Influencing Robot Learning Through Design and Social Interactions — A Framework for Balancing Designer Effort with Active and Explicit Interactions

Yuval Marom

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
2003

Abstract

This thesis examines a balance between designer effort required in biasing a robot's learning of a task, and the effort required from an experienced agent in influencing the learning using social interactions, and the effect of this balance on learning performance. In order to characterise this balance, a two dimensional design space is identified, where the dimensions represent the effort from the designer, who abstracts the robot's raw sensorimotor data according to the salient parts of the task to increasing degrees, and the effort from the experienced agent, who interacts with the learner robot using increasing degrees of complexities to actively accentuate the salient parts of the task and explicitly communicate about them. While the influence from the designer must be imposed at design time, the influence from the experienced agent can be tailored during the social interactions because this agent is situated in the environment while the robot is learning. The design space is proposed as a general characterisation of robotic systems that learn from social interactions.

The usefulness of the design space is shown firstly by organising the related work into the space, secondly by providing empirical investigations of the effect of the various influences on the robot's experiences and how learning performance varies as a function of these influences, and finally by identifying how the conclusions from these investigations apply to the related work for improving learning performance. The empirical investigations implement different learning approaches, and are conducted with simulated and physical mobile robots learning wall-following and phototaxis tasks from an experienced simulated robot or an experienced human, and with a simulated humanoid robot learning an object-interaction task from an experienced simulated robot. The design space is used not only to characterise these investigations and related work, but also to characterise a typical performance surface that can be used to guide the design of new and existing systems. The characterisation shows that a particular level of performance can be maintained by compensating one source of influence for the other, and that performance can generally be improved by increasing the influence from any of these sources. It also shows that the best performance depends on various factors that affect the robot's overall learning potential, such as the available learning resources.

The thesis argues that characterising the balance between designer effort and social interactions and how learning performance is affected is crucial for addressing a difficult trade-off: increasing designer effort for biasing the learning of a particular task in a particular environment and thus providing more reliability, versus increasing the influence from the social interactions thus providing more generality.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yuval Marom*)

Acknowledgements

I would firstly like to thank my supervisor, Gillian Hayes, for her valuable criticisms, suggestions, and ideas, and for always finding the gaps in my work. Thank you also for helping me put the pieces of the puzzle together into a coherent and sensible story. Thanks also to my second supervisor, John Hallam, for providing valuable feedback and suggestions, and to the head of the Institute of Perception, Action and Behaviour (IPAB), Bob Fisher, for providing valuable feedback and guidance. Other people who have read and commented on various chapters are Paul Crook, Tim Lukins, Tim Taylor, and Jay Bradley from IPAB, and Aude Billard from EPFL — thank you all!

I carried out a substantial amount of work in collaboration with a fellow PhD student at IPAB, George Maistros, so a tremendous amount of thanks to him for putting up with my peculiarities, and for not exposing too many of his own! It's been a pleasure working together with him (honestly!) and he's also been a great mate. Thanks also for lots and lots of stimulating discussion, particularly on the various ideas of this thesis, and for reading and commenting on most of it. Special thanks must also go to Stephen Marsland from The University of Manchester, for making parts of his PhD work available for me to use, and especially for reading and commenting very diligently on drafts of various chapters of this thesis. There are many people who have helped to provide a friendly, interesting, and motivating working environment: all the various past and present members of IPAB and particularly the Mobile Robotics Research Group (mrobots). Specifically thanks to Yiannis Demiris and Jose Carmena for providing a great source of motivation and encouragement.

There are people outside the working environment who have provided valuable friendships and encouragement. I have been lucky to find great friends, themselves PhD students in various areas, and so sharing PhD worries with them has been essential to maintaining an efficient level of sanity. People have come and gone in the last few years, many thanks to you all, especially to Nick, who also read parts of the thesis, and kept reminding me that the end is near! My closest friend and partner, Gwyn, was there for me no matter what my stress level was. At times, her support, love and companionship seemed crucial — thank you!

And lastly, this would not be possible without the love and support of my family, who never stopped believing in me. Thanks to my two sisters, Michal and Dorit, for keeping a constant interest in my life. And massive thanks to my parents, Irit and Murray, for reading through various drafts of this thesis and providing many comments and suggestions, and more generally for being my mentors and number-one fans!

To Irit and Murray Aitkin.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Work in Artificial Intelligence, and particularly work involving robots, has been focusing increasingly on the physical situatedness, embodiment, and development of autonomous systems. It is now believed that in order to have fully autonomous robots, one cannot simply provide them with plans and reasoning mechanisms. Robots exist in an environment and interact with it. Therefore, their internal knowledge and representations of the environment and the possibilities that it presents for action must be influenced by these interactions (Brooks, 1991; Matarić, 1997; Brooks et al., 1998; Pfeifer et al., 2001a,b; Balkenius et al., 2001; Prince et al., 2002; Ziemke, 2002). One way that this can be achieved is through learning — both supervised and unsupervised machine learning approaches have been used widely in robotic platforms, where robots learn the perception-action regularities in an environment from experience. These approaches can generally be referred to as *robot learning*.

There is a great advantage in having robots that can learn to perform tasks from their own experiences, over robots that need to be programmed to perform tasks. Programming a robot can be difficult, inaccurate, and time-consuming. For non-programmers it would in fact be impossible to program a robot, but even for advanced programmers it might be difficult to understand the way the robot-environment interactions work; a lot of guess-work and trial-and-error can be involved in discovering what the robot perceives and the consequences of its actions.

However, programming robots to learn is also difficult, because robots are continuously faced with sensorimotor information, which is usually noisy and not well-structured. In order to learn a particular task in a particular environment, a learning system needs to determine what are the meaningful components of the sensorimotor data corresponding to this task and environment. But the notion of a task is usually external to the robot, imposed on it in order to fulfill particular purposes or needs. So in order for the robot to be able to generalise from all its perceptions appropriately for this particular task and address the particular needs, some external influence or bias must be provided. A lot of this bias is already present in the choice of learning architecture. However, the learning must also be influenced according to the *relevance* and *saliency* of information. The former is needed in order to reduce all the robot's experiences into a subset of those related to the task; the latter specifies the granularity at which significant differences occur in the sensorimotor data for the particular task. Both types of information are subjective to the task and environment. For example, an obstacle-avoidance task and an object-pushing task involve rather conflicting experiences, one involves moving away from objects, and the other approaching objects; further, if the task involves manipulating objects, the robot needs to detect finer differences between objects than if the task involves pushing these objects.

One approach to dealing with relevance is to let the robot explore its environment, and provide it with an internal mechanism that favours or rewards certain sensorimotor experiences over others. However, programming such a mechanism can be as difficult, inaccurate, and time-consuming as programming the robot to perform the task in the first place. Further, it might take the robot a long time to go through all the possible experiences and discover those relevant to the task. But there is no need for the robot to explore its environment *alone*. There is an increasing emphasis in the robotics field on designing robots that can exist in the same environment as other robots or humans and interact with them. Moreover, if these other robots or humans already know how to act in the environment, there is potentially much useful knowledge in such 'social environments', which can be obtained through 'social interactions', rather than through programming.

Therefore, another approach for dealing with relevance is to provide the robot with mechanisms for interacting with other agents in its environment, where the purpose of these interactions is to expose the robot to relevant experiences. The mechanisms required for the interactions can be designed independently of the task and environment, as long as they are useful and general enough for learning different classes of tasks. The work reported in this thesis takes this second approach, where a learner robot learns a task through interactions with an expert who is performing the task, and the learner implicitly relies on the expert to expose it to relevant experiences. Although social interactions restrict the sensorimotor experiences of the robot, there is still a major difficulty that must be addressed: finding saliency in the experiences. This is particularly difficult when the robot's exposure to sensorimotor data is imprecise due to the fact that it copies the expert's actions imprecisely.

1.1 Dealing with Saliency

In this thesis the word 'saliency' is used in a rather technical sense[1] — as pointed out earlier, it refers to the level of granularity or abstraction at which significance is assessed. As mentioned above, learning directly from the robot's raw sensorimotor data is very difficult because they are noisy and unstructured; similarly, dealing with saliency at this low level of representation is difficult. Researchers often simplify the problem by choosing a suitably high level of abstraction at which learning occurs, where the sensorimotor space is restricted to fewer, more precise, discrete structures. This can be achieved either in an ad-hoc manner, by providing the low to high-level mappings at design time, or by enabling the robot to acquire the mappings autonomously. The former case requires careful programming, and is not guaranteed to be 'correct', that is, useful for the robot; and it can only be useful for the specific task and environment for which it was designed. The latter case requires a reliable and general mapping mechanism, which inevitably has to deal with the issue of saliency; with such a mechanism, however, the acquired mappings are more faithful and adaptive to the robot's experiences.

Whether learning occurs at a low level of abstraction or a higher one, the issue of saliency must be dealt with. This generally corresponds to setting saliency detection parameters. Again, the faithfulness to true saliency is dependent on how careful the designer is in setting these parameters, or on whether the robot is able to autonomously tune and adapt them. In both cases it is very important that the designer identify the important parameters.

It was pointed out above that social interactions provide task-relevance to a learner robot, and this is indeed widely recognised in the literature, as well as the fact that social interactions are useful for speeding-up learning in robots (Demiris and Hayes, 1996; Schaal, 1999; Matarić, 2000; Gaussier et al., 1998). However, what is not recognised is that social interactions can be useful for actively influencing the learner's detection of saliency. There are many different forms of social interactions, and they can be used in different ways to influence the learning of a robot: from a simple demonstration of a task by an expert, to more active interactions with which the expert can manipulate the learner's exposure to experiences, accentuate the salient differences in these experiences, and even explicitly communicate about them.

To summarise, the learning by a robot must be biased by an external source, who knows about the task to be learned. This source can either be the designer, who must guess the learning dynamics at design time, or it can be an experienced agent, who is situated in the same environment and performs the same task, while the robot is learning the task in the environ-

[1]rather than its more common everyday use relating to conspicuity.

ment. The problem that this thesis addresses is finding a balance between these two sources of influence, in a way that achieves a learning setup which requires less designer effort, and is more general, adaptive, and faithful to the robot's experiences.

1.2 Balancing Saliency Bias

Because related work on socially-interactive learning robots has not yet explored the possibility of using social interactions to purposely and actively influence saliency detection, it has also not considered balancing the designer effort in biasing saliency with the exploitation of social interactions for biasing saliency. Instead, researchers generally choose a particular type of social interaction of interest, and then choose a suitably high level of abstraction that works, by either pre-classifying the sensorimotor data, and/or fixing saliency detection values; a suitable learning architecture is then able to learn the salient sensorimotor regularities in the perceptions that the robot is exposed to through the interactions. The classifications, and especially the setting of saliency parameter values, are generally performed ad-hoc through trial-and-error, rather than through parameterised mechanisms that recognise the important parameters.

The work presented in this thesis considers social interactions of *differing* complexities, and parameterised mechanisms for detecting saliency and representing data at *different* levels of abstraction, and shows how they interact to affect learning performance. That is, the work shows how abstraction by the designer can be balanced by higher complexity in the social interactions, and the effect this has on learning performance. The explicit modelling of saliency is realised through the concept of *attention*, which is recognised as responsible for detecting saliency and thus abstracting from the raw data; the influence of the parameters is tested, and the most important ones are identified. It is important to note that in this thesis, attention is modelled in the *temporal* dimension, rather than the spatial. That is, attention selects salient data points based on previous data, rather than selecting salient spatial features in the data.

The work reported in this thesis mainly involves a mobile robot learning wall-following and phototaxis tasks from an expert; a simulation environment contains both a simulated learner and a simulated expert, while a physical environment contains a physical robot and a human expert. Another component of the work involves a simulated humanoid robot learning from an identical simulated expert to interact with an object. The different levels of social interactions and designer effort that are identified in this thesis provide a useful and general way of categorising work involving different kinds of robots that learn from different kinds of social interactions, and should therefore be useful to researchers wishing to design and implement such systems.

1.2.1 Levels of Social Interactions

The varying complexities of social interactions are related to the role that an expert takes in demonstrating a task to a learner robot, as shown in Figure 1.1. The expert can demonstrate the task passively, *i.e.* independently of what the learner is doing, or actively, *i.e.* by tailoring the demonstrations based on what the robot is doing, in order to expose it to experiences which will increase the robot's ability to learn from them. A further increase in the complexity of the interactions involves the expert sending the robot explicit signals which are used directly by a learning architecture to influence the learning.

Another increase is suggested in this thesis, but not implemented, whereby the expert and learner communicate about their internal states, for the purpose of tuning saliency parameters, and thus completely eliminating the need to tune parameters at design time. Of course, this can only be possible if the important saliency parameters are identified and modelled explicitly in the first place.

1.2.2 Levels of Designer Effort

Designer effort in this thesis corresponds to the activities performed by the designer in biasing the detection of saliency and hence determining the level of abstraction at which learning occurs. Different amounts of designer effort thus correspond to different levels of abstraction, as shown in Figure 1.1. The thesis will refer to 'amounts' and 'levels' of designer effort interchangeably. At the lowest level, learning occurs on the raw, unstructured, sensorimotor data. At this level, learning and saliency detection are difficult because the robot is faced with a continual input stream, which is noisy and unstructured. The thesis shows that designer effort here is useful in designing attention saliency mechanisms for modulating the amount of data for learning, in the face of limited learning resources.

An increase in designer effort is shown in this thesis by demonstrating that attention can also be used to organise and classify the perceptual experiences of the robot into discrete structures which are then used directly for learning, and therefore learning occurs at a higher level of abstraction. At this level of abstraction there is an information loss from what is, in essence, a compression of the raw data. Therefore here, relative to the low level of abstraction, more care must be taken by the designer in setting the saliency parameters, in order to achieve a representation that is useful for the robot to learn the task with.

In the majority of related work, learning occurs at an even higher level of abstraction, where the classifications of the robot's experiences are derived ad-hoc, using criteria set by

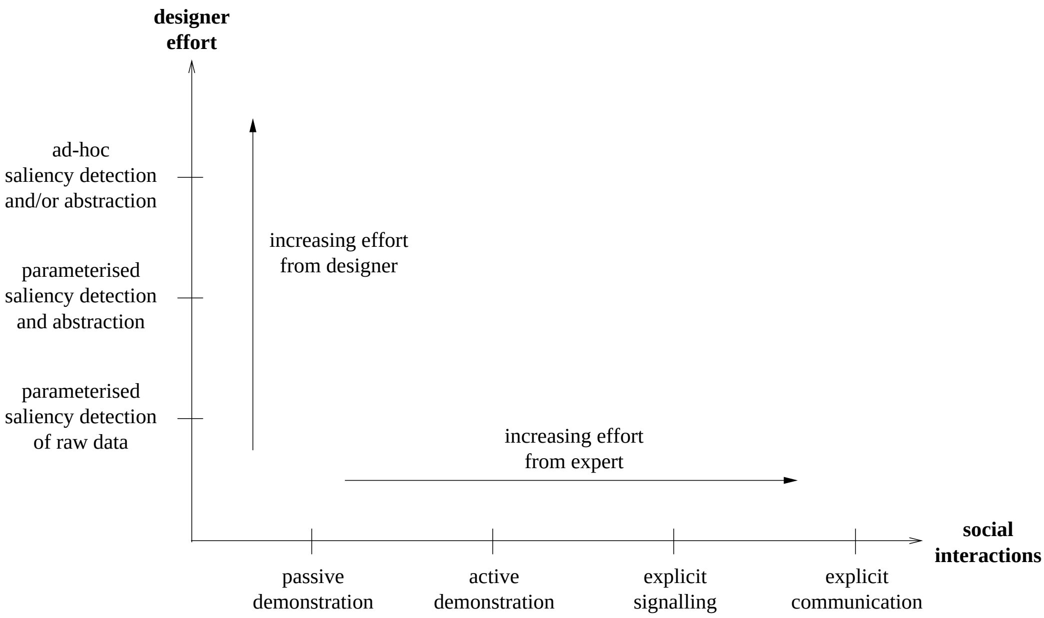


Figure 1.1: Levels of social interactions and designer effort. The designer effort axis corresponds to different levels of effort from the designer, required for learning at different levels of abstraction; the social interactions axis specifies interactions of different complexities, corresponding to different levels of effort by the expert demonstrator. The thesis argues that the expert's effort can be used to balance the effort required in programming the robot at design.

the designer, and not through the robot's real experiences; in these cases saliency detection can be quite trivial, and is therefore not always treated explicitly. In other cases where the classifications *are* derived automatically from the robot's experiences, the saliency parameters are generally set ad-hoc. In both of these cases the designer has to take great care when setting classification boundaries, and/or saliency parameter values, because this is biasing what kind of representation the robot can achieve, and therefore what it can learn, and this bias is fixed before the robot has even started to interact with its environment. Of course, if the designer manages to design a useful and reliable representation then this is clearly advantageous, however, it lacks generality.

