations.

5.3.2 Distance Distortion

This section will present the results gained from running the experiments with agents that had an impaired sensory distance perception. Hereby, the results will be split up with respect to the different comparison strategies employed by the agents. These strategies are mainly distinguishable in their tolerance of accepting two angles or distances as being equal and therefore in the probability of perceiving a node of the agents STM as being equal to a node of its LTM.

Three different tolerance settings have been tested, namely small tolerance values, semi-large values that show greater tolerance in the case of angle comparison and large tolerance settings being extremely tolerant concerning both angle and distance comparison.

Small tolerance values - The following two equations were used to define the

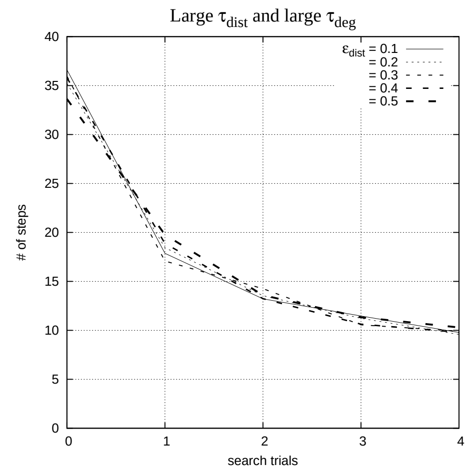

Fig. 5.7: Distance distortion. Agent with distorted distance perception with large tolerance settings.

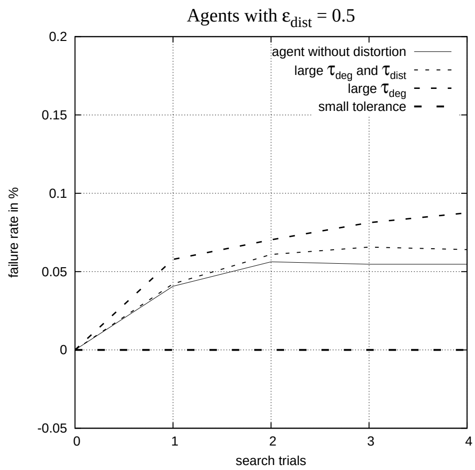

Fig. 5.8: Distance distortion. Percentage of failed runs using the different tolerance settings.

tolerance of distance and angle comparison:

$$\tau_{dist} = \frac{\varepsilon_{dist} * \delta_{ij} + \varepsilon_{dist} * \delta'_{ij}}{2}, \tag{5.1}$$

$$\tau_{deg} = \frac{\varepsilon_{deg} * 2\pi}{2}. \tag{5.2}$$

These tolerance settings, however, proofed to yield a comparison strategy much too restricted as they lead to the performance illustrated in figure 5.5. It shows a selection of five differently skilled agents with ε_{dist} of 0.1 to 0.5.

Compared to the perfect agent's performance (figure 5.2) the other agents were not able to show comparably good results and, more importantly, none of them showed any success of learning as none of the number of step curves decreases with an increasing number of trials.

The agents were clearly unable to make any use of their previously learned knowledge of the environment and therefore were bound to the number of steps the depth-first search needed that is depicted in the column of trial number 0.

Large tolerance on angle comparison - The semi-tolerant settings tested consist of a comparison strategy using this tolerance equation for angle comparison:

$$\tau_{deg} = \varepsilon_{deg} * 2\pi \tag{5.3}$$

that is actually the same as in equation 4.4 on page 32.

Furthermore, a minimum distortion of angle measurements of $\varepsilon_{deg} = 0.2$ was assumed leading to the results illustrated in figure 5.6.

As one can see, the results were significantly better than compared to the smaller chosen tolerance of equation 5.2. In fact, the results for agents with $\varepsilon_{dist} \leq 0.3$ showed performances equal to the agent with none distorted sensors. Only a distortion of $\varepsilon_{dist} > 0.3$ showed noticeable shortcomings in the number of steps needed to reach the goal location.

Importantly, figure 5.6 shows the successful usage of the knowledge-based map learning system, the agents are equipped with, as the number of steps needed decreases significantly with an increasing number of trials.

Large tolerance on angle and distance comparison - The next tested comparison strategy loosens the requirements of matching angles and distances even more using the formula

$$\tau_{dist} = \varepsilon_{dist} * \delta_{ij} + \varepsilon_{dist} * \delta'_{ij} \qquad (5.4)$$

being the same as in 4.3 (page 31). This time the minimum ε_{deg} and ε_{dist} were set to 0.5.

Figure 5.7 shows the results gained by these new assumptions for τ_{dist} and τ_{deg}. This experiment shows for all agents with distorted distance measurements of up to $\varepsilon_{dist} = 0.5$ an outstanding learning success that is, indeed, comparable and even equal to the performance achieved by the perfectly skilled agent.

These results suggest that the comparison strategy, and therefore the tolerance values, are independent of the actual distortional errors as all the different agents show comparable learning results.

It remains to investigate how these tolerance settings affect the failure rate of those agents.

Failure rate - Figure 5.8 shows the impact of the different tolerance settings on the failure rate of the agents. The figure is restricted to agents with the maximum noise of $\varepsilon_{dist} = 0.5$ as they show the most distinct results in the different experiments concerning the tolerance settings.

As one can see, the agent that have used small tolerance values of equations 5.1 and 5.2 did not fail in any of the runs it performed. Comparing this low rate of failure with its performance (figure 5.5), however, shows that this is due to the fact of that the memory is not merged at all. This is indicated by the fact that the agent never exceeds the result of the depth-first search.

The second agent equipped with the improved τ_{deg} of equation 5.3 as well as the agent additionally using τ_{dist} as specified in equation 5.4 showed an increasing failure rate of up to approximately 9%.

Nevertheless, this failure rate is only slightly higher than the failure rate of even the perfectly skilled agent (also shown in figure 5.8). Furthermore a failure rate of less than 10% seems to be acceptable regarding a performance gain of nearly 75% in terms of needed steps to reach the goal.

Conclusion - To be able to cope with large measurement noise on distances, agents require a very tolerant comparison strategy to show performances on knowledge-based navigation that are comparable or even equal to the perfectly skilled agent. However, given these tolerance settings, no significant impact in terms of failure rate can be observed.

This behavior can be explained by referring to two different kinds of errors that can happen during the labyrinth exploration and map-building processes namely *false positive matching* and *false negative matching.*

- False positives are those errors letting the agent have the impression to deal with a known node although it is actually another one.

  In the scope of this simulation a false positive would mean the erroneous merging of the agent's STM and LTM leading to an incorrect map representation and therefore, most probably, to a failed search as the agent fails already if it tries to use an edge that is not present in the labyrinth.
- False negatives are those errors letting the agent reject a node because of believing that it is a different one, i.e. the agent does not recognize it though it should.

  But on the contrary to the false positives, this error would not result in a failed search as the map, the agent builds in its STM, still remains valid. The only impact, this error shows, is on the agent's performance as a merging of its memory and therefore the faster reaching of the goal is being delayed.

The discrimination of these two classes of errors explains the different outcomes of the experiments:

- The agents that were using the small tolerance settings of equations 5.1 and 5.2 showed a failure rate of 0%, i.e. they did not encounter the problem of false positives. Their bad performance regarding the needed steps, however, is probably due to a high amount of false negatives. The agents rejected all the points they visited because of the tolerance being much to narrow.
- A higher tolerance used in the second experiment resulted in a decreasing amount of false negatives leading, for the first time, to the ability of merging the agent's STM and LTM. Therefore, a learning result became visible connected with an increase of the failure rate due to the fact that false positives became possible as well.
- The agents equipped with the comparison strategy using the highest tolerance settings were able to show supreme results, which are comparable to the modelled agent capable of undistorted perception while still having a failure rate of less than 10%. This shows that a tolerance setting has been found in which the number of false negatives is reduced to a minimum while still constraining the amount of false positives to an acceptable number yielding a simulation system highly robust to noisy measurements at least on the encountered set of labyrinths.

As a final remark it is to state that the relaxation algorithm, presented in section 4.3 did not show any effect concerning the agent's performance. Though leading to visibly good results as seen in figure 4.7 (page 4.7), the comparison

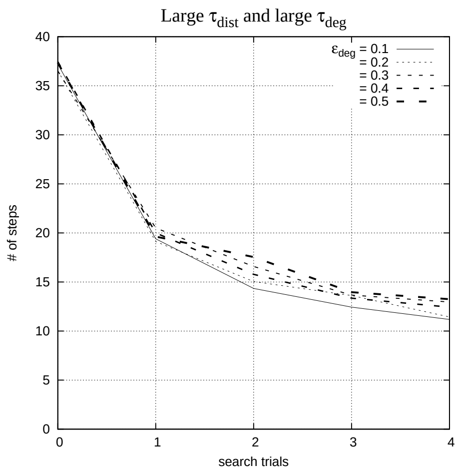

Fig. 5.9: Angle distortion. Agent with distorted angle perception and large tolerance settings.

strategies, using tolerance values that high, resulted in agent performances as good as without the relaxation algorithm. Because of having the additional benefit of yielding consistent map representations and as the resulting graph has proven to be much closer to the original environmental layout, the algorithm has nevertheless been used. Even though it does not increase the agent's performance at least on this set of environments.

5.3.3 Angle Distortion

The experiments concerning distorted angle perception showed very similar results as already encountered in the case of distorted distance perception. The results were again best with using the large tolerance settings leading this time to the performance depicted in figure 5.9 which shows the conform decrease of needed steps for all types of agents up to $\varepsilon_{deg} = 0.5$.