1.2.3 The Interplay between Social Interactions and Designer Effort

The identification of different levels of social interactions and designer effort is crucial for investigating the interactions between them. Such an investigation is missing in current research, but is necessary in order to address the problem posed at the start of this chapter: balancing between the effort required at design time and at interaction time, in providing external bias to the learning of a task by a robot who has no internal notion of this task. In this thesis, social interactions and designer effort are identified as two ‘dimensions’ in the design space of robotic learning systems, as shown in Figure 1.1, and the interplay between them is characterised. The usefulness of this characterisation for suggesting how to improve existing systems or design new ones is shown, firstly by demonstrating how the characterisation applies to related work (Chapter 2), secondly by presenting an empirical investigation of how performance varies as a function of the two dimensions (Chapters 3–5), and lastly by considering the implications of this investigation to the related work (Chapter 6).

The investigations from this thesis show that when a lot of effort is given by the designer in setting saliency parameters and abstracting the sensorimotor data, the interactions between the expert and learner can be relatively simple. However, a similar level of performance can be achieved with less effort by the designer, and more active interactions and explicit signalling of saliency from the expert. This conclusion is demonstrated in the experiments, which compare performance of learning at different levels of abstraction, using different amounts of effort by the designer, and with the different types of social interactions. The experiments also show how this performance is affected by limited learning resources.

The work in this thesis can be used more generally to guide the design of robotic systems that learn from social interactions, as follows. If a designer has a particular type of social interaction in mind, and existing saliency detection mechanisms that operate at a particular level of abstraction, the characterisation can be used to suggest what is the best performance that can be *expected* given these resources, and more importantly, how performance can be *improved*. Similarly, the characterisation can be used to suggest what combinations of social interactions and designer effort are necessary to achieve a desired level of performance.

It is important to note that designer effort in this thesis refers *only* to biasing the robot’s *existing* learning capabilities. Of course, other types of designer effort are involved in providing the robot with such capabilities in the first place; for example, choosing the robot’s morphology and thus biasing the nature of the robot’s sensorimotor data; or choosing a particular learning architecture, which is set up to learn in a particular way. The designer could spend more effort, for example in choosing specific sensors, or choosing a specific learning architecture, which will work well for a particular task, however such effort will not be reflected by an increase on the vertical axis in the space identified in this thesis.

Therefore the characterisation does not give an *absolute* measure of performance, because it is specified independently of other design issues. Indeed, learning performance is characterised in this thesis using experiments where different learning architectures are used. The characterisation is useful for indicating how performance can be *improved* through a combination of social interactions and designer effort, with the *given* learning capabilities. In exactly this way the characterisation can be used by others to suggest how to improve the performance of their systems by following the two dimensions identified here.

1.3 Contributions

In comparison to related work in the field, where a particular type of social interaction is chosen and the representation is then confined to a particular level of abstraction, the work presented in this thesis provides the following contributions:

1. It evaluates the benefit of social interactions for learning, by considering interactions of different complexities, from passive to more interactive ones, and shows the increasing influence they have on the learning by a robot.

2. It does not restrict the representation to a convenient high level of abstraction, but rather considers explicit parameterised mechanisms of attention that operate at different levels, starting from a low level where little design effort is required. Parameterised mechanisms enable the designer to design saliency more reliably, and also allow for saliency to be biased by social interactions. The traditional role of attention as a mechanism for dealing with limited resources is also demonstrated.

3. It provides an investigation into the interplay between social interactions and designer effort, which shows that the amount of effort required by the designer in biasing the robot's learning of a task can be balanced by the amount of effort from an experienced teacher in influencing the learning during the social interactions, while maintaining the learning performance. This means that not only can the learning setup be more general and require less design effort, but also that learning can be more adaptive and faithful to the robot's experiences.

4. By considering social interactions and designer effort as 'dimensions' in the design space of robotic learning systems, a sub-space is identified and characterised, which is useful for two principal reasons:

- **Analysis**. It helps to analyse and characterise the findings mentioned in point 3 above, and compare them to findings from related work.
- **Design**. It can be used by others to guide the design of similar systems, by suggesting a level of performance based on available resources (for example, the particular social interactions available for use, or the available mechanisms of saliency detection), and learning limitations (for example, a bound on the number of learning examples that can be considered).

1.4 The Organisation of the Thesis

The remainder of the thesis is organised as follows.

Chapter 2 defines in detail the 'social interactions' and 'designer effort' dimensions used to make up the space shown in Figure 1.1, organises the related work into the space, identifies the gap that exists in current research with respect to this space, and specifies how this gap is filled by the work reported in this thesis. The chapter also briefly discusses how the investigation in the experimental chapters that follow will lead to a discussion on performance within the space. Further, the biological and psychological inspirations that have given rise to the various ideas in this thesis are presented.

Chapter 3 illustrates what it means to consider data at different levels of abstraction, and what is the effect of increasing the complexity of the social interactions. This chapter does not present any actual *learning* of a task. Rather, it presents methodologies for inspecting the data that the robot is exposed to through the interactions. Such inspections are useful for indicating how well the robot might learn under various conditions, and they can therefore guide design decisions such as the choice of learning architecture. The first part of the chapter provides a visual and statistical analysis of low-level, 'raw', sensorimotor data. The various tasks presented here are: wall-following and phototaxis in a simulated Khepera mobile robot platform, consisting of a learner robot following behind a teacher robot, who is executing the task; wall-following with a RWI-B21 physical mobile robot, following behind a human teacher; and a simulated 11 degrees of freedom humanoid in the DynaMechs simulator, learning to interact with an object from observation of an identical teacher humanoid. The learning of these tasks is presented in Chapters 4 and 5.

The second part of the chapter presents a growing, unsupervised, clustering method for mapping raw data into higher-level classifications that represent the perceptions of the robot. The implementation of this method is essentially the attention system: it detects saliency and performs the low to high-level mappings. The crucial parameters of this attention system are identified and made explicit, and two roles of attention are identified, as described below, for learning at different levels of abstraction — each depending on different parameters that reflect different forms of saliency.

Chapter 4 presents learning at a low level of abstraction, where the learning occurs on the raw sensorimotor data, and the role of the attention system is to modulate the amount of learning. The chapter shows that this role for attention is useful when learning from raw data, if one cannot afford to consider all the available data points to learn from, for example if the number of learning examples that the learning architecture can deal with is limited. It also shows that at this level of abstraction, the learner benefits from as much complexity in the social interactions as possible.

Chapter 5 provides examples of learning at a higher level of abstraction, where the learning occurs on the classifications of the raw data provided by the attention system; here attention is still used to modulate the amount of learning, but its more significant role now is to organise the perceptions of the robot usefully for learning. Learning is achieved by associating the perceptual classifications with appropriate representations of the motoric information. Three sets of experiments in this chapter present different ways of implementing such learning, corresponding to different levels of effort required by the designer (relating to the abstraction of motor data); they show how this effort can be balanced by the social interactions.

The first set of experiments requires the least effort by the designer, and consequently the robot needs the social interactions to have a sufficiently high complexity. The second set of experiments explicitly compares learning with and without additional effort by the designer, and shows that learning is improved either through this additional effort, or by increasing the complexity of the social interactions. The third set of experiments requires the most effort by the designer, and here social interactions are less important, and in fact the robot benefits from some self-learning.

Chapter 6 discusses the conclusions from all the experiments, and casts them back to the space suggested in Figure 1.1. The usefulness of this space as a design tool is discussed in light of the results, by identifying a typical performance surface, comparing the results to results from related work and showing that this surface applies to related work, and suggesting how the current and related work could be improved in the space.

Chapter 2

Characterisation of a Design Space

This chapter defines the two dimensions identified in Chapter 1 for characterising a design space. These two dimensions are 'social interactions' and 'designer effort', and the purpose of the design space is to provide a framework with which to balance two sources that can be used to influence the learning by a robot. The social interactions dimension characterises different ways in which an expert can influence a robot's learning while the learner and expert are interacting, whereas the designer effort dimension characterises different ways in which the designer can influence the learning at design time by abstracting the robot's experiences. As this chapter will show, different factors are taken into consideration when balancing these two sources of influence: the reliability, robustness, faithfulness, adaptiveness, and generality of the influence on learning, as well as the effort involved (from the designer and from the expert).

Sections 2.1 and 2.2 respectively deal with each of these two dimensions, and Section 2.3 then organises related work into the proposed design space. Section 2.4 highlights the gap in the space left by current research, and mentions how the work in this thesis fills this gap. Section 2.5 briefly explains how the investigation in the experimental chapters that follow will lead to a discussion on performance within the space, and how performance is affected by the available learning resources. Finally, Section 2.6 presents the biological and psychological inspirations that have given rise to the various ideas in this thesis.

2.1 The Social Interactions Axis

Designing socially-interactive robots involves developing systems that can benefit from the existence of experienced and knowledgeable agents, artificial or human, in their environments, to develop their own skills. As discussed in Chapter 1, this is appealing because one can then avoid the need to explicitly program these systems, which could be difficult, inaccurate, and time-consuming. The human-artifact relationship is particularly appealing because humans want robots as aides and companions in their daily life, and we want them to interact with us in natural ways.

2.1.1 Social Interactions for Different Purposes

A growing interest in social interactions has existed in robotics since the early 90's. Dautenhahn (1995) summarised existing work and motivated such research by highlighting features from primate cognition that might be useful for robots. She has recently contributed to a survey into socially-interactive robots (Fong et al., 2003). In that survey, different purposes for social interactions are identified that have been addressed by different types of research up-to-date, and they are summarised as follows:

- designing useful interaction mechanisms for encouraging human-robot interactions;
- designing believable human personalities for robots, for example for autism therapy, or for replacing humans with artificial agents to provide automated advice or information;
- studying embodied models of social behaviour, such as relationships, emotions, and intentionality;
- studying the development of social and other skills.

Common to all these purposes is the importance for the system to be able to exploit the presence of other agents in its environment, by interacting with them usefully; and on the other side of this relationship, it is important for the agent being 'exploited' to be able to interact with the system in a natural manner, so it can provide the system with what it needs, and feel comfortable interacting with it. As Fong et al. (2003) argue, robots still lack the social skills necessary for them to be accepted as natural interaction partners. Much of the research is focused primarily on the social interactions themselves, and it faces very difficult issues, from technological ones such as real-time vision, spatial attention, and motion control, to theoretical cognitive ones, such as emotions and personalities.

A different line of research, related to the fourth purpose listed above, addresses the *benefits* of social interactions for learning new skills. That is, it investigates learning processes that take advantage of potential sources of information present in the environment explicitly or implicitly emanating from other agents. This line of research is not mutually exclusive from the rest. Since in animals the process of learning is influenced by forming relationships, finding a ‘good’ teacher, and other social aspects, this might also be beneficial for robots. Dautenhahn pointed this out in her 1995 paper and again in a recent book on imitation in animals and artifacts (Dautenhahn and Nehaniv, 2002).

The work reported in this thesis focuses on the benefits of social interactions for learning, but it does *not* consider social issues relating to forming relationships etc. Therefore, in this kind of work there is a single learner robot, and a single experienced teacher, and the learner is guaranteed that interacting with this teacher can only be beneficial for its learning needs. The remainder of this review will focus on this kind of work.

2.1.2 Types of Social Interactions for Learning

The survey by Fong et al. (2003) demonstrates a wide range of robotic and software systems, which utilise social interactions of various types, and where different design issues are addressed. For example, robots that are capable of expressing emotions must be able to communicate through speech and facial expressions; in robots where embodiment is important the morphology of the robot must be carefully designed; and in robots where human-centered perception is important the robot must be able to track people, and perform speech, gesture, and face recognition. These distinctions are made with reference to the capabilities of the robot (or software agent).

In contrast, the different types of social interactions identified in this thesis refer to different capabilities *of the expert*. More precisely, they refer to different ways in which the expert can interact with the robot to *purposely* influence the robot’s learning, and these different interactions are claimed to be of increasing complexity — they require different levels of ‘effort’ by the expert. Four types of interactions are identified, and will be discussed below: passive demonstrations, active demonstrations, explicit signalling, and explicit communication. In the first two the teaching is implicit — there is no explicit transfer of information between the teacher and the learner; the teacher merely demonstrates a task, and the learner learns in terms of its own experiences. In the latter two types of social interactions the demonstrations are enhanced with an explicit one-directional influence from the teacher on the learning by the learner, or with two-directional exchange of information between the teacher and learner, respectively.

With regards to design issues, the relevant ones identified in the survey by Fong et al.

(2003) are those concerning human-centered perception. The different types of social interactions mentioned above rely on the fact that the robot has the appropriate mechanisms for these interactions. That is, it must be able to track the teacher who is demonstrating a task, and it must be able to receive and send explicit information, as required. Further, these capabilities should be task-independent, that is, they should be useful for interacting with the expert regardless of what task is involved.

Passive Demonstrations

The minimal amount of effort required by an expert wishing to teach some task to a learner robot is to execute the task as if there is no learner. In other words, the expert 'demonstrates' the task *passively* and independently of the states or actions of the learner. In the early work on learning by imitation (Hayes and Demiris, 1994), this kind of minimalistic effort from the expert is actually promoted as one of the advantages of programming robots through demonstration, especially when the demonstrator is another robot, and one wishes to capitalise on its existing knowledge in training another robot with the least effort. This is indeed an advantage if learning from such passive demonstrations is possible. However, a passive demonstration is not always a sensible demonstration strategy because, for example, the ability of the learner robot to learn could be hampered if it loses sight of the teacher or if it struggles to copy the actions of the teacher. Indeed, when the expert is a human, he/she inevitably takes more care in demonstrating the task, whereas examples of passive demonstrations are generally attributed to robotic demonstrators, as will be shown in Section 2.3.

This problem of the learner having the ability to match the actions of the teacher is very challenging, especially when we consider that the learner and teacher can have different morphologies (see, for example, the 'correspondence problem', formulated by Nehaniv and Dautenhahn, 2000). However in the majority of related work, the researchers design their learners and teachers in such a way that they can bypass this problem, and therefore ensure that the learner is able to copy the actions appropriately for the particular task. In contrast, Alissandrakis et al. (2000) address the correspondence problem in a simulated chess-board environment, where the different pieces are allowed to move in different ways; this work is discussed in more detail in Section 2.3.

Active Demonstrations

There are at least three ways in which the expert might tailor the demonstration according to the states or actions of the learner, and therefore demonstrate the task more actively. Firstly, the teacher could adapt the demonstrations in order to make it easier for the learner to match the teacher's actions, for example, by slowing down the demonstration. In the mobile robot experiments by Billard and Hayes (1999), a learner robot follows a teacher around an environment, but the teacher can also detect the learner and align itself in front of it, thus reducing the possibility that the learner loses the teacher.

Secondly, the teacher might perform the demonstration in such a way as to ensure that not only does the learner not get lost, but that it is actually exposed to 'clean', consistent, and distinct experiences. By observing and inferring the action-copying behaviour of the learner, the teacher can manipulate the learner's experiences. For example, Gaussier et al. (1998) report that in physical experiments involving a human teaching a mobile robot various 'dances', the demonstrations are inevitably more adaptive than in similar simulated experiments involving a simulated teacher; the human teacher ensures the learner passes exactly through correct edges in the trajectories of the dance, and that the timings of the learner's actions are precise, by adapting his (the teacher's) own trajectory and speed.

The third way in which the expert can influence the demonstrations is by deviating from the 'natural' demonstration in order to 'exaggerate', or accentuate, the important differences between the components of the task. These kinds of interactions are not reported in related work, although a human demonstrator could be performing such demonstrations without realising it.

Kaplan et al. (2001) suggest other active demonstration methods, which are inspired from techniques used by humans to train animals, especially dogs. For example, they suggest more physical interactions (termed 'modelling', or 'moulding'), where the trainer physically manipulates the animal into the desired positions. In robotics, this could correspond, for example, to manipulating the robot with a joystick (Kaiser and Dillmann, 1996; Hugues and Drogoul, 2001).

These various kinds of active demonstrations are difficult to program for a robotic teacher, whereas for a human teacher such demonstrations are not only easier, but also more adaptive. The reason for this is that a human teacher is situated in the environment *together* with the learner, *while* the learner is learning, and can therefore tailor the interactions 'on-line' in response to what the robot is doing. In contrast, in order to program an equivalent robotic teacher, the designer would have to guess what would be a good active strategy *before* the interactions begin. Also, it can be argued that human demonstrations are inevitably adaptive to the robot's interests. More specific examples of active demonstrations are given in Section 2.3, with a particular emphasis on why such demonstrations are useful for influencing learning.

Explicit Signalling

In the examples of social interactions presented so far, there has been no explicit transfer of information between the expert and the learner. There is, however, an *implicit* transfer of information, because the learner learns to perform a novel task through the social interactions. If the expert has the ability to send explicit signals to the learner as well as demonstrate the task, and the learner has the ability to detect and interpret these signals, this more complex type of interaction can have various uses.

The signals from the expert can form part of the stimulus that the learner learns from, with the aim of learning a symbolic representation of the sensorimotor data, such as a language for communication (Billard and Hayes, 1999; Kaplan et al., 2001). In other approaches where symbolic learning is not required, and thus the signals do not form part of the learning, the signals can still be used to directly *affect* the learning. One purpose of such signalling is to explicitly draw the learner's attention to salient experiences (Moukas and Hayes, 1996; Nicolescu and Matarić, 2003), and another purpose is to provide the learner with feedback about its actions. This latter kind of signalling is generally used when it is the sole source of social interactions, that is, when demonstrations are not available; instead, the learner already has basic sensorimotor skills, and the expert teaches a task utilising these skills by rewarding the relevant ones (Nehmzow and McGonigle, 1994; Kaplan et al., 2001).

Explicit Communication

With explicit signalling, the expert sends the learner signals that influence the learning directly and explicitly. When sending these kinds of signals, it might be useful for the expert to know how the learning is being influenced, for example how the learning is progressing. If the learner could send signals back to the expert about its internal states, for example how familiar experiences are, then the expert could use such information to determine how to proceed with the demonstration and signalling. For example, if a particular experience is not familiar enough then the expert might demonstrate and signal it more frequently. Explicit signals from the expert might also be used by the robot to tune parameters that determine how it perceives and learns from experiences, in which case it might be beneficial for the expert to have an idea of the effect that the signals have on such parameters.

This type of social interactions is only considered speculatively in this thesis — it is not actually implemented. However, the thesis will discuss the implications of the increase in complexity of the social interactions that *are* implemented for this type of social interactions. The thesis will also discuss how particular design choices, relating to the level of abstraction used for learning, are guided by the overall aim of transferring as much of the external influence on the robot's learning from the designer, to the expert involved in the social interactions; it will be argued that the most influence an expert can have on the robot's learning is achieved through explicit communication.

Kaplan et al. (2001) utilise a combination of explicit (verbal) and implicit (non-verbal) communication between an expert and a robot to refine the robot's learned sequence of behaviours. The robot demonstrates the sequence of behaviours that it has just learned (non-verbal communication), which might include some irrelevant behaviours, and the expert only rewards (verbal communication) the relevant behaviours; the robot then updates its internal measure of similarity between behaviours which influences their sequencing (see more on this in Section 2.3). Klingspor et al. (1997) identify different types of verbal and non-verbal communication strategies, with which a robot gives feedback to its user about its perceptions and actions when it executes a learned task. However, they do not discuss how this feedback is used by the user to influence further demonstrations.

These two approaches use communication to refine a behaviour which is already learned, whereas communication is proposed here as an on-line approach for influencing the learning *while the robot is learning*.

2.2 The Designer Effort Axis

The previous section described different ways in which an expert can influence the learning of a robot, using social interactions while the robot is learning. In this section we will discuss another source of influence on the robot's learning, arising from the designer's choices of level of abstraction and measure of saliency at design time[1]. The thesis argues that in order to be able to balance these two sources of influence, one must treat saliency explicitly, and this is addressed in this thesis using the concept of *attention*. This is not to say that one must always consider 'attention' — it is simply a conceptual choice in this thesis. Nevertheless, this concept will be used here to refer to the different levels of designer effort relating to biasing saliency

[1] See start of Section 1.1 on page 3 for the definition of 'saliency'.

and abstraction in the literature, although it is not always the concept that others use. If one considers attention as the capability with which an agent controls its perceptual strategy, in order to be selective in its perception, then it is useful to keep in mind the following two objectives identified below by Hayes-Roth (1995), when evaluating the benefits of attention:

> "In general, a resource-bounded agent... must be highly selective in its perception of the environment and it must adapt its perceptual strategy to balance two objectives. First, from a purely quantitative perspective, the agent must maximise its vigilance, perceiving as much information as possible, while avoiding perceptual overload. Second, from a qualitative perspective, the agent must maximise goal-directed effectiveness, readily acquiring data that are relevant to its currently important reasoning tasks, while avoiding distractions by irrelevant or insignificant data." (Hayes-Roth, 1995, p. 339)

These objectives are addressed in this thesis as follows. The robot's 'currently important reasoning task' is learning; attention *selects* perceptions for learning; the quantitative benefit of attention is that it selects some perceptions over others, therefore reducing perceptual load and using fewer learning resources; the qualitative benefit of attention is that it selects perceptions based on their significance, or *saliency*. Hayes-Roth argues that the agent should also avoid irrelevant data, and the importance of relevance was also mentioned in Chapter 1. In this thesis the issue of relevance is handled implicitly by the social interactions, in the sense that the learner implicitly relies on the expert to only expose it to experiences which are relevant to what it needs to learn.

There is a very important distinction that must be made with regards to what selection refers to. The selection of perceptions can either be based on spatial saliency, in order to allocate perceptual resources in space, or it can be based on temporal saliency, in order to allocate resources in time. Spatial attention involves selecting salient spatial locations where a location's saliency is calculated with respect to its neighbouring spatial locations. An alternative view of attention involves a temporal selection of salient perceptual *instances*, where saliency is calculated with respect to previous experiences, or instances, and/or a prediction of future ones. So rather than asking *where* the interesting locations are given a snap-shot of the environment, temporal selection is concerned with *when* a snap-shot should be taken in the first place.

This distinction is shown schematically in Figure 2.1. Each instance in Figure 2.1 might correspond to the pixels in a visual image, or the collection of sonar sensor readings of a robot. A system with temporal selection alone will need to deal with the full instance of sensor readings, since there is no spatial selection. Therefore the purpose of a spatial attention is

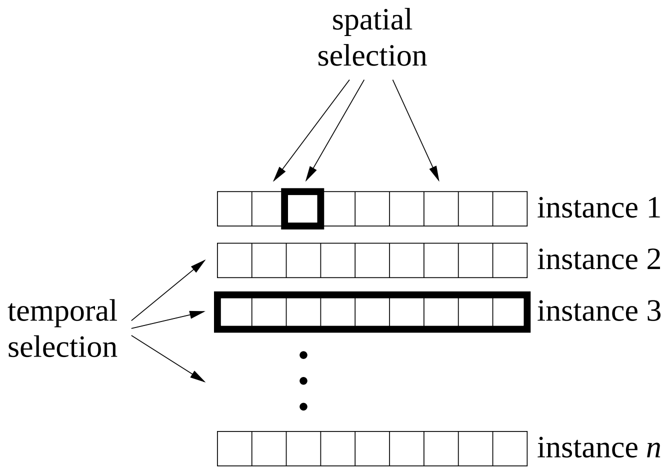


Figure 2.1: Spatial versus temporal attention. In spatial selection a location on a given perceptual instance is inspected (highlighted 'pixel' on instance 1), where the location's saliency is calculated with respect to its neighbouring spatial locations. Temporal selection involves the decision of which instance to inspect, where it is the saliency of the whole instance (instance 3 in this example) which is calculated, with respect to previous and/or future instances.

to reduce the dimensionality of the data being considered, whereas the purpose of temporal attention is to reduce the amount of data that are considered.

Both types of attention are very important. In computational sciences (especially Computer Vision), attention is generally used in the spatial sense. Some examples are saliency maps with winner-take-all (WTA) units (Koch and Ullman, 1985; Tsotsos et al., 1995), where input units from an image are connected to various intermediate layers before activating some higher-level output layer, and the units are then manipulated both bottom-up (Itti et al., 1998; Itti and Koch, 2001; Vijayakumar et al., 2001) and top-down (Breazeal and Scassellati, 1999a) in order to select regions of interest; and systems with active vision, or active perception (Aloimonos et al., 1987; Bajcsy, 1988), where sensorimotor control is used to facilitate the inspection of a particular part of the scene over others, using decision systems that are based on reinforcement-learning (Whitehead and Ballard, 1990; Balkenius and Hulth, 1999), Bayesian techniques (Rimey and Brown, 1994; Knill and Richards, 1996), or other approaches (Rimey and Brown, 1990).

However, every artificial system inherently also has a temporal selection component, a trigger mechanism that signals that the perceptual system has detected a change, novelty, or something of other importance, in the sensed environment. Every system, that is, which is autonomous, situated, continuously faced with data on-line, and is expected to do something useful and meaningful with these data, such as learn, plan, navigate, infer, detect faults, and more. However, in most of the cases this detail is invisible in the architecture and not mentioned by the designers; in some cases this is tied to sensor sample rates, and determined outside the control architecture. In this thesis, attention is used in the temporal sense, and it is treated explicitly. The next section presents different forms of temporal saliency.

2.2.1 Forms of Saliency

The most common type of bottom-up temporal selection mechanisms is one that deals with *event detection*, or *perception of change*. There are many examples of work that contains such a mechanism, and probably many more where this is not explicitly mentioned. It is a simple mechanism for deciding that the sensors are perceiving a change, which might be important, for example corresponding to a state-transition. Some examples are: a change greater than two in the sum of all sensor values of a mobile robot performing general tasks in an open environment (Nehmzow and McGonigle, 1994); a change in direction for a robot moving around a maze (Hayes and Demiris, 1994); a change of 1 bit in a bit-string sensor representation of a mobile robot (Billard and Hayes, 1999); a unit in a saliency map being active for more than six frames in a robotic visual tracking system (Wessler, 1995).

Another type of bottom-up selection is one that deals with *novelty detection*. There is a large body of work that deals with methods for novelty detection — for a good review see (Marsland, 2001, 2003). These include neural network (Kohonen and Oja, 1976; Kohonen, 1984) and statistical (Bishop, 1994) approaches. Most of these approaches, however, operate off-line by building a model of the data, and novelty of an incoming input is then determined on-line by measuring its deviation from the model, but the new input is not incorporated into the model. One class of novelty detection approaches that is more appropriate for on-line systems is the Self-Organising Feature Map (Kohonen, 1982). It is an unsupervised learning approach where units in the map cluster similar experiences, and the system as a whole represents at any given time the current ‘memory’ of what has been perceived so far; a significant deviation from this memory signals novelty.

As well as discrimination between novel and familiar instances, the knowledge of *how familiar* an instance is might also be incorporated into the selection strategy. This corresponds to keeping a count on how many times a particular instance has been seen. One biological equivalent to this is *habituation*, which can be complemented by *forgetting* (dishabituation). Stephen Marsland has incorporated the notions of habituation and forgetting as part of his self-organising architecture that grows from experience, for a mobile robot that collects sonar readings of corridors in an office environment (Marsland, 2001; Marsland et al., 2001, 2002). His architecture is implemented in work reported in this thesis, and will be described in detail in Chapter 3. Other examples of computational implementations of habituation are off-line neural-network architectures by Wang and Arbib (1992) and Kohonen (1984), and biologically-inspired models of associative and non-associative learning mechanisms (habituation, attention, and conditioning) such as the cognitive model of Balkenius (2000) and the spiking-neurons model of Damper et al. (2000).

A different biologically-inspired measure of familiarity is demonstrated by Bogacz et al. (1999), who have implemented a computational model of familiarity discrimination based on findings from monkeys (Brown and Xiang, 1998). It is a Hopfield neural network that calculates an energy measure for incoming patterns, based on the patterns already seen, as recorded through the weights of the network; familiar patterns will produce a high energy, as opposed to novel ones, and this is how novelty is detected. Crook (2000) has implemented and shown the usefulness of this architecture on a robotic platform, where a mobile robot moves along a wall, inspecting novelty in visual images through its on-board camera.

The examples of temporal selection presented so far correspond to bottom-up selection strategies driven purely on previous experiences, whether based on change, novelty, or familiarity. These examples are most relevant for this thesis. However, there are also examples of bottom-up strategies that use the previous experiences to predict or value current and future experiences. A class of such examples is *classical conditioning*, where saliency values are assigned to stimuli based on expectations about when their rewarding goal events occur (Balkenius and Morén, 1999; Hallam, 2001). There are also probability-based approaches that bias selection based on probabilities of future states, calculated from previous ones (Langley, 1997; Sebastiani et al., 2000).

Top-down mechanisms are also useful for temporal selection, for example where selection is biased by the *affect* of stimuli. There are numerous such examples, where affect is modelled in the system through a set of motivations, emotions, needs, drives, etc., which the system tries to maintain at homeostatic levels; significant changes in perception are then detected when one or more of these homeostatic variables exceed their levels (Gadanho and Hallam, 1998; Breazeal and Scassellati, 1999b; Breazeal and Velásquez, 1998).

Saliency Parameters

In all of the examples above, there is an essential parameter that determines what constitutes saliency. This parameter generally corresponds to a threshold that must be exceeded before a change, or novelty, is deemed sufficiently significant; or an input is deemed sufficiently familiar or probable; or an expectation of a reward, or a violation of an emotion, is deemed high enough.

As mentioned by Hayes-Roth (1995) in the quote at the start of this section, perceptual strategy is tailored towards effectively achieving the agent's "currently important reasoning task", which, for the point of view of this thesis, is learning. One might want to design learning systems of different types for different purposes. The purpose that the learning is trying to achieve has a very subjective influence on what constitutes saliency in the perceptions from which the agent is learning. As discussed at the start of Chapter 1, the purpose is usually external to the agent, so the learning must be externally biased towards this purpose. A lot of this bias is in fact given to the system by the designer, who determines what constitutes saliency in the agent's perceptions.

However, as mentioned earlier, in many cases the issue of saliency is not mentioned by the designer, or its effect is not reported. This could mean that a designer has not recognised the dependence of their learning system on saliency, and instead designed a saliency mechanism in an ad-hoc manner; or the designer *has* recognised the importance of saliency detection, but not tested or even considered its effect on the learning system by designing a parameterised mechanism and testing the important parameters. As argued in Chapter 1, a crucial factor for minimising designer effort in influencing the robot's notion of saliency, while keeping this notion faithful to the robot's experiences, is the explicit identification of the important saliency parameters.

2.2.2 Learning at Different Levels of Abstraction

Different forms of saliency were discussed above, and the issue of identifying the important saliency parameters was highlighted. As argued in Chapter 1, the issue of saliency must inevitably be dealt with when considering a robot's sensorimotor experiences for learning. As shown in Figure 2.2, learning corresponds to forming associations between the different modalities of the robot, for example learning motor actions for particular perceptual experiences. Learning could also correspond to forming associations within a particular modality, for example learning a sequence of motor actions. The forming of the associations can occur using data represented at different levels of abstraction, as will now be discussed. Note that the level of

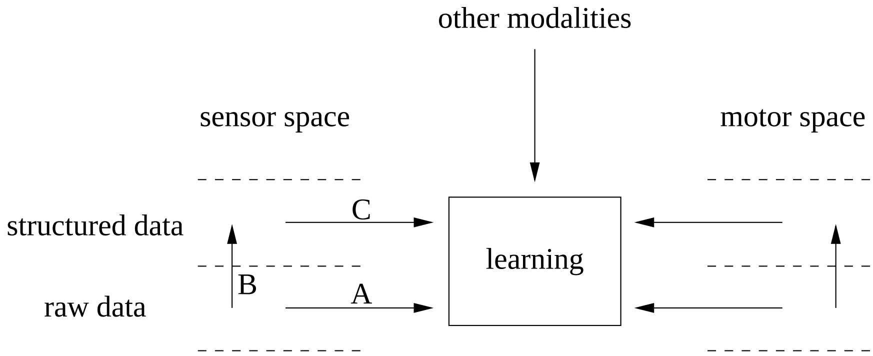


Figure 2.2: Learning at different levels of abstraction. In robotics, learning from the raw sensorimotor data is difficult, and it requires saliency detection (A); some researchers choose to first abstract the raw data by finding structure in them, either through self-organisation which also requires saliency detection (B), or by forcing this structure through design; in both cases learning from the abstracted data is then easier, because saliency detection (C) is easy or even trivial. Saliency and abstraction can be dealt with differently for each of the robot's modalities.

abstraction is not necessarily the same for each modality.