5.3.4 Mixed Angle and Distance Distortion

The connection of distorted angle and distance measurements showed again similar results to the cases described above. This can be seen in figures 5.10 and 5.11 depicting the number of steps needed by the agents while using only the large τ_{deg} in the first case and both the large distance and angle tolerance in the second case.

Moreover, figures 5.12 and 5.13 show, for the case with the semi-large tolerance, the consequences of the noise values on the number of steps needed to first merge the memory and on the failure rate as well. As one can see here, both the number of steps before the first merge attempt as well as the failure rate increase uniformly to the height of ε_{deg} and ε_{dist}.

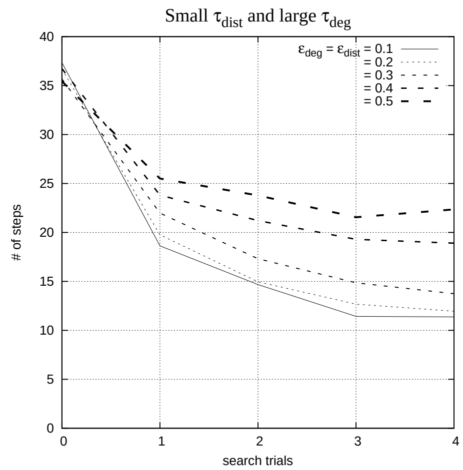

Fig. 5.10: Mixed Angle and Distance Distortion. Number of steps needed with the semi-large tolerance settings.

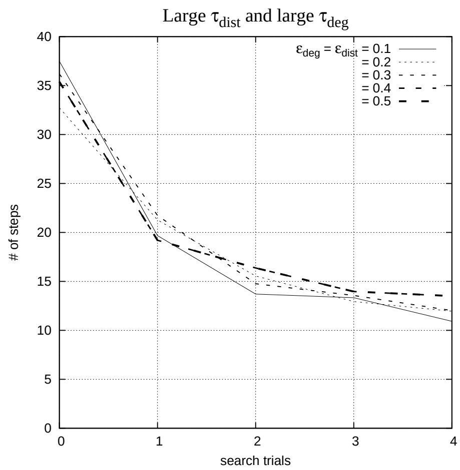

Fig. 5.11: Mixed Angle and Distance Distortion. Number of steps needed with the large tolerance settings.

5.3.5 General Perceptional Error

The general perceptional error was defined in 4.1 as a sort of cognitive error that drives the agent to experience an angle or distance contradictory to its original belief. I.e., if the agents would originally accept that angle or distance to be equal to the one it compares it with, this error would yield to a rejection of that angle or distance and vice versa.

The performances of agents under the influence of this error is illustrated in figure 5.14 showing their steps needed to reach the goal as well as in figure 5.15 that shows the percentage of failed runs for the different agents. Both of the figures are based on data resulting from experiments containing the large tolerance values described above. The agent's performances are, however, worse than in the cases described above as this general error ε_{rec} apparently produces more false negatives compared to the approaches above.

This becomes clear by elucidating what is needed to produce the two classes of errors as

- a false negative only needs the rejection of one true comparison of an angle or distance connected to the node itself or to one of its neighbors as far as the neighbor nodes are concerned by matching the memories.
- A false positive on the contrary results only from the acceptance of *all* compared values. Only one correctly rejected value would reject the whole node. Thus, a false positive is much less likely to happen compared to the appearance of a false negative.

As figure 5.15 shows, the percentage of failed runs still lies beneath 10%. This is no significant increase of false positive errors whereas figure 5.14 shows an increase of false negatives that impair, at least for the higher values of ε_{rec}, noticeably the general performance of those agents.

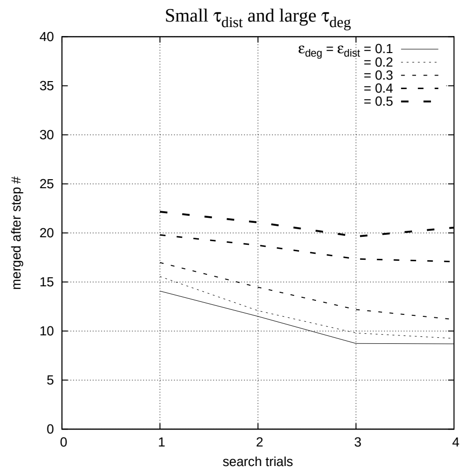

Fig. 5.12: Mixed Angle and Distance Distortion. The figure illustrates the connection of a rising distortion value to the time of the first merge attempt.