As pointed out in Chapter 1, learning from the robot's unstructured and noisy experiences is very difficult. Because the purpose of learning is to develop new skills, one must provide the robot with mechanisms to generalise information from the raw data that can be used to modify the robot's controller reliably and robustly. For the same reason it is difficult for a designer to reliably specify what constitutes saliency. If one looks beyond the field or robotics, then the machine learning literature (for example, Mitchell, 1997; Hertz et al., 1991) offers many examples that deal with 'raw' data and where the issue of saliency is not of great concern. On the contrary, researchers wish to find important regularities in data produced by some process, and they desire as many data as possible, that is, they have no need for a selection mechanism. Further, in general machine learning success is demonstrated when the system outputs are correct *on average*, whereas robotic controllers need more determinism, especially when it comes to motor control.

A robot could arguably ignore saliency and learn from the raw data, and in simulated low-dimensional data this might indeed be possible though it is very rarely demonstrated. For a real, physical robot with many sensors, saliency detection is generally a non-trivial problem. Researchers address this problem in different ways, depicted in Figure 2.2. As mentioned above, learning from the raw data (A in Figure 2.2) is difficult, so there are many cases where saliency is first used to self-organise or cluster the raw data into discrete structures (B in Figure 2.2); once the structure in the data has been found, the representation used for learning is at a higher level of abstraction. Many researchers avoid the problem of finding saliency in raw data altogether, by forcing a structuring on the data in an ad-hoc manner, rather than through the robot's experiences; thus the data with which the robot learns the task is at a high level of abstraction from the start. Whether the abstraction is achieved through self-organisation or through ad-hoc design, because the learning data are at a higher level of abstraction, saliency (C in Figure 2.2) is trivial or at least easier than at the lower level of abstraction (A in Figure 2.2), as discussed below.

Abstraction Through Self-Organisation

One of the most common techniques for self-organising a robot's experiences is Kohonen's Self Organising Feature Map (Kohonen, 1982); it not only finds representative clusters in the sensorimotor data, but also models the relationships between the clusters corresponding to how similar or dissimilar they are. The map provides a discrete representation of the data, as a set of nodes and edges. The process of fitting a map to the data involves moving nodes in the space to cover dense regions, and the topology in the data is captured through the edges connecting the nodes; thus when a node moves towards a data point, it moves with it the nodes connected to it, according to how similar they are. When this process is complete (as governed by some convergence criteria), instead of dealing with the raw, continuous data, one can simply find the node closest to an incoming data point, and saliency can be detected trivially as a change in node-activation.

Of course, the difficulty of dealing with saliency is in the actual fitting of the map to the data; in this kind of approach, saliency is generally in the form of novelty detection (see Section 2.2.1) — a measure that determines how similar two points in the space have to be to belong to the same cluster. In Kohonen's original implementation, this is implicitly enforced by the designer by specifying the number of nodes to be used in the map, and therefore the level of granularity at which data are considered significantly different. However, there are variations that grow on-line from scratch in response to in-coming data, where the structure of the map, and hence the level of granularity, is not pre-determined. These approaches still need a measure of saliency to determine, for example, when new nodes are required; thus the level of granularity is explicitly controlled by the saliency parameter. A review of these approaches and other clustering approaches is given in Section 3.3, where the method used in this thesis is presented.

Abstraction Through Ad-Hoc Design

As stated above, designers often avoid dealing with saliency in raw sensorimotor data by forcing a structure on the data in an ad-hoc manner. This can be done in many different ways — some examples are:

- coding: the robot's experiences are coded into binary strings, and saliency thus corresponds to a particular number of bits changing their value. Here saliency is not trivial, but it is certainly easier than detecting saliency from the raw (uncoded) data.

- behaviour-based and reinforcement learning systems: the robot's raw experiences are reduced to a set of discrete states and/or actions, and saliency thus corresponds to a trivial change in state or action.

We will see some specific examples in Section 2.3.

2.2.3 Designer Bias and Effort — Reliability and Generality

Different approaches were presented above that deal with saliency for learning at different levels of abstraction (A, B, and C, in Figure 2.2). In each approach, saliency can correspond to any of the forms presented in Section 2.2.1, and thus in each approach a source of bias from the designer exists in setting parameter values. However, designers that force a structuring on the raw data introduce an additional bias, because they are biasing what data are actually used as input to the learning process, and not just the saliency in them.

Increasing the amount of bias by the designer can be advantageous if it leads to more reliable learning and control of the robot. This depends on the designer introducing the bias in a way which is useful and reliable for the robot, and this would generally require a lot of design effort: the designer has to discover what is a useful structure in the robot's perception, or what is a useful measure of saliency; abstracting the motor space requires the careful design of structures that reliably handle low-level control of the robot. These different kinds of design effort can be very useful, if done correctly. However, this large amount of design effort may be so biased towards the particular task and environment that it does not prove useful for other tasks in other environments, where this effort will need to be repeated. As mentioned in Chapter 1, other design biases exist that are not considered in the characterisation of designer effort in this thesis, such as the choice of sensors used by the robot which affect how the robot perceives its environment, and the choice of learning architecture which affects what kind of learning is possible; they too can be over-biased towards a particular task.

There is a tradeoff between designing a system perfectly for a particular purpose, and designing a general-purpose system. This is similar to the over/under fitting tradeoff that exists in general machine learning, whereby one wishes to find a balance between fitting a model to some data very well in order to make it a reliable predictor of these data, but at the same time ensure that this model would generalise to other unobserved data. The thesis argues that the kind of social interactions suggested in Section 2.1.2 can be generic and independent of the task, and can be used to influence the abstraction or the robot's experiences regardless of the task. However, such influence is only possible if the abstraction is not already fixed at design. The thesis therefore suggests to balance designer effort involved in abstracting the experiences of the robot usefully for learning, as discussed in this section, with social interactions of increasing complexities, presented in the previous section. The reliability and generality of the learning system must both be taken into account in finding the right balance. The next section will present related work on socially-interactive learning systems, and identify how the chosen social interactions affect the detection of saliency.

2.3 Characterisation of the Related Work

The previous two sections respectively described the two dimensions that make up the design space introduced in Chapter 1 (Figure 1.1), for characterising the work in this thesis and related work in the field. This section provides the bulk of the literature review in this thesis, and shows how the related work can be organised into the space, as shown in Figure 2.3. It presents work on robotic systems that learn from some sort of interaction with an expert.

As mentioned in Section 2.1, the different types of social interactions are recognised in this thesis as activities with which an expert can *purposely* influence the learning of a robot to different (increasing) degrees. It was claimed in Chapter 1 that this notion of increasing the complexity of the social interactions for the purpose of strengthening the influence on the robot's learning is novel, however individual examples of the different types of social interactions do in fact exist in the literature. The purpose of the review below is to place the existing work in the context proposed here. As discussed in Section 2.1.2, the reasons why an expert should have a more active influence than simply demonstrating a task are related to helping the learner match the expert's actions and thus exposing it more reliably to experiences, and to accentuating the saliency in the demonstrated experiences. The review below is organised based on how researchers have dealt with the first issue, namely the imprecise action-copying by the learner, and the discussion below focuses on the implications that this has for saliency

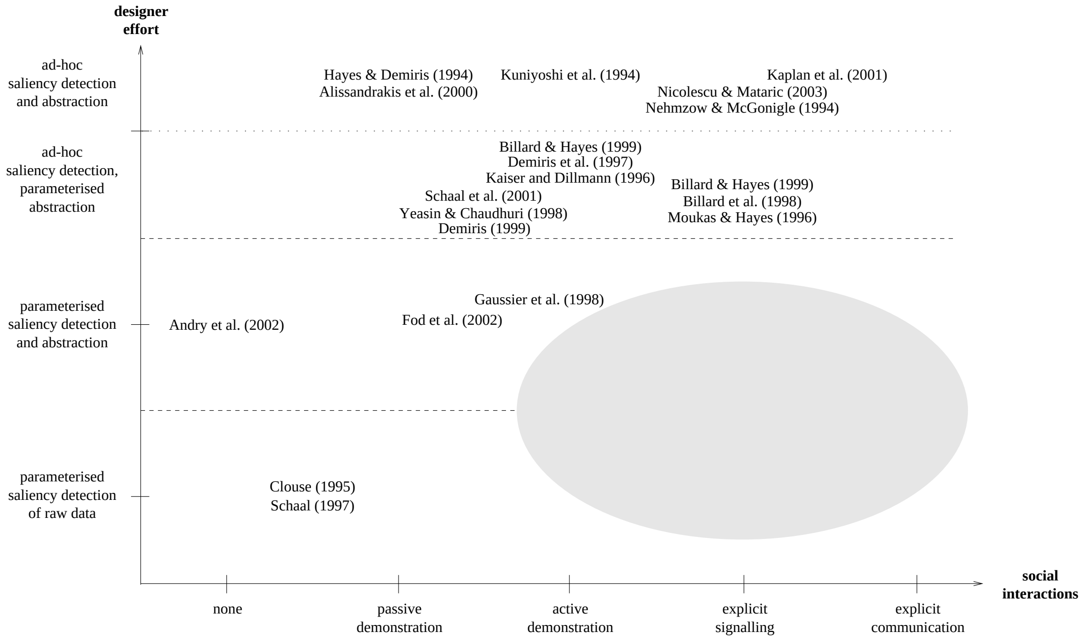


Figure 2.3: Organisation of the literature into the space identified in Figure 1.1, where the same labels are used to refer to the space, except that the top level of designer effort has been split into two categories. Both these categories refer to systems where saliency is treated by the designer in an ad-hoc manner, which means that the effect of saliency detection mechanisms (both categories) and the effect of the designed level of granularity (top category) on learning, are either not recognised, not tested, or not discussed, as opposed to the bottom two levels of designer effort, where saliency is treated explicitly by identifying important saliency parameters and their effect on learning. The thesis argues that the latter is crucial for enabling social interactions to influence saliency and learning, and thus balancing designer effort. Note that within each label on the designer effort axis the groupings are arbitrary. The grey area marks a gap identified in the current research, which is addressed in this thesis.

detection.

With regards to the space proposed in this thesis (Figure 2.3) as a framework for balancing designer effort with social interactions, the review demonstrates an interplay between social interactions and designer effort, in particular:

- the implication of the choice of social interactions for saliency detection, and
- the implication of the choice of level of abstraction for the difficulty of dealing with saliency and for the suitability of the chosen social interactions.

Therefore, references will be made to the types of social interactions described in Section 2.1.2, the forms of saliency described in Section 2.2.1, and the different approaches for abstracting sensorimotor data described in Section 2.2.2 — Figure 2.2 in particular.

Imprecise Action-Copying Mechanisms

In experiments involving mobile robots, an imprecise copying of the expert's actions by the learner corresponds to an imprecise teacher-following behaviour, which results in the learner not following exactly the same path as the expert, or even losing the expert. A similar problem exists in experiments involving other types of robots, for example, a humanoid robot imprecisely copying the actions of a human demonstrator, or even failing completely to copy the actions. The result of imprecise action-copying is exposure to undesirable sensorimotor data, that is, exposure which is not completely as intended by the teacher.

In the maze experiments of Hayes and Demiris (1994), where a learner robot learns in simulation to navigate a maze by following behind a teacher, the simulation consists of discrete states, corresponding to a wall being perceived on either side and at the front, and discrete actions, corresponding to movement in any of four directions; the learning consists of simple if-then rules; the learner detects a significant event when the teacher changes its direction, that is, when the action changes (C in Figure 2.2). Here the demonstrations are passive: due to the restricted maze environment, it is not likely that the learner will lose the teacher, and there is no need for the teacher to highlight saliency because it is trivial.

In similar physical experiments involving two physical robots (Hayes and Demiris, 1994), the learner's perception from the infra-red sensors, and its actions corresponding to the rotation angle required to keep the teacher in view, are coded into discrete values that are used in the same rule-based learning architecture as in the simulated experiments (C in Figure 2.2); here event detection is signalled when the rotation exceeds a given threshold. Because the maze consists of right-angle corners, and these correspond to the salient parts of the environment, the teacher naturally performs significant turns at these corners, which are detected as salient by the learner, and hence the demonstrations are passive even in these physical settings due to the restricted nature of the environment and task.

Demiris has also conducted imitation experiments in less restricted environments. One set of experiments involves a robotic vision head imitating a human's rotational head movements (Demiris et al., 1997). Here, however, the *task* is restricted to horizontal pan movements, that is, only involving one degree of freedom. The robot detects the human's movements using optical flow techniques, which have natural segmentation points when the velocity of the movements is zero, that is, when there is a change in direction. So here the demonstrator inevitably accentuates saliency by simply performing the task (moving his head from side to side). Of course, this relies on the human to demonstrate distinct, continuous segments; such demonstrations can thus be regarded as active, even though the robot only replicates the movement after the demonstration is complete. From the observations of the movements, the salient postures are extracted, and their sequence is learned (B–C in Figure 2.2).

The same architecture is also implemented on a simulated humanoid learning less restricted tasks, involving different hand-postures corresponding to a semaphore code alphabet (Demiris, 1999). These experiments explicitly compare the difference between designed movements, or behaviours, with learned ones. So here, as well as finding salient postures for each movement, there is also a need to match an observed movement with the set of existing known ones, so that novel movements can be learned. A novelty detection mechanism is used to measure the similarity between the observed movement and all the movements already known, and an empirically derived threshold is used to signal that the observed movement is novel. Here it is difficult to say whether the demonstrations have an active or passive nature in terms of accentuating saliency — it depends on the nature of the movements that are programmed for the demonstrator.

As mentioned in Section 2.1.2, Gaussier et al. (1998) recognise the distinction proposed in this thesis between passive and active demonstrations, and the benefit of active demonstrations. They mention that in their physical experiments involving a human teaching a mobile robot various 'dances', the demonstrations are inevitably more adaptive than in similar simulated experiments involving a simulated teacher; the human teacher ensures the learner passes exactly through correct edges in the trajectories of the dance, and that the timings of the learner's actions are more precise. For their learning system this is important, because it takes as input a complete trajectory, so the raw data must be segmented into trajectories that are significantly different from each other; the sequence of trajectories is then learned (B–C in Figure 2.2). They also recognise the importance that the level of abstraction has, because they set up their architecture to detect saliency at different levels of granularity, as controlled through a number of time constants, or parameters.

One of the few examples involving a robotic demonstrator who is specifically programmed to have an active component in the demonstration is given by Billard and Hayes (1999). As mentioned in Section 2.1.2, their use of active demonstrations is for ensuring that the learner does not get lost. A learner robot follows behind a teacher around an environment, but the teacher can also detect the learner and align itself in front of the learner. Two sets of experiments consist of a corridor environment, and an open environment; and the robot learns how to respond to different environmental configurations, both in simulation and with physical robots. Even though the data are coded and therefore more reliable for learning (see Section 2.2.2), at this level of abstraction there is still a sufficient amount of noise due to the imprecise teacher-following behaviour, which led the designers to opt for this kind of active demonstration, probably because passive demonstrations did not work well. Here, this active demonstration is used for reliable exposure, not for accentuating saliency. There is a parameter in the architecture which determines the detection of saliency, which, as argued in Section 2.2.2 is easier at this level of abstraction: it simply corresponds to a change in one of the binary units (C in Figure 2.2).

In the experiments by Billard and Hayes (1999) there is also another kind of learning, involving associations of the learner's perception with a set of symbols that the teacher signals for communicating about locations in the environment. Here the timing of learning is much more precise — it occurs when the signals from the teacher are received; the learning architecture detects these because a signal is trivially salient compared to no-signal, again, corresponding to a change in one of the binary units corresponding to the radio sensor modality. As discussed in Section 2.1.2, this is an example where explicit signals form part of the learning data.

Moukas and Hayes (1996) utilise explicit signals to highlight saliency, where a teacher mobile robot signals to a learner robot when distinct trajectories of a demonstrated movement start and end. The different trajectories correspond to 'dances', the combinations of which form 'words' that are then associated with different food sources. The data from each trajectory are sampled a specified number of times, at regular intervals, and these samples are then self-organised using a Kohonen map with a specified number of nodes, and with a specified neighbourhood function. So although some of the saliency is signalled by the teacher in the form of event detection, a lot of it is still determined by the designer through the level of granularity, as described in Section 2.2.2. The nodes in the Kohonen map are then associated with the location and nature of the food source, using a feed-forward network (B–C in Figure 2.2). Here there is no need for active demonstrations for reliable exposure to sensorimotor data, because the learner does not actually follow the teacher, but rather monitors its movements using an overhead camera.

Imprecise Demonstrations

As mentioned above, the result of imprecise action-copying is exposure to undesirable sensorimotor data. The same problem could also arise from an imprecise *demonstration*. Billard et al. (1998) mention that in their experiments involving a robotic doll learning from observing a human demonstrator, the precision of the demonstrator's movements affects how long it takes the system to learn; it is therefore desirable to have a more active demonstrator. The same architecture is used in these experiments as in (Billard and Hayes, 1999), and therefore saliency is treated similarly.

As mentioned in Section 2.1.2, one of the reasons for tailoring demonstrations to the needs of the learner, in order to deal with imprecise exposure to experiences, is if the learner has difficulties in copying the actions of the demonstrator, which is particularly a problem if it has a different morphology to the demonstrator. In the work by Alissandrakis et al. (2000) involving simulated chess-world agents, even though the system consists of discrete states and actions and thus saliency is trivial (C in Figure 2.2), the issue of different levels of granularity is important, because agents have different 'morphologies' — they follow different rules for moving around the board — and therefore cannot always copy each other's behaviours exactly. Using different imitation strategies they attempt to either copy exact movements, selected landmarks in the movements, or simply the end position, each involving imitating at different levels of granularity. Here, however, the demonstrator does not tailor the demonstration for particular learners, and in fact, there is no particular demonstrator — the experimenters' aim is to test different imitation strategies between different agents.

Difficulties arising from different morphologies are particularly evident in assembly robotic systems that learn from human demonstrations, where the demonstrator does indeed need to tailor the demonstrations to the robot's needs. Yeasin and Chaudhuri (1998) argue that it is crucial to impose constraints on the trajectory of a human operator, because the human's hand has more than 30 degrees of freedom compared to their robot manipulator's six. They also identify that a demonstration must be repeated in order to obtain an 'averaged' smooth trajectory from numerous imprecise demonstrations, thus supporting the need for active demonstrations. Further, their segmentation method relies on motion 'break-points' calculated as velocity changes, which means that the demonstrations can be regarded as active if the human purposely slows down to accentuate these segmentation points. The actual segmentation method is a k-means clustering algorithm, where the number of clusters is determined ad-hoc; as discussed in Section 2.2.2, this determines the level of granularity, and hence saliency (B in Figure 2.2). By placing colour markers on the human's fingers, the hand configurations are segmented, and then somehow mapped onto the robot's motor commands that achieve appropriate grasps, which are then formulated into a plan (C in Figure 2.2).

The work by Kuniyoshi and colleagues (Kuniyoshi et al., 1994; Kuniyoshi and Inoue, 1993) provides another example of an assembly robotic system that learns from observing human demonstrations, but where objects are treated explicitly in the perception of these demonstrations. Here the need to impose constraints on the human's trajectories is also recognised; the allowed movements are in fact restricted to a very small and well defined set: vertical up-down movements. Also, objects are detected through reference to a database of object shapes, and the locations of these objects in the environment are determined before the demonstrations begin; thus the perceptual input to the learning system consists of the collection of objects. Through very clever engineering, event detection is then very simple: the robot can detect the hand and whether it is holding an object, and then an inspection in the vertical direction results either in an object appearing where the hand is expected, disappearing from where the hand originated, or none, corresponding to a 'place', 'pick-up', or 'align' operation. Because the demonstrator only uses precise vertical movements, there is no need for complicated segmentation — saliency detection amounts to simple qualitative state changes (C in Figure 2.2). Since the demonstrations are restricted in order to improve the ability of the system to detect these salient changes, the demonstrations can be regarded as active.

In the above two examples, the assembly system learns from observations of human demonstrations, and therefore the problem of imprecise demonstrations is an obvious difficulty. However, there are also systems that do not have to visually observe a demonstration, because the demonstrations are provided as training data, directly in terms of the system's control architecture, which is usually achieved by controlling the robot with a joystick (Kaiser et al., 1995; Kaiser and Dillmann, 1996). Even though these kinds of demonstrations can be regarded as active because the demonstrator directly manipulates the robot's actions, as mentioned in Section 2.1.2, Kaiser et al. (1995) point out that even they can be imprecise, in that they provide examples that can be contradictory and insufficiently distributed over the input space. They mention that as long as the desired goal is eventually achieved, their system can learn at least some solution, which can later be refined on-line if an adaptation mechanism is provided. However, the latter requires an automated performance measure, which they argue is difficult to obtain. Their learning method consists of a radial-basis function network, which is a set of local receptive fields, whose strengths of attraction to the data are modified during learning, thus achieving a kind of self-organisation (B in Figure 2.2); the number of fields, and their centers and widths, are all determined *a priori*, which affects the level of granularity, as discussed in Section 2.2.2.

Beyond Demonstrations

Some researchers address the problem of imprecise copying of demonstrated actions, or imprecise demonstrations, by not having any demonstrations. Instead, the robot explores the environment using *existing* skills, and the social interactions correspond to an expert encouraging or rewarding the ones relevant to the task. The existing skills could be basic actions, or they could be high-level behaviours.

In the work of Nehmzow and McGonigle (1994), a mobile robot has a default set of motor actions (turn left or right, and move forward), which it tries in turn until one is rewarded by a teacher; a learning architecture then associates the rewarded action with the current perception, and thus the robot learns to perform tasks such as obstacle avoidance, wall-following, phototaxis, and box-pushing. The rewards from the teacher serve not only to signal relevant learning examples, but they also implicitly deal with saliency — they reduce the continual data stream to only those data that are rewarded. However, saliency is also treated explicitly through a trigger mechanism that detects when the robot's perceptions change by some pre-determined amount; although the input data correspond to the robot's encoded infra-red sensor data *and* 'raw' light-dependent (LDR) data, the trigger mechanism is only applied to the encoded data (C/A in Figure 2.2).

In the work by Kaplan et al. (2001), involving the Sony AIBO dog, the robot has a set of high-level behaviours including object-interactions (pushing and kicking a ball), and a teacher trains the robot to sequence these behaviours to form new ones (C in Figure 2.2). Here the teacher not only rewards relevant actions, but also encourages the robot to perform these actions by 'luring' the robot; this can be compared to a teacher-following behaviour in the mobile robots experiments. Similarly to the experiments by Nehmzow and McGonigle (1994), the robot tries different behaviours until one is rewarded, but here there is a representation of similarity of the behaviours in the form of a topology, so there is a sense of order with which the robot can attempt behaviours. This topology can in fact be thought of as representing saliency related to novelty detection, because similar behaviours are close to each other, whereas dissimilar ones are far. The topology is initially programmed by the designer, but it can later be refined by the robot demonstrating what it has learned, and the teacher rewarding only certain experiences. This is the only example found in the literature where a teacher directly influences the robot's representation of saliency.

Nicolescu and Matarić (2003) utilise a mixture of learning from human demonstrations and learning through rewarded (or rather penalised) explorations. In their experiments, a mobile robot learns to move around an open environment with obstacles, picking and dropping objects, by sequencing its existing high-level behaviours. The teacher can either let the robot explore the environment alone and signal to the robot when it is going through an undesirable experience, which is then ignored, or the teacher can demonstrate a particular experience that was missed by the robot, by guiding the robot through this experience. Further, in the demonstrations, the teacher highlights particular experiences by signalling to the robot, and instructs the robot to perform relevant object-related actions (pick-up and drop), which can be regarded as a form of active demonstrations. Thus the explicit signals from the teacher serve both to ensure the robot only learns from relevant experiences, and to signal significant events. However, as mentioned in Section 2.2.2, the issue of saliency is trivial in this high level of abstraction (C in Figure 2.2).

Developmental Approaches

Some researchers approach the problem of action-matching by identifying the need to have basic sensorimotor skills *prior* to social interactions, usually for the purpose of reliable imitation. This is especially the case in work involving imitation of continuous human movements; the motor control required for a humanoid robot to imitate such movements is difficult, and the approach is to first self-organise the robot's low-level proprioception into movement *primitives* (B in Figure 2.2), which would then allow for a reliable imitation. In principle, the robot could then learn a task in terms of these primitives (C in Figure 2.2), thus corresponding to a developmental approach. However, the existing literature does not report any further learning once the primitives are found.

Andry et al. (2002) present experiments where a mobile robot with a manipulator acquires basic skills by autonomously exploring its sensorimotor space. The robot learns how to move its manipulator to locations in its visual field (B in Figure 2.2), and this is later used to 'imitate' a human arm pointing to various locations in the robot's visual field, however, this does not initiate any further learning.

Other researchers provide motion-capture recordings of human movements for robotic systems to process off-line; the systems build movement primitives from the data, and these are then used to imitate a human (Schaal et al., 2001) or simulated (Fod et al., 2002) demonstrator.

It is difficult to say whether the human movements that are used to build the primitives are of a passive or active nature, because they are provided as off-line data, and when they are captured the robot is not always present. However, because it is usually the designers who provide these movements, it is possible that they do so subjectively to help the segmentation (perhaps without realising it). In the experiments reported by Fod et al. (2002), the movement data in fact come from a rather indirect demonstrator: these are data captured from human subjects imitating a human on a video screen, where this second human is the experimenter (Pomplun and Matarić, 2000). Therefore, as opposed to the experiments by Schaal et al. (2001), where the experimenter directly provides the movement data, it is more difficult to say that the demonstrations in the experiments by Fod et al. (2002) are subjective, because the data are influenced only indirectly by the experimenter.

Both groups of researchers utilise parameterised, statistical techniques to segment continuous movement data (B in Figure 2.2). Schaal et al. (2001) utilise a probabilistic learning method called Locally Weighted Regression, where local regression models are fitted to distinct parts of a trajectory. Each of these local models is a receptive field, whose region of validity is determined from the data. As with other clustering methods, like the Kohonen map (see Section 2.2.2), the number of such models is specified *a priori*, which therefore determines the overall level of granularity (even though each one locally models a different level of granularity). Fod et al. (2002) first employ velocity-based segmentation on the continuous movement data, then they project all the segments onto a chosen lower dimension using Principal Components Analysis, and finally they cluster the projected data points using k-means. Each of these stages involves a parameter: determining the velocities that correspond to edges of trajectories, determining a suitable dimension that captures the complexities of the different trajectories, and finally the number of clusters to use. Fod et al. (2002) recognise the importance of these parameters, and discuss their effect.

Combining Social Interactions with Non-Social Learning

Yet another way of addressing the problem of action-matching is to equip the robot with mechanisms to learn on its own, for the situations when it can not match the teacher's actions. Schaal (1997) and Clouse (1995) present examples of systems that learn from a combination of reinforcement learning and learning from demonstrations. In these and similar approaches, however, the expert's knowledge provides a *secondary* source of learning to reinforcement learning, rather than the other way around.

Schaal (1997) reports simulated cart-pole experiments where a reinforcement learning system benefits from an initial demonstration of rewarding experiences from an expert; this initial boost to the system helps to speed up the learning of the system when it subsequently starts to explore its sensor-action space. The simulation experiments of Clouse (1995) consist of an agent learning to navigate through a race-track; the agent can either act on its policy, or execute an action suggested by an automated expert. The results show that it is in fact not beneficial for the agent to completely rely on the expert's advice, because it misses out on valuable negative experiences from self-exploration. Both these systems utilise reinforcement learning methods that operate on continuous real-valued data. That is, they deal with 'raw' (simulated) data, which do not need to be abstracted to a higher level (A in Figure 2.2), except that in (Clouse, 1995) only the state space is continuous — the action space consists of a discrete set of pre-determined actions.

2.4 Summary

The summary of the review given above is shown graphically in Figure 2.3. It shows that much of the work deals with saliency and abstraction in an ad-hoc manner at design time. This means that the effect of saliency detection mechanisms on learning is either not recognised, not tested, or not discussed. In the upper category of the top level of designer effort, saliency is imposed in an ad-hoc manner indirectly due to the designer structuring the robot's sensorimotor data. Again, this means that the designers either do not recognise, test, or discuss the effect of the resulting level of abstraction, or granularity, on learning, and its generality for other tasks. Therefore the way that most designers deal with saliency has the potential of introducing a bias on what the robot can learn, which is not necessarily faithful to the robot's perceptions, and the more effort spent by the designer in biasing the learning for a particular purpose, the less adaptive it is, and the less it can be influenced by social interactions.

2.4.1 Research Gap

The thesis suggests that dealing with these issues, in a manner that is more faithful and adaptive to the robot's experiences, can be achieved by using parameterised saliency mechanisms and increasing the influence of the social interactions on the precision of the experiences and the detection of saliency in them. The thesis will show that the designer then does not need to provide the bias, or at least not as precisely. Some of this responsibility can be transferred to the expert involved in the social interactions: saliency that is accentuated (active demonstrations) or signalled (explicit signalling) by the expert is more faithful because the expert is situated in the environment with the learner, and does not need to know what the robot is actually perceiving. Further, saliency parameters can potentially be tuned (explicit communication) to reflect more faithfully the saliency in the robot's perceptions. The collection of examples from the literature demonstrated these issues to some extent, and they will be shown experimentally in the remainder of this thesis. These issues have not previously been considered, and so the work in this thesis fills a gap in the existing research (marked as the grey region in the Figure 2.3), as discussed below.

2.4.2 How the Work in This Thesis Fits In

The experimental work in this thesis addresses some parts of the gap identified in current research: the utilisation of active demonstrations and explicit signalling for purposely influencing a robot's learning, and thus reducing the amount of designer effort in influencing the learning. The issue of explicit communication is not addressed experimentally, but is discussed as an extension of the experimental work (see Chapter 6). In order to facilitate the balancing of the two sources of influence on the robot's learning, parameterised attention mechanisms are used to deal with saliency explicitly. In this thesis these saliency mechanisms are used for temporal selection. However, the problem of saliency detection could in principle also affect spatial selection, as described at the start of Section 2.2. Thus all the issues discussed in this thesis concerning saliency, such as attention, abstraction, and level of granularity, apply generally to both temporal and spatial selection.