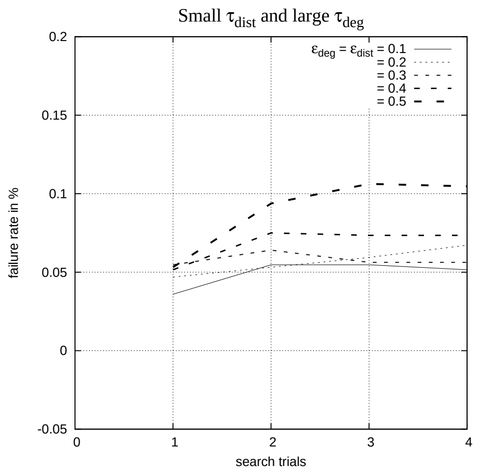

Fig. 5.13: Mixed Angle and Distance Distortion. The figure illustrates the connection of a rising distortion value to the failure rate of an agent.

To draw a conclusion, ε_{rec} shows an impact on the appearance of false negatives an therefore on the agent's performance. It shows, however, no significant effect on false positives and the agent's failure rate respectively.

5.4 Conclusion

As presented in this chapter, the extensions made to the simulation and especially their impact on agent performance have been extensively tested showing good results regarding the robustness of the system.

The findings for the overall performance of the system confirmed the results stated by Baumann (2003) as it has been shown that the number of needed steps to find a goal drops significantly by up to 75% while obtaining low failure rates of, in the worst case, approximately 10%.

Furthermore a direct correlation between the grade of exploration and the time an agent firstly recognizes a location within its memory has been shown explaining the actual characteristics of the agent's learning curve that decreases with an increasing grade of exploration.

Running experiments with agent's under the influence of distorted perception and noisy measurements respectively showed that the parameters crucial for the agent's performance are not the actual distortion on the sensor readings but, instead, the tolerance values τ_{dist} and τ_{deg} as defined in section 4.1.

Given the proper choice of these tolerant comparison settings, the results gained for the perfectly skilled agents, as presented in section 5.3.1, have been achieved with agents of all the different distortion classes. None of them showed divergences severe enough to impair the performance not even with perceptual distortions of up to 50% which is, in fact, an error that should rarely occur at least in real world robotics.

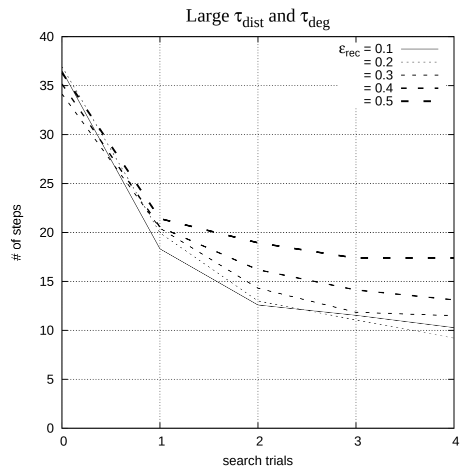

Fig. 5.14: General Perceptional Error. Needed number of steps.

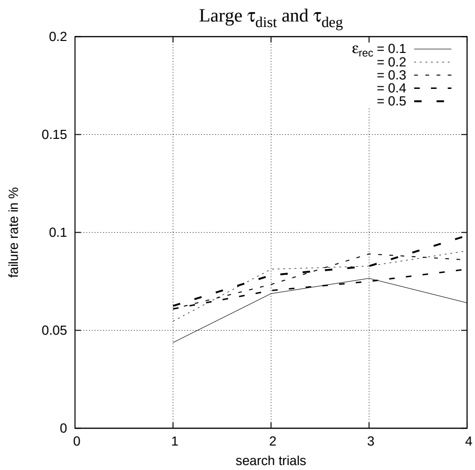

Fig. 5.15: General Perceptional Error. Percentage of failed runs.

The general error, the agents were tested with, resulted in a more frequent appearance of false negatives showing an impact on the agent's performance while not affecting their general failure rate significantly.

As these results are, generally speaking, averaged over a great diversity of different labyrinths, an evaluation of the system's performance regarding so-called corner cases (cf. Baumann, 2003) might be of interest. E.g. environments containing very similar structures in distinct locations might be an interesting challenge for this approach as environments that are more likely to produce false positive errors might show the need for a comparison strategy based on smaller tolerance values than the presented. Given these environments might as well show the need of additional map processing algorithms like the relaxation approach or its extension of the embedded angle version.

The simulation showed, nonetheless, extremely robust results insusceptible even to very high measurement noise for the tested, already very broad, field of environments.

6. CONCLUSION AND OUTLOOK

Throughout the thesis, real life navigation has been related to the concepts and techniques of biomimetic robotics. Hereby, a broad overview of animal and human navigation has been given to provide insights into diverse systems capable of dealing with navigational tasks as well as into the structures enabling the necessary functionality.

These systems and structures have been compared to the field of robotics emphasizing their affinity to the methods and devices encountered therein.

In connection to this the structure of a cognitive spatial map has been elaborated as the main structure concerning the acquisition of environmental spatial knowledge on the one hand and the computational functionality necessary for the task of way-finding on the other hand.

This structure inspired, above all, the usage of topological and symbolic maps that represent an environment combining it with the computational power of certain algorithms to maintain it and to work on that representation.

A simulation that is closely related to biological systems of navigation and that is based on such a representation has been introduced. This representation was embedded into a knowledge-based structure based upon the concepts of short- and long-term memory.

This approach, simulating navigation of agents, has been extended under consideration of the biological systems introduced. Therefore agents have been equipped with varying sensory skills by simulating both measurement noise on the agent's perception as well as a general error to simulate general impairments.

The system has been further extended to deal with some of the problems that resulted from these previous extensions, namely a relaxation algorithm for coping with distorted distance measurements.

Moreover, evaluation criteria have been developed to test the performance of a diverse variety of agents fulfilling navigational tasks within various simulated environments.

The experiments conducted have shown an outstanding robustness of the system in terms of dealing with high sensor distortion while proving that the right choice of the comparison strategy is crucial for the agent's performance.

Such a comparison strategy has been elaborated and tested showing that it leads to performances equal to those of the non-impaired agents.

These results should yield a good starting point for further investigations:

The experiments showed that an investigation of so-called corner cases might lead to interesting results as a stronger appearance of false positive matching errors might be triggered by environments that show layouts with similarities in different parts. The more frequent occurrence of this type of error would most probably show an impact on the agent's failure rate demanding comparison strategies that are even more elaborated.

Such an extension might, as well, bring up the need for an additional processing of the map. This could be achieved e.g. by improving the presented relaxation approach through the introduction of the more general version capable of handling angle distortion as well. Related to this the presented approach can be extended to deal with measurement sets that are actually able to store collections of measurements for traversed edges yielding a map statistically even more precise.

According to the agent's exploration strategy, its performance might be further improved by changing its path-finding behavior. Up to now the agent follows a path to the goal, once its position is known, that is optimal regarding the agent's knowledge. The environment may, however, provide a shorter path that is unknown to the agent as the environment is usually not fully explored. Therefore it would probably yield an even better result if the agent develops an exploration strategy that actively searches for shortcuts. E.g. two adjacent nodes that have exits leading in a direction pointing to each other might most probably be connected by an edge and could therefore be used by the agent as a shortcut.

The presented approach of simulating knowledge-based learning of symbolic maps provides furthermore a good basis for ongoing extensions concerning its biological plausibility. For instance agents, as they already have short- and long-term memory, could easily be extended to support other memory related functions like for instance the ability to forget certain aspects of previously learned information. An agent might e.g. forget values or even nodes over time if it does not visit them for a certain amount of trials.

Another idea would be an extension to dynamic environments that might show the need of adapting structures of knowledge representation as the knowledge base would have to be extended to deal with changes in the agent's environment in order still provide the functionality of a robust place recognition and navigation system.

This actually might benefit from an approach in which an agent is able to recognize its own faults. The ability of dealing with incorrectly merged maps would imply a strategy of splitting them again in order to try to repair them which would result in a strategy even more robust to false positives.

On the whole the presented simulation shows many points for possible extensions and further investigations both on the technical side in terms of algorithmic enlargement as well as biologically inspired extensions of the biomimetic simulation perspective.

7. ACKNOWLEDGEMENTS

I would like to thank my supervisor Ute Schmid for introducing me to the topic of this thesis and Roland Hafner for his support during its development as well as Udo Frese for his patience and support on the relaxation algorithm.

Furthermore I would like to thank my family and my friends especially Tobias Kringe for taking over a lot of work that had to be done besides this thesis and Christine Carl for giving excellent remarks and helpful suggestions concerning this work.

Osnabrück, Germany   Christopher Lörken
July 26, 2004

APPENDIX

A. IMPLEMENTATION DETAILS

The main extensions made to the simulation presented by Baumann (2003) are embedded in the following classes[1]:

***AdvancedAgent* -** This class provides the necessary extensions to deal with different levels of sensory distortion. The properties of the agents specify its ε_{deg}, ε_{dist} and ε_{rec}.

The agents have been equipped with the ability to track their own position, presented as its distorted coordinates, as it is now independent of its actual position within the labyrinth.

Furthermore it is the task of the agent to specify its comparison strategy. That is why the tolerant comparison of angle and distance values is embedded in this class.

Additionally the agent has the ability to count certain mistakes it makes like e.g. the false negative matching of an angle.

***AdvancedNode* -** The nodes of the labyrinth themselves take care of the actual distortion of their properties with regard to the abilities of the current agent as well as their comparison regarding to the agent's comparison strategy. All the methods needed for the consistent setup and alteration of a labyrinth have therefore been adjusted to generate the according counterparts of their distorted values and connections. These methods contain for instance methods connected to the initial construction of the node as the addition of exits and connections as well as those methods needed to maintain the network like rotation and translation.