The benefits of increasing the strength of the social interactions are evaluated firstly by addressing the need to deal with imprecise exposure to sensorimotor data due to an imprecise action-copying mechanism, and leading to the need to deal with saliency. The former is tested by considering the effect that the learner's failure to copy the expert's actions (hence learning 'alone') has on learning, and the latter is tested by considering active demonstrations and explicit signalling. Figure 2.4 shows a summary of the experiments presented in this thesis, as a function of social interactions and designer effort, thus addressing the gap identified in the research. The 'alone & social' category refers to situations in which the robot's experiences are considered for learning even when it loses the teacher and moves around randomly in the environment, as opposed to the higher categories of social interactions where such experiences are ignored; the 'alone' category refers to learning from experience obtained purely from a

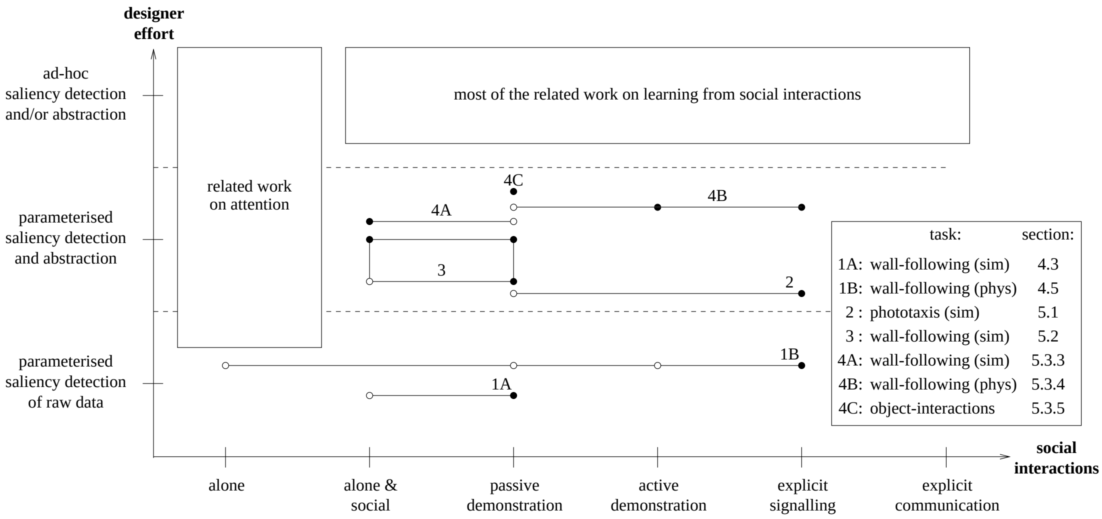


Figure 2.4: The experiments reported in the thesis examine social interactions of different complexities, and learning at different levels of abstractions requiring different amounts of designer effort. The experiments that are actually performed are marked with circles, with a solid circle marking the best performance achieved. Within the second category on the vertical axis, the vertical distinction relates to different amounts of effort required by the designer, except that the distinctions within experiment set 4 are purely for visual purposes. Similarly, the vertical distinctions within experiment set 1 are for visual purposes only.

random wandering behaviour.

The next chapter presents an empirical investigation into the influence that social interactions and designer effort have on the *exposure* of a learner robot to sensorimotor data. Different tasks are used to show that the level of granularity at which saliency occurs is dependent on the actual task and environment, which shows the need for external influence on the learning. Chapters 4 and 5 test the actual learning of these tasks. Chapter 4 investigates learning from the raw data, without any abstraction (A in Figure 2.2), and with modulation achieved through abstraction (B/A in Figure 2.2). Chapter 5 investigates learning from abstracted data (B–C in Figure 2.2).

It is important to note that saliency is treated explicitly only for the robot's *perceptual* modalities, not its motor modalities (see Figure 2.2). This means that attention is applied to the information from the perceptual sensors, and not to the information from the motors. For reliable execution of the task, the motor information is mapped to a high level of abstraction in an ad-hoc manner (consisting of discrete actions: 'turn left','turn right', and 'move forward'), either during learning or during recall of a behaviour. The reason for this is that acting reliably on 'raw', low-level, motor commands is very difficult, at least in the kind of mobile robotics tasks implemented here. In contrast to non-robotics work, where recall is successful if the output is correct *on average*, robot controllers need more deterministic output (see Section 2.2.2). This is an important problem in robotics, which is not addressed in this thesis. For this reason, the output from learning must be mapped to a higher level of abstraction, where 'actions' contain the low-level information needed to produce motor commands in a reliable way. However, as mentioned in Section 2.2.2, all the issues concerning saliency and abstraction can in theory also be applied to the motor modalities. Therefore, although for the kind of tasks used in this thesis motor-command versatility is not essential, the use of attention could in fact also be applied to the motor modalities if such versatility *were* required.

As noted earlier in this chapter, this means that compared to the abstraction of the perceptual data, which is handled by a parameterised attention system, the abstraction of the motor data relies on trial-and-error for empirically obtaining saliency parameter-values that reliably translate the raw motor data into actions, as required for the particular purposes in each of the experiments.

2.5 Performance and Resources

As discussed in Chapter 1, the aim of identifying the design space is not only to characterise the research, but also to guide the design of existing and new systems, by characterising *performance* within the space. The thesis argues that balancing designer effort with social interactions for increasing the generality of a learning system is not at the expense of learning performance. This will be shown by demonstrating how performance varies, and most importantly, how it *improves* within the space, as shown in Figure 2.4. A characterisation of a typical performance surface will be extrapolated from the results in Chapter 6, and the literature identified in Section 2.3 will be revisited for a discussion on the implication of such a performance surface for the related work.

The thesis will also show that the way performance varies is dependent on the available learning resources. In some cases the benefit of influencing the learning is only seen clearly when the robot is forced to manage limited learning resources usefully.

2.6 Biological and Psychological Inspirations

2.6.1 Social Learning

As with other robotic work involving social interactions, the work in this thesis is inspired by the fact that in humans and other animals, a major source of influence on learning emanates from interactions between conspecifics. More interestingly for this thesis, there are different ways in which a novice's learning can be influenced by an experienced conspecific — a model, such as a caretaker. The model's role in the learning process can range from a passive execution of a behaviour, oblivious to the fact that it is being learned from, to more active and interactive training or tutoring, where the model adapts the social interactions to suit the needs and capabilities of the learner.

The study of different learning phenomena arising from social interactions in animals falls into the field of *social learning* (Zentall and Galef, 1988). The study of social learning dates back to the end of the 19th century. Bennet G. Galef, Jr. gives a historical review of the development of this subject in the biological and psychological sciences (Galef, 1988). A major problem that researchers have faced is one of terminology: many different conflicting labels have been used, resulting in a lack of understanding and poor communication in the field. Galef lists some of the labels used, including: imitation, true imitation, allelomimetic behaviour, mimesis, protoculture, tradition, contagious behaviour, social facilitation, local enhancement, matched dependent behaviour, stimulus enhancement, vicarious conditioning, observational conditioning, copying, modelling, social learning, social transmission, and observational learning (Galef, 1988, p. 11). However, regardless of the labels attached to the phenomena, Galef suggests that the overall goal is to understand the ways in which social influences on learning and performance contribute to the development of adaptive behaviour. This is indeed the point relevant for this thesis. The labels most useful for thesis will now be discussed briefly.

Perhaps the simplest form of social learning is *social facilitation*. It refers to situations when it is the simple presence of others that influences the behaviour of an individual. Social facilitation usually results from disinhibition of *existing* behaviours because of reduction of isolation-induced fear (Galef, 1988). For example many species are more likely to exhibit a feeding behaviour in an area containing other members of the same species. Galef claims that social facilitation alone cannot produce social transmission, but together with individual learning it can. This kind of 'social interactions' can be regarded as passive because the model does not even need to know that there is any social transmission going on.

Another type of passive social transmission can occur when an animal's attention is directed to particular stimuli (objects or locations in the environment) through the presence and actions of a model. This is often called *stimulus enhancement*. According to Galef (1988), in stimulus enhancement the probability of exposure to one set of stimuli rather than others is increased through a tendency of the individual to approach conspecifics and from alterations conspecifics have made in the environment. Some examples are rats marking foods they have eaten from, thus drawing conspecifics to eat from the same food (Galef and Beck, 1985), and young rats watching their mothers eating and then 'stealing' a partially-eaten food and eating it themselves (Terkel, 1996). Stimulus enhancement has also been observed in birds (Hogan, 1988; Palameta and Lefebvre, 1985).

Of course, stimulus enhancement could also result from a model actively (*i.e.* purposely and intentionally) drawing the attention of a novice. This leads to a phenomenon observed in human infant-caretaker interactions, called *scaffolding* (Wood et al., 1976). Scaffolding involves a caretaker attracting and maintaining the attention of the child, reducing the degrees of freedom of the task by simplifying it, marking the critical features of the task, controlling the frustration of the inexperienced infant, and most importantly demonstrating the task fully. With scaffolding the learner gets to experience performing the task before it actually knows how to execute it on its own. One of the most interesting of the the components of scaffolding for this thesis is marking critical features: "a tutor by a variety of means marks or accentuates certain features of the task that are relevant" (Wood et al., 1976, p. 98).

However, the most common phenomenon studied in social learning is *imitation*. It is also probably the most debated and disagreed-upon topic of social learning. Researchers have argued about the conditions necessary for imitation, and therefore which species are capable of it. Whiten (2000) claims that some people believe that there is only good evidence for imitation in non-primates, *e.g.* Galef (1988) and Heyes (1993), whereas others believe that there is only good evidence in primates, *e.g.* Byrne and Tomasello (1995), and Meltzoff and Moore (1983). The main debating point seems to be whether the observed 'imitation' is just a copying of a pattern of behaviour exhibited by others, or whether there is also a transmission of higher-level information such as goals. For example Byrne and Russon (1998) suggest a categorisation based on 'program-level' versus 'action-level' imitation, where the former involves copying the basic components of the action (low-level imitation), and the latter involves merely copying the overall strategy and organisation of the action, including sequence and hierarchy (high-level imitation). They use the example of food-copying patterns in mountain gorillas, and show that while there is a large variability in exact hand-configurations and movement patterns used within a family of gorillas, there is almost no variability in the sequencing and general organisation of the low-level components of the action, suggesting that what is important for these animals is the transmission of higher-level information concerning the actions rather than their low-level details.

This debate supports the distinctions made in this thesis that recognise that learning from social interactions can occur at different levels of abstraction. In fact, recent findings in neurophysiology have exposed neurons that are believed to form the fundamental basis for imitation in primates — 'mirror neurons' (Rizzolatti et al., 2000); they are active both when a monkey observes a demonstrator interacting with an object, and when the monkey executes the same interaction. What is interesting is that these neurons are very specialised to respond to specific types of grasps, and further — they seem to specialise at different levels of granularity: some neurons discharge only to specific finger configurations of specific grasps, others discharge to a specific grasp regardless of finger configuration, and yet others discharge to the achievement of a goal, regardless of the way this is achieved. Thus there is an inherent level of granularity in what these neurons represent and the issue of saliency must be somehow incorporated in their activation. These findings have inspired one of the implementation reported in Chapter 5 in this thesis, where more biological details are provided (see Section 5.3).

2.6.2 Attention

Attention is identified in this thesis as responsible for detecting saliency and thus abstracting from raw sensorimotor data. This means selecting perceptions to process based on their saliency, and keeping some higher-level (abstracted) representation of these perceptions. These features are all inspired from the following psychological concepts.

The selective role of attention is to act as a bottleneck, or filter, to select certain stimuli over others (Kahneman, 1973). Some interesting questions then arise, such as at what stage of processing does selection occur? 'Early' theories such as the Filter Theory (Broadbent, 1958) propose that selection occurs just prior to perceptual analysis which involves extraction of the physical attributes of a stimulus and its identification. In other words, stimuli are selected before they are analysed; information is thus analysed in sequence where the processing of one stimulus must be terminated before a new one can be processed. 'Late' theories (such as Deutsch and Deutsch (1963)) propose that selection occurs just prior to response selection, *i.e.* after perceptual analysis; multiple stimuli can therefore be analysed in parallel, but only one response can be selected at any one time.

This issue is not important for attention in this thesis because it does not need to deal with multiple stimuli. Rather, attention simply needs to decide whether to select the current perception or not. This is achieved in this thesis by determining the saliency of perceptions, and is inspired from how humans react, respond, or orient to significant events, and further — by how familiarity of events affects these reactions by inhibiting them and thus modulating perceptual processing.

Orienting seems to be one of the key activities involved in attention, since it is responsible for keeping relevant information in a useful, convenient focus, for further analysis. As Posner and Peterson (1990) claim with respect to visual orienting, "foveating a stimulus improves efficiency of processing targets ... " (p. 27). Orienting (and foveating) can occur either overtly through saccades, or covertly without any movements at all. The Orienting Response (OR), also referred to as the Orienting Reaction and Orienting Reflex, is a pattern of physiological responses which is elicited by novel or significant stimuli (Kahneman, 1973). It was first introduced by Pavlov in 1910, and later by Sokolov (1963) who claimed that the OR is elicited through a *neuronal model* of a stimulus which is used to match incoming stimuli and hence determine their novelty. The neuronal model is built up from exposure to the stimulus, and after many presentations it habituates and no longer responds to this stimulus. Therefore the neuronal model forms an important part of the selection mechanism, and it can also be thought of as the abstraction of the stimulus.

The role and nature of habituation is very intuitive and straight-forward: it inhibits the OR to a stimulus incrementally as the stimulus is presented repeatedly (Wang, 1995). What is more interesting is how the OR is reinstated (dishabituation); Balkenius (2000) provides a good summary of some of the factors responsible, such as: a stimulus change, the passage of time (forgetting), a new stimulus, a new context, and drowsiness.

If indeed stored representations (neuronal or otherwise) of stimuli are matched and compared with incoming stimuli, what is the actual process of comparison, and how is the result of this comparison evaluated? In Section 2.2.1 it was argued that all robotic learning systems must have a temporal selection component which usually corresponds to a detection threshold. Perhaps an equivalent psychological explanation can be found in Signal Detection Theory (Green and Swets, 1966; Macmillan and Creelman, 1991), which claims that people judge the significance of stimuli based on a certain criterion, and this criterion is applied at a location of the observer's choice. The theory relates this choice behaviour to a psychological *decision*

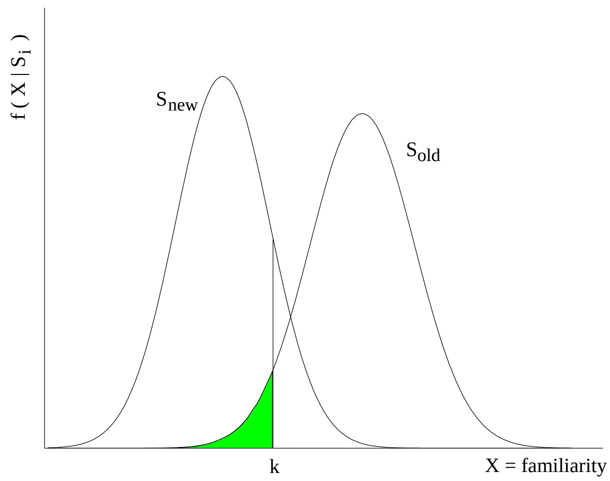


Figure 2.5: The decision space of Signal Detection Theory. The items in the 'new' distribution have low familiarity, whereas the items in the 'old' distribution have high familiarity. The shaded region is the error rate corresponding to designating a stimulus as 'new', when it is in fact 'old', and it corresponds to the Type I error of statistical hypothesis testing.

space, consisting of two conceptual distributions, for familiar (old) and unfamiliar (new) items, as shown in Figure 2.5. The space consists of two sampling distributions of 'familiarity', S_{new} and S_{old}; the density function of each distribution is given by $f(X|S_i)$, where X is a measure of the familiarity of the stimulus. The threshold k is regarded as the decision criterion; if familiarity is lower than this threshold, the observer designates the stimulus as 'new', with some unknown error; however if the stimulus is in fact 'old' (shaded region of Figure 2.5), this error is equivalent to the Type I error of statistical hypothesis testing (rejecting the Null Hypothesis when it is in fact true).

The different components of attention, such as those discussed above (selection, orienting, habituation, and detection) are sometimes considered to form part of an information-processing system, and they are believed to be affected by both top-down and bottom-up factors (Cowan, 1988; Rensink et al., 1997; van Reekum and Scherer, 1997; Kahneman, 1973, and many others). Different examples of bottom-up and top-down attention in robotic systems were presented in Section 2.2. It was also argued in that section that attention is mostly treated as a spatial mechanism, rather than a temporal one as used in this thesis. This is also true for psychological experiments on human attention. Such experiments are generally based on the classic target detection experiments by Posner et al. (1980), involving visual targets appearing at peripheral locations, immediately preceded by visual cues that either correctly or incorrectly predict the targets (some more examples are Husain and Kennard, 1996; Desimone and Duncan, 1995; Behrmann and Haimson, 1999).

Pashler (1998) points out that in experiments of this type the effects of temporal cues can sometimes reflect changes in criteria for producing responses. However, the role of temporal information in orienting attention is relatively unexplored, as recognised by Coull and colleagues (Coull and Nobre, 1998; Miniussi et al., 1999; Coull et al., 2000). In a series of brain experiments on humans they separate spatial effects from temporal selection and show that temporal information is used flexibly and actively to process stimuli. In their findings it appears that spatial and temporal orienting activate a common network of frontoparietal regions, but that this overlap is not complete, suggesting that the two activities share some general 'attention' functionalities, but also distinct ones. They also investigate and confirm a distinction between voluntary, top-down orienting, and automatic, bottom-up orienting.