Therefore, they have been extended to maintain a second set of coordinates and relations regarding their distorted locations altering them accordingly to the changes performed on their real values.

They hereby store their original relations for the purpose of relaxation while additionally maintaining a representation of their estimated coordinates.

They furthermore provide the necessary tools for propagating changes that affect the whole node network like for instance the update of their distorted coordinates to the values estimated by the relaxation requiring the complete reevaluation of the relations between connected nodes.

Moreover, AdvancedNodes call the relaxation algorithm to update the graph each time the agent closes a loop of the environment.

[1] In this context only the extensions made will be introduced. Refer to Baumann (2003) for a description of the basic functionality of the simulation environment .

***AdvancedMemory* -** The memory class has been extended to support memories composed of AdvancedNodes. Hereby especially the merging and transformation of the memories has been extended to provide consistency concerning the distorted values of the nodes forming the memory.

***AdvancedLabyrinth* -** The labyrinth class mainly had to be extended to support labyrinths consisting of AdvancedNodes.

***SimpleRelaxation* -** This class provides the implementation of the algorithm presented in section 4.3. It therefore contains the necessary methods and fields for maintaining position estimates stored in the nodes as well as of assigning covariance matrices to all edges and to update these values according to the iterative approach of relaxation. It furthermore contains some measurement methods to get the according position estimates and their relations to each other and additionally the functionality of aligning the estimated positions to the original origin that has been the agent's start location. This method is necessary as even that position is being adjusted by the relaxation what would result in a displaced relaxed graph that might even show the characteristic of a static drift resulting from the relaxation.

Note that this class uses the JAMA matrix package, that can be found at Hicklin et al. (1999), to support the calculations performed.

***MapTransform* -** This class contains the second approach of map transformation that has not been described in this thesis as it did not show results comparable to the relaxation method. The approach was inspired by hebbian learning while assuming a fixed weighting formula to associate confidence probabilities to different relations between points. Since this confidence rating is not being adjusted over time, the results gained were considerably worse than compared to the relaxation of section 4.3.

Moreover, the following classes contain extensions coping with the changes described above:

***Batch* -** See appendix A.1.

***AgentSettingsDialog* -** This class provides the dialog frame allowing the user to change the settings of the current agent.

***AgentPanel* -** The class specifies how the agent settings are displayed in the GUI.

***Trace* -** This class takes care of visualizing the values the AdvancedAgent counts during its runtime like e.g. the false positive and negative matchings.

***RoutenGraph* -** The graph representing the agent's STM has been extended to show the agent's belief of its environmental layout as well. This means that the graph based on the distorted values experienced by the agent is being visualized on top of the actual environmental layout as it would have been perceived by the perfectly skilled agent. The actual methods needed to paint this visualization are provided by the AdvancedNodes themselves.

***AdvancedNode* -** In addition to the extensions described above, this class provides some extra output tools. It is supported to write the agent's STM graph to a file generating an according plot file in Gnuplot format with the additional possibility of automatically generating encapsulated postscript representations of the graph consisting of the undistorted as well as the distorted layouts.

***MainWindow* -** Of course the GUI's main window has been adjusted to support the actions necessary for the extended parts of the program.

A.1 Batch System

The Batch class provides the utility necessary to conduct large scale experiments. The experiments proceed in the way presented in section 5.2 while the different parameters have been described in sections 5.1.1 and 5.1.2. The experiments can be triggered to run in batch mode that disables the GUI for faster performance.

The parameters are specified in the files *agents.txt* and *labyrinths.txt* that are shown below. The batch system extensively logs and stores results consisting of the average performance values of the measured data as described in 5.1.3 for each agent as well as the different agent performances for each class of labyrinths.

The batch system is called from the settings menu of the GUI and will automatically run the agent and labyrinth text files located in the relative batch folder.

These files are specified as follows. Note that they show the specifications that lead to the results of the experiments described in chapter 5.

A.1.1 Agent Batch Settings

```
# The first number is the total amount of specified agents.
#
# The second number specifies if the program is run in batch
# mode (disabled visualization but better performance).
#       1 = batch mode, 0 != batch mode
#
# each row contains the following settings:
#   - noise on angle perception (e_deg)
#   - noise on distance perception (e_dist)
#   - general impariment (e_rec)
29
0
#agents
0.0   0.0   0.0
0.0   0.025 0.0
0.0   0.05  0.0
0.0   0.1   0.0
0.0   0.2   0.0
0.0   0.3   0.0
0.0   0.4   0.0
0.0   0.5   0.0
```

```
0.025 0.0   0.0
0.05  0.0   0.0
0.1   0.0   0.0
0.2   0.0   0.0
0.3   0.0   0.0
0.4   0.0   0.0
0.5   0.0   0.0
0.025 0.025 0.0
0.05  0.05  0.0
0.1   0.1   0.0
0.2   0.2   0.0
0.3   0.3   0.0
0.4   0.4   0.0
0.5   0.5   0.0
0.0   0.0   0.025
0.0   0.0   0.05
0.0   0.0   0.1
0.0   0.0   0.2
0.0   0.0   0.3
0.0   0.0   0.4
0.0   0.0   0.5
```

A.1.2 Labyrinth Batch Settings

Note that the connectivity values are represented differently in this context. The values of the batch file have the form $100\% - connectivity * 100$

```
#Labyrinth setting batch file
#
#
# - First row: Number of specified labyrinths.
#
# - second row:
#        experiment specific settings:
#    - number of trials in one and the same labyrinth
#    - number of equivalent labyrinths
#               (i.e.  - same settings,
#                      - different labyrinth,
#                      - same number of trials)
# each row:
# Labyrinthspecific settings:
#        - nodes,
#        - Number of Landmarks,
#        - 100 - connectivity,
#
# Annotation lines have to _start_ with '#'.
# Please avoid extra tokens and empty lines.
# Tokennumber for a valid Lab has to be 3
16
5 40
```

```
# Labyrinths:
20 1 0
20 1 25
20 1 50
20 1 100
40 1 0
40 1 25
40 1 50
40 1 100
20 5 0
20 5 25
20 5 50
20 5 100
40 5 0
40 5 25
40 5 50
40 5 100
```

B. SUPPLEMENTARY DISK

The supplementary disk attached to this thesis contains the program in the version that has been used to gather the results described in chapter 5. The main class that has to be called in order to start the program is:

> $gui.MainWindow$.

The most of the changed files as described in appendix A are located in the package

> *advaced*.

The batch class and the *agents.txt* and *labyrinths.txt* are located in package

> *batch*.

The results of running experiments using the batch system will be stored in the *batch* folder as well.

Additionally the results gained from the experiments that lead to the values as presented in chapter 5 are located in folder

> $results$.

BIBLIOGRAPHY

Moritz Baumann. Navigation in merkmalsarmen Umwelten. Master's thesis, TU Berlin, 2003.

A. T. D. Bennett. Do animals have cognitive maps? In *Journal of Experimental Biology, 199*, pages 219–224. 1996.

Aaron P. Blaisdell and Robert G. Cook. Integration of spatial maps in pigeons. 2004.

Marc Bolduc, Eric Bourque, Gregory Dudek, Nicholas Roy, and Robert Sim. Autonomous exploration: An integrated systems approach. In *AAAI/IAAI*, pages 779–780, 1997.

Michael D. Breed. *Animal Behavior - An Online Textbook*. 2001.

W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.

B. Claus, K. Eyferth, C. Gips, R. Hrnig, U. Schmid, S. Wiebrock, and F. Wysotzki. Reference frames for spatial inference in text understanding. In C. Freksa, C. Habel, and K.F. Wender, editors, *Spatial Cognition - An interdisciplinary approach to representing and processing spatial knowledge*, pages 241–266. Berlin: Springer-Verlag, 1998.

Mark Cutkosky. Biomimetic robots - multi-university research initiative. http://www-cdr.stanford.edu/biomimetics/, 2004.

Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2000.

Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. In *IEEE Transactions on Robotics and Automation, 7(6)*, pages 859–865, 1991.

Gregory Dudek, Paul Freedman, and Souad Hadjres. Using local information in a non-local way for mapping graph-like worlds. In *IJCAI*, pages 1639–1647, 1993.

David Ferguson, Aaron Christopher Morris, Dirk Haehnel, Christopher Baker, Zachary Omohundro, Carlos Reverte, Scott Thayer, William Red L. Whittaker, Chuck Whittaker, Wolfram Burgard, and Sebastian Thrun. An autonomous robotic system for mapping abandoned mines. In *Advances in Neural Information Processing Systems (NIPS 03)*, 2003.

D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, 2000.

Matthias O. Franz, Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bülthoff. Learning view graphs for robot navigation. *Auton. Robots*, 5(1):111–125, 1998. ISSN 0929-5593. doi: http://dx.doi.org/10.1023/A:1008821210922.

Udo Frese and Tom Duckett. A multigrid approach for accelerating relaxation-based slam. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003)*, pages 39–46, Acapulco, Mexico, 2003.

Daniel D. Fu, Kristian J. Hammond, and Michael J. Swain. Navigation for everyday life. Technical Report [TR-96-03], 12, 1996.