Chapter 3

An Empirical Investigation of Exposure to Data

As mentioned in the previous two chapters, the aim of this thesis is to investigate a balance between biasing saliency through design and through social interactions, so that learning can be more general, adaptive, and faithful to the robot's experiences[1]. This chapter starts the investigation by giving an empirical account of a learner robot's experiences, using different tasks and environments in order to show that saliency is particular to the task and environment. The chapter does not actually show how the robot learns from its experiences, but rather inspects these experiences with the above considerations in mind, and discusses the *implications* for learning.

Section 3.1 specifies the different types of social interactions that are under investigation in this chapter. They are specifically chosen to investigate the problem of imprecise exposure to experiences due to imprecise copying of the expert's actions by the learner, and they will be used to motivate the use of active demonstrations. Social interactions of a higher complexity are presented in the next chapter.

Also under investigation in this chapter are different levels of designer effort involved in specifying what constitutes saliency. The first and main distinction related to designer effort, which was recognised in Chapter 2 and is investigated here, is whether a low level of abstraction is used for learning, where the data are noisy and unstructured, or a higher level of abstraction is used, where a structure exists. This chapter investigates two such levels of abstraction, respectively in Sections 3.2 and 3.3. A second distinction that was recognised in Chapter 2 is

[1]See start of Section 1.1 on page 3 for the definition of 'saliency'.

whether the structure is imposed by the designer in an ad-hoc manner, or discovered through self-organisation where saliency is treated explicitly. The former does not contribute to the aim of this thesis mentioned above, and therefore it is not considered. Section 3.3 presents an attention system that models saliency explicitly to self-organise the raw data into distinct structures, and these structures are then compared with the raw data inspections performed in Section 3.2. The issue of designer effort is considered further in Section 3.3 through the identification of different ways in which the attention system can be used for learning, requiring different effort involved in abstracting the raw data and therefore biasing saliency. Section 3.4 discusses how the objectives of this chapter are met, and how they contribute to the rest of the thesis.

3.1 Imprecise Exposure to Data

The previous chapter discussed a general difficulty for socially-interactive learning systems that learn from demonstrations from an expert. This difficulty is related to imprecise copying of the teacher's actions by the learner, or imprecise demonstrations by the teacher, both of which lead to an imprecise exposure to experiences by the learner (see Section 2.3). The reason this is a problem is that a learning architecture might have difficulties in generalising from such experiences.

Most of the experiments in this thesis involve a mobile robot following behind a teacher. In simulation experiments the teacher is also a mobile robot, identical to the learner; in physical experiments the teacher is a human. In these experiments, imprecise copying of the teacher's actions corresponds to an imprecise teacher-following behaviour. There are two sources of imprecision. First, the learner does not follow exactly the same path as the teacher, because the teacher-following behaviour drives it to cut corners. Second, the detection of the teacher is noisy, and because the teacher does not detect the learner, the learner can sometimes completely lose sight of the teacher and be 'lost' for a period of time until it finds the teacher again.

3.1.1 Testing-Scenarios for Evaluating Problems with Passive Demonstrations

The first source of imprecision is tested by emulating an ideal, noise-free, teacher-following behaviour, by equipping the learner with a hand-crafted behaviour corresponding to the particular task that is examined; the robot moves alone in the environment with this behaviour. In the simulation experiments the hand-crafted behaviour is the same behaviour that controls the teacher, and therefore the learner takes exactly the path that the teacher would have taken. The second source of imprecision is tested by using a 'social facilitation' flag that is on whenever the learner loses the teacher; comparisons are then made between the two situations where the robot's experiences are either ignored when this flag is on, or all the experiences are considered. The reference to 'social facilitation' is inspired from the social learning phenomenon described in Section 2.6 as social transmission that relies on the presence of another individual. A final comparison is made, emulating a worst case teacher-following behaviour with the robot moving around alone in the environment with a random wandering behaviour.

To summarise, the following scenarios are tested in the experiments:

1. **hand-crafted**: the learner is equipped with a hand-coded behaviour;
2. **following with social facilitation**: the learner follows behind a teacher who is performing the task, and does not consider input perceived while the teacher is lost;
3. **following**: as above, but perceptions are always considered;
4. **random**: the learner moves around randomly in the environment.

Scenarios 1 and 4 are used as upper and lower baselines against which to evaluate the imprecision of the teacher-following behaviour.

3.1.2 Active Demonstrations in the Physical Experiments

In the physical experiments, the first source of imprecision, that is, noisy detection, is in fact a major problem. The implementation of the robot's human-tracking system is not robust enough to allow the robot to follow a human demonstrator if the human is moving freely and independently of the robot, that is, if the human is passively demonstrating the task. It was identified that the demonstrator must adapt his movements and speed to help the robot's tracking system, that is, demonstrate the task actively. Therefore, in the physical experiments the robot does not lose the teacher for significant periods of time because the teacher responds to these situations by moving back into the robot's field of vision.

Of course, this problem is an implementation problem, rather than a theoretical one. That is, one could in principle improve the robot's tracking system to be more robust in order to track and follow a human perfectly. However, it was decided that active demonstrations are useful for other purposes, mentioned below, and that it would suffice to emulate a perfect teacher-following behaviour with a hand-crafted behaviour, as explained above. Therefore, no extensive attempts were made to improve the tracking system, and instead, active demonstrations were introduced.

The physical experiments will show that active demonstrations are useful not just for ensuring the robot does not get lost, but also for accentuating the saliency in the robot's experiences. This then suggests that active demonstrations should also be attempted in the simulation experiments. But programming an active robotic demonstrator is difficult, because the designer has to guess what would be a good active strategy before the robots start interacting, as opposed to a human demonstrator, who is situated in the same environment as the robot, and can therefore tailor the demonstration 'on-line' in response to what the robot is doing. This thesis argues that if one has the capability to provide a learner robot with active demonstrations, then this is advantageous. However, if this requires additional effort from the designer, then that is a consideration that needs to be taken into account.

3.2 Raw Data Inspections

In this section we will examine the characteristics of sensory data as *external observers* — we will look at the statistical properties of *complete* datasets of perceptions, in order to identify structure in the perceptual data. This kind of statistical analysis of the data is only suggested as a methodology for learning about the bottom-up nature of the robot's perceptions, rather than as a tool with which the robot itself can find the structure in its perceptions. The latter issue is considered in Section 3.3, where an attention system is presented for processing data on-line, rather than complete datasets.

The motivation for exploring the robot's raw perceptual data is that it provides the designer with a good starting point for assessing how well-structured the data are, what might constitute saliency in them, and the implications of the social interactions on how well the robot might be able to learn from the data. Further, the results from the statistical analysis performed here can be used to evaluate the performance of the attention system presented in Section 3.3.

3.2.1 Method

The statistical tool we will use is Principal Component Analysis (PCA), which is a multivariate data analysis tool useful for dimensionality reduction (see, for example, Venables and Ripley, 1994; Afifi and Clark, 1996). PCA finds the components $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m$ that explain the maximum amount of variance in an m-dimensional dataset, X, of size n, as follows.

The sample covariance matrix[2] of X is

$$W = \frac{1}{n-1}(X'X - n\overline{\mathbf{x}}\overline{\mathbf{x}}') \tag{3.1}$$

where X' is the transpose of X and $\overline{\mathbf{x}}$ is a row vector of the means of the columns of X. W is a symmetrical matrix and therefore its singular value decomposition is

$$W = U'LU \tag{3.2}$$

where U contains the real eigenvectors of W, and L is a real-valued nonnegative diagonal matrix containing the eigenvalues of W, λ_i, such that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0 \tag{3.3}$$

The principal components (PC's) $\mathbf{s}_i$ are then simply the column vectors in U, and a linear combination of the first k PC's can be used to project the original dataset onto a lower dimension $k < m$, using a simple matrix multiplication $Y = X * [\mathbf{s}_1, \ldots, \mathbf{s}_k]$, where the resulting Y is a $n \times k$ matrix. The amount of variance accounted for by each PC is proportional to its corresponding eigenvalue, and can thus be calculated relative to the sum of all eigenvalues (which is the total variance of the dataset):

$$\frac{\lambda_i}{\sum_{j=1}^{m} \lambda_j} \qquad i \in \{1 \ldots m\}. \tag{3.4}$$

More useful is the amount of variance accounted for by the first k PC's, calculated as

$$c_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{m} \lambda_j} \qquad k \in \{1 \ldots m\}. \tag{3.5}$$

Because of the ordering of the λ_i (Equation 3.3), the most representative one-dimensional projection of the data is given by $\mathbf{s}_1$, and the most representative k-dimensional projection is given by $[\mathbf{s}_1, \ldots, \mathbf{s}_k]$, *i.e.* by the first k PC's. As we use more and more PC's the relative variance c_k increases, until we can explain all the variance in the data by using all m PC's. One can plot the relative variance accounted for by the first k PCs for all values of k — such a plot is called a scree plot. An example of a scree plot for a 50-dimensional uniformly-distributed dataset with 1000 points is given in Figure 3.1.

[2] *c.f.* the sample variance of a random variable $\mathbf{x}$, which is $\frac{1}{n-1}(\Sigma x_i^2 - n\bar{x}^2)$

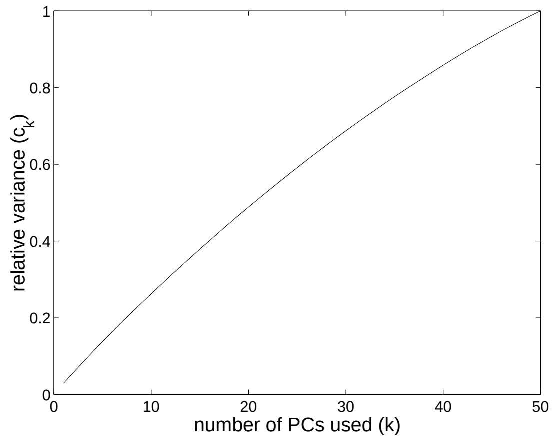


Figure 3.1: A scree plot for a uniformly distributed dataset of 50 dimensions.

The value for k is chosen so that the projected dataset Y is of a conveniently low dimension — useful for example for visualisation, as we will see in the remainder of this section — while accounting for as much of the variance as possible.

A Simple Example

The use of PCA will now be demonstrated with a very simple example. First let us introduce Gillespie, the robot that is used in experiments throughout this thesis, shown in Figure 3.2. Gillespie is a Real World Interface (RWI) B21 robot, with multiple sensory modalities: a layer of 24 sonar sensors, 2 layers of 24 infra-red (IR) sensors, and a video camera mounted on a pan/tilt head; it also has a compass and bump and tactile sensors, although these are not used in the experiments reported in this thesis. The input to attention and learning in all the experiments in this thesis is made up of 20 of the sonar sensors around the front of the robot, as shown in Figure 3.3. The back four sensors are not used in the makeup of the input[3].

We will test PCA by exposing Gillespie to three different perceptual states, captured through 20 of the sonar sensors, so the input space has 20 dimensions. First, Gillespie is placed in the middle of the room with no objects around it for a short period of time, and data are collected from its sensors; then it is placed parallel to a wall on the left side, about 10cm from the wall,

[3]The reason for this is firstly that these sensors will not be needed for executing the task that Gillespie will be trained to perform, namely wall-following (see Section 3.2.2) — the robot will not be expected to sense walls behind it. Secondly, the back sensors were not used so that the experimenter could walk closely behind Gillespie and inspect various outputs without interfering with the robot's perceptions; output is displayed on the laptop computer mounted just above the back four sensors.

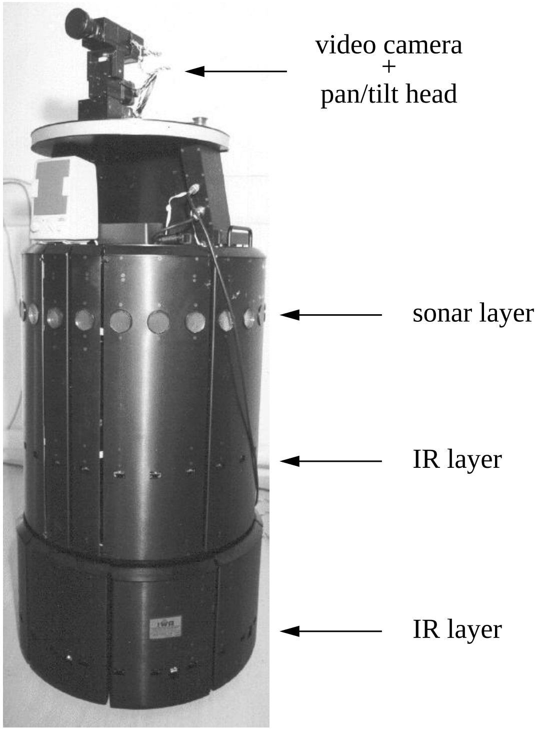


Figure 3.2: The RWI-B21 robot, Gillespie, has a layer of 24 sonar sensors, 2 layers of 24 infra-red (IR) sensors, and a video camera mounted on a pan/tilt head; it also has a compass and bump and tactile sensors, although these are not used in the experiments reported in this thesis.

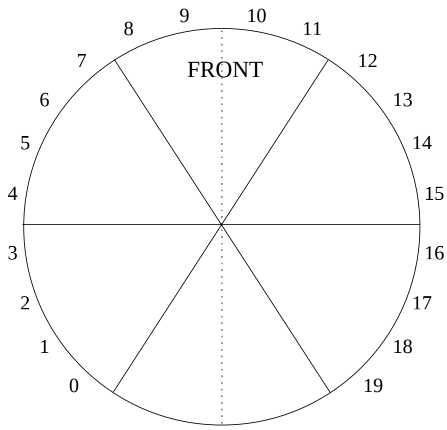


Figure 3.3: A schematic diagram of the locations of Gillespie's sensors that are used as input to attention and learning in all the experiments in this thesis.

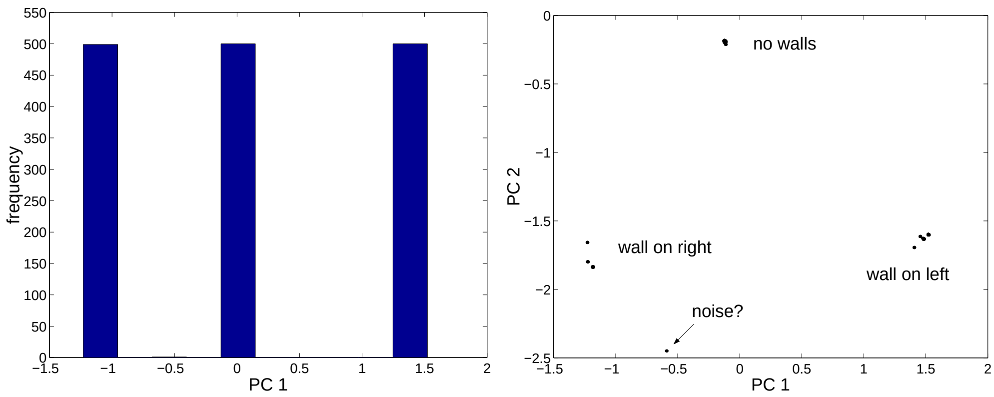


Figure 3.4: Projection of all the data from the simple example onto one dimension (left), and two dimensions (right).

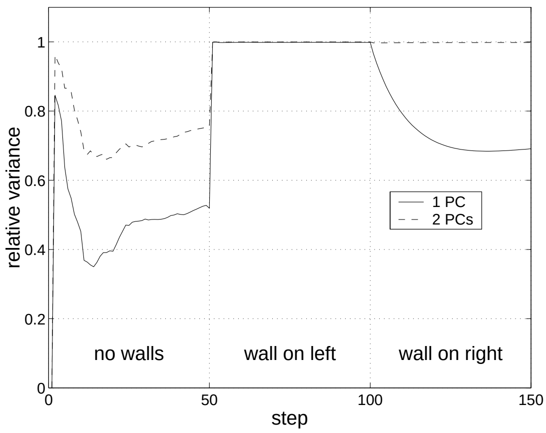


Figure 3.5: Convergence of the eigenvalues for the simple example.

for an equal amount of time, and again, data are collected; and finally it is placed parallel to a wall on the right side, again 10cm from the wall and for an equal amount of time, during which data are collected; the robot does not move during data collection. We accumulate the perceptual input at each step into a dataset, so that at the end of the test we have a complete dataset of Gillespie's perceptions. We project the final dataset onto the first PC, and the first two PCs, shown in Figure 3.4, and also plot the relative variance accounted for by these PCs for the duration of the test, *i.e.* we calculate c_1 and c_2 at each step using the dataset accumulated thus far, in Figure 3.5. Thus the relative variances of the projections in Figure 3.4 are given by the final values shown in Figure 3.5.