M. Golfarelli, D. Maio, and S. Rizzi. Elastic correction of dead-reckoning errors in map building, 1998.

D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient fastslam algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proc. of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

T. Hartley, E. A. Maguire, H. J. Spiers, and N. Burgess. The well-worn route and the path less traveled: Distinct neural bases of route following and wayfinding in humans. *Neuron*, 37:877–888, 2003.

J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)*, pages 68–73. IEEE Press, 1996.

Joe Hicklin, Cleve Moler, and Peter Webb. Jama: A java matrix package. http://math.nist.gov/javanumerics/jama/, 1999.

Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. *Principles of Neuroscience*. McGraw-Hill, Appleton and Lange, 2000.

Roberta L. Klatzky. Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections. *Lecture Notes in Computer Science*, 1404:1–17, 1998. ISSN 0302-9743.

Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierachy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

Eleanor A. Maguire, David G. Gadian, Ingrid S. Johnsrude, Catriona D. Good, John Ashburner, Richard S. J. Frackowiak, and Christopher D. Frith. Navigation-related structural change in the hippocampi of taxi drivers. In *Proc Natl Acad Sci USA*, volume 97 (8), pages 4398–4403, 2000.

Hanspeter A. Mallot and Matthias O. Franz. Biological approaches to spatial representation - a survey (biomimetic robot navigation). *Robotics and Autonomous Systems*, 1999.

T. P. McNamara and A. L. Shelton. Cognitive maps and the hippocampus. Trends in cognitive sciences. In T. Hartley, E. A. Maguire, H. J. Spiers, and N.r Burgess, editors, *The well-worn route and the path less traveled: Distinct neural bases of route following and wayfinding in humans. Neuron, 37*, pages 877–888. 2003.

L. Nadel, K.G.F. Thomas, H.E. Laurance, and R. Skelton. Human place learning in a computer generated arena. In Christian Freksa, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition*, volume 1404 of *Lecture Notes in Computer Science*, pages 399–428. Springer, 1998.

John O'Keefe and Neil Burgess. Space, memory and the hippocampus. http://behemoth.maze.ucl.ac.uk/, 2004.

John O'Keefe and Lynn Nadel. *The Hippocampus as a cognitive map*. Oxford University Press, Walton Street, Oxford OX2 6DP, 1978.

Andrew Peper and Hitesh Tolani. Landmark navigation using neural nets. http://www.augsburg.edu/compsci/REU/navigation, 2001.

Emilio Remolina and Benjamin Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104, 2004.

Thorsten Ritz, Salih Adem, and Klaus Schulten. A Model for Photoreceptor-Based Magnetoreception in Birds. *Biophys. J.*, 78(2):707–718, 2000.

B. Ronacher, K. Gallizzi, S. Wohlgemuth, and R. Wehner. Lateral optic flow does not influence distance estimation in the desert ant Cataglyphis fortis. *J Exp Biol*, 203(7):1113–1121, 2000.

Rainer Rothkegel, Karl Friedrich Wender, and Sabine Schumacher. Judging spatial relations from memory. In Christian Freksa, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition*, volume 1404 of *Lecture Notes in Computer Science*, pages 79–106. Springer, 1998.

Etienne Save, Arnaud Cressant, Catherine Thinus-Blanc, and Bruno Poucet. Spatial firing of hippocampal place cells in blind rats. *The Journal of Neuroscience*, 18(5):18181826, 1998.

N. A. Stillings. Cognitive psychology: The architecture of the mind. In N. A. Stillings, M. H. Feinstein, J. L. Garfield, E. L. Rissland, D. A. Rosenbaum, S. E. Weisler, and L. Baker-Ward, editors, *Cognitive Science: An Introduction*, pages 17–63. MIT Press, Cambridge, MA, 1987.

S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R.P. Bonasso, and R Murphy, editors, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, 1998.

S. Thrun, D. Fox, W. Burgard, and Dellaert. F. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 2001.

Sebastian Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

Sebastian Thrun. Learning occupancy grids with forward sensor models. *Autonomous Robots*, 15:111–127, 2003.

Edward C. Tolman. Cognitive maps in rats and men. *The Psychological Review*, 4(55):189–208, 1948.

R. Wehner, B. Michel, and P. Antonsen. Visual navigation in insects: coupling of egocentric and geocentric information. *J Exp Biol*, 199(1):129–140, 1996.

S. Werner, B. Krieg-Brückner, and T. Herrmann. Modelling navigational knowledge by route graphs. In C. Freksa, C. Habel, and K.F. Wender, editors, *Spatial Cognition II*, number 1849 in Lecture Notes in Artificial Intelligence, pages 295–317. Springer-Verlag; D-69121 Heidelberg, Germany, 2000.

Steffen Werner and Thomas Schmidt. Investigating spatial reference systems through distortions in visual memory. In Christian Freksa, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition*, volume 1404 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 1998.

Wikipedia. The Free Encyclopedia. www.wikipedia.org, 2004.