Figures 3.5 and 3.4 are interpreted as follows. Initially, PCA is exposed to a random cloud of points, similar to the dataset used in Figure 3.1 but where the distribution is that of the noise in the sensors; one PC accounts for only about 50% of the variance, two PCs account for about 75%. As soon as PCA starts to see points for 'left wall', most of the variance accounted for is now between the two clusters, rather than within the first ('no wall') cluster. Therefore the relative variance increases rapidly to 1, and in fact we see that one dimension is sufficient at this stage to explain all the variance in the data; this is because there are only two clusters. When data start appearing for 'right wall', the ability of one dimension to explain the variance decreases (steps 100–150 in Figure 3.5), whereas two dimensions can still explain the variance in the data perfectly.

These interpretations can be taken further. At step number 50 there is a very abrupt change in the level of granularity at which variations are considered significant. In the first 50 steps, the robot is exposed to a cloud of points corresponding to no wall perception; the variations in the data are due to the noise in the sensors, and then even two PCs cannot capture these variations well. However, after step 50, the robot is exposed to completely different perceptions, and now the original variations in the data (before step 50) are considered insignificant; the notion of significant, or salient, variations has changed; differences are detected on a larger scale. Note that at step 100, the scale, or level of granularity, does *not* change; two PCs still capture the variations perfectly, and one PC gradually captures them less well (as opposed to the abrupt change at step 50) — this simply means that in one dimension the data become less well-structured as the third cluster appears.

Note that the fact that the data are better structured in two dimensions than in one is not very useful on its own, because it is the case in general that it is more difficult to detect structure when projecting onto a lower dimension than onto a higher one. However, knowing that data can be represented well in a particular low dimension is informative for comparative purposes. The particular dimension gives an idea of the complexity in the data, for example, the simple toy example here suggests that although the dimensionality of the data is 20, there are strong correlations between the sensors and in fact two dimensions are enough to account for the significant variations in the data. As we will see in Section 3.2.2, when the data come from exposure to a task, the low dimension might correspond to the complexity in that task. Then, comparing exposure for this task under different conditions can be very informative: if under some condition the variance accounted for by the chosen dimensions is lower than in another condition, this means that there are variations that are not related to the task, or equivalently, that the perceived data are less well-structured with respect to the task. This is therefore a useful comparative methodology for a designer wishing to find the conditions under which the data are better structured, which gives some measure of complexity in the data (for example, number of required dimensions, and relative variance).

The designer can get more insight into the data by looking more closely at the projection space suggested by PCA. One could inspect what the principal components for the complete (final) dataset actually are. Recall that the data are projected onto a lower dimension through a linear combination of the PCs, so it might be interesting to inspect which are the components most influential in this projection; this gives an indication of which are the important sensors that contributed to the calculation of the PCs[4] . Because the PCs have the same dimension as the data, this interpretation can be difficult for a high-dimensional dataset such as this one, but this is nevertheless an interesting exercise. The first three PCs calculated by PCA for the simple example are shown in Figure 3.6; see Figure 3.3 for the locations of the sensors on the robot (it seems that there was something wrong with sensors 4, 7, and 15, because they did not contribute to any of the PCs; this could have happened if these sensors happened to be at a critical angle where they did not receive any of the sonar reflections, and therefore have a zero input; this problem does not exist when the robot is in motion, as we will see in the next section).

From Figure 3.4 we see that the first PC mainly distinguishes between sensing the wall on the left and on the right. This is confirmed when we inspect the values of the first PC in Figure 3.6: the values corresponding to the left of the robot (values 0–9) have an opposite sign to those corresponding to the right of the robot (values 10–19); this PC can be thought of as capturing the contrast between left and right, and according to the relative variance calculation, c_1 (see the final value of the solid curve in Figure 3.5), this accounts for about 70% of the variance in the data. Similarly the second PC seems to distinguish between sensing the wall or not (see Figure 3.4) and by inspecting the PC in Figure 3.6 we see that it seems to correspond to the total signal in the sensors (all values have the same sign). Therefore, the second PC represents how much the wall is sensed (*i.e.* how close the robot is to the wall), whereas the first PC represents on which side the wall is sensed; together these two PCs represent almost all (99.8%) of the characteristics of this dataset, as calculated by the relative variance calculation, c_2 (see Figure 3.5). The third (and any other) PC corresponds to the noise in the sensors. Further, from Figure 3.4 we see that PCA has designated the 'no-wall' state as the origin in

[4]Many thanks to Prof Murray Aitkin for suggesting and helping with the numerical and analytical inspections of the principal components throughout this chapter.

	PC 1	PC 2	PC 3	
0:	0.079744	-0.314937	0.218893	
1:	0.242706	-0.211609	-0.356506	
2:	0.313199	-0.184187	0.164485	
3:	0.323470	-0.209022	0.178676	
4:	0.000000	0.000000	0.000000	left
5:	0.307507	-0.200411	-0.432462	side
6:	0.303194	-0.195919	0.165539	
7:	-0.000000	-0.000000	-0.000000	
8:	0.215967	-0.139555	0.117508	
9:	0.193080	-0.126529	-0.494709	
....	....	....	....	
10:	0.193937	-0.125319	0.105838	
11:	-0.031881	-0.346442	0.136648	
12:	-0.016489	-0.409397	0.161531	
13:	-0.222327	-0.192879	-0.477697	
14:	-0.283702	-0.188561	0.002292	right
15:	0.000000	0.000000	-0.000000	side
16:	-0.281897	-0.314941	0.055543	
17:	-0.273142	-0.305104	-0.070959	
18:	-0.270547	-0.220679	0.016877	
19:	-0.255960	-0.217536	0.021592	

Figure 3.6: Three of the principal components calculated for the simple example.

PC-space, *i.e.* the point (0, 0). We will see similar PCs appearing in more interesting situations when the robot is in motion, in the next section.

In conclusion, this 20-dimensional dataset can be explained perfectly with two dimensions. To further demonstrate this we mix the dataset and hence present the data points to PCA in a random order; the same clustering is achieved (and therefore not shown), and the convergence of the relative variance is now shown in Figure 3.7; c_1 and c_2 converge to the same values as before, but much faster.

Note the emergent clustering effect of PCA, demonstrated in Figure 3.4. Looking at these clusters is useful for interpreting what the robot is perceiving, especially when it is difficult to interpret the principal components. One can be confident that an attention system will be

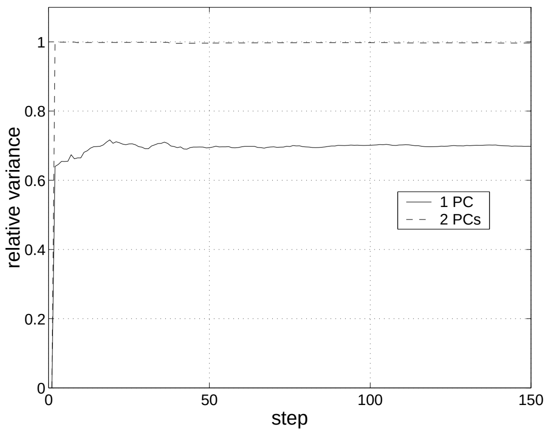


Figure 3.7: Convergence of the eigenvalues for the simple example, where the ordering of the data points is randomly chosen.

able to find structure in a perceptual dataset such as this one quite easily, because the clusters here are very distinguishable in a low dimension. We see that already in this over-simplified example there is a bit of noise (belonging to the 'right wall' cluster); unfortunately this problem is magnified in the real experiments and makes both the interpretation and the attention process harder; this is the challenge of modelling real sensory data from a robot.

3.2.2 Robotic Tasks

The remainder of this section will involve analysing data obtained from exposure to particular robotic tasks. The differences between types of social interactions will be discussed, as well as the implication for attention. As in the simple example above, a dataset of perceptions is accumulated as the robot is exposed to the environment by a demonstrator, and we look at both the convergence of the relative variances of the chosen PCs, and the projection of the final data onto those PCs.

Wall-following

The example given above is an extremely simplified version of a wall-following task: the three main states are being parallel to a wall on either side, and not being near a wall at all. We will now examine this task more closely, both in simulation and with Gillespie. Figure 3.8 shows the environments used in these experiments.

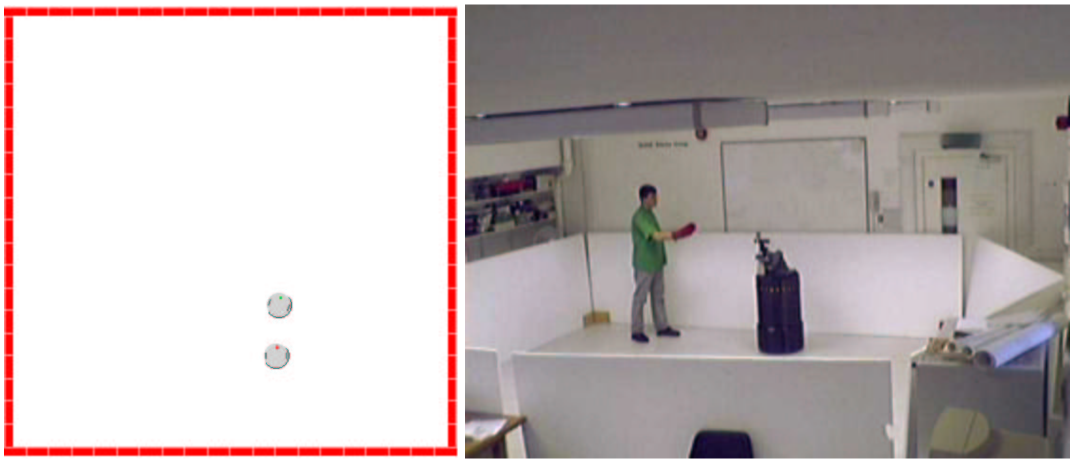


Figure 3.8: The simulated (left) and physical (right) environments used in the wall-following experiments.

The platform used for the simulation experiments is the 2D Khepera simulator (Michel, 1996). In this environment there are two robots, a learner and a demonstrator. Each robot perceives information using six infra-red sensors around it, and moves around using two motors, as shown in Figure 3.9; the sensors can perceive ambient light, object-distance information (*i.e.* the presence of objects), and robot-distance information (*i.e.* the presence of other robots). The learner follows behind the demonstrator using a hand-coded teacher-following behaviour; the demonstrator is constantly moving around the environment executing a particular hand-coded task; the learner's perception of the demonstrator is only used for the teacher-following behaviour, *not* as part of the perception of the task.

For the wall-following task, the robots perceive *distance* information through their IR sensors. The demonstrator executes the task: it wanders randomly in the environment until it finds a wall; it follows the wall, *i.e.* moves parallel to it; at regular intervals an 'interrupt' is signalled which forces the demonstrator to move away from the wall and hence wander again. The interrupt is used to ensure that the demonstrator exposes the learner to the full complexity of the task, rather than follow the wall only on one side for the duration of the run. A run here consists of 50000 steps, with the interrupt signalled every 5000 steps.

The physical experiments are very similar in nature to the simulated ones. They involve Gillespie and a human demonstrator; the robot is programmed to detect and follow the human using its on-board video camera, which is done using a simple colour-tracking algorithm — the

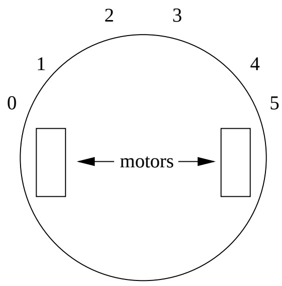


Figure 3.9: A schematic diagram of the Khepera mobile robot; it can perceive distance, ambient light, and the presence of other robots, using six infra-red sensors (numbered 0–5); it moves using two motors.

demonstrator wears a green shirt which is easily detectable[5] . The arena is an approximately 5m $\times$ 5m square (see Figure 3.8). The demonstrator moves around the arena following walls, turning into the middle of the arena occasionally (4-5 times during the run) as with the 'interrupt' in the simulation. Gillespie perceives through 20 sonar sensors around its front (it does not perceive through the four sensors at its back), as shown in Figure 3.3, which in practice are not affected by the presence of the demonstrator; as mentioned before, the images from the camera are *not* used as part of the input. A run here consists of 10000 steps.

The four testing scenarios mentioned in Section 3.1 (hand-crafted, following with social facilitation, following, and random) are used here. Note that in the second scenario the dataset obtained is smaller, because experiences are ignored when the learner loses the teacher. In the plots presented below, this difference is approximately 3000 steps in the simulation (although it can go as high as 10000), and approximately 500 steps in the physical experiment.

Figure 3.10 shows the convergence of the relative variance accounted for by two PCs in the simulation and physical experiments, for the four different scenarios. As mentioned above, the length of the dataset from the second scenario is actually smaller than the rest, however in Figure 3.10 the corresponding curve is 'stretched' to fit in the plot.

In both experiments the quality of the structure in the data is ordered as we expect: the data from the hand-crafted behaviour are best structured, followed by the pure social interactions (following + social facilitation), the social interactions without social facilitation, and finally the random behaviour. In fact, in the simulation we see that with social facilitation the data are almost as well structured as with the hand-crafted behaviour, and this is because these two

[5]This tracking system, as well as much of the low-level control of the robot, was developed by Tetsushi Oka.

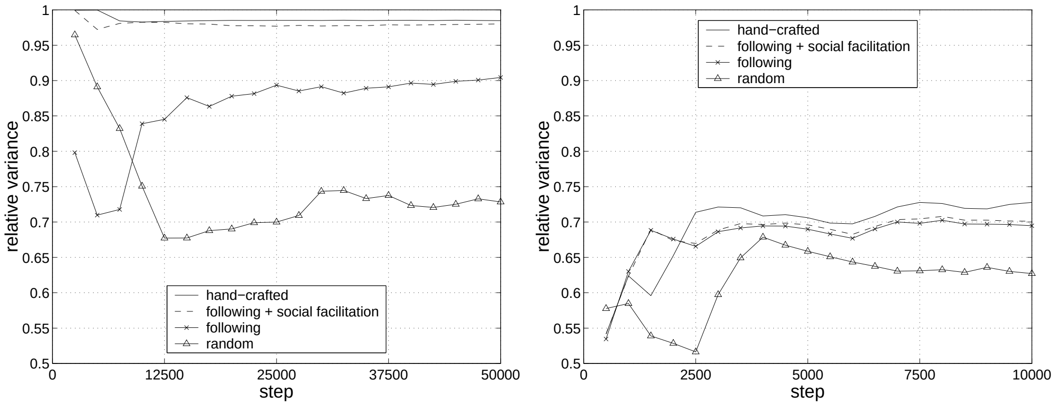


Figure 3.10: Convergence of the relative variance of the first two principal components from the simulation (left), and physical experiments (right).

scenarios are exposed to almost identical perceptions. We also see that social facilitation has no effect in the physical experiment and this is because of the reason mentioned in Section 3.1, namely that the demonstrator reacts immediately when not detected by Gillespie's tracking system. Because of the active demonstrations, the robot's perceptions are always consistent with respect to the task being demonstrated (it never gets the chance to lose the teacher), as opposed to the simulation experiments, where the robot is exposed to irrelevant perceptions when it loses the teacher.

An important difference between the simulated and physical experiments is the actual relative variance accounted for. Whereas in the simulation two dimensions can explain the data almost perfectly in the best scenarios, in the physical experiment they can only explain approximately 73% at best. Therefore in two dimensions the data from the physical experiment are not as well structured as in the simulation, even though they are modelling the same task. This is not surprising as Gillespie has many more sensors (20 compared with six), and the noise in them is not simulated! One would thus expect an attention system to deal with these data sources differently. The final values of the relative variance should be kept in mind when interpreting the projections of the complete datasets onto the two PCs.

Figure 3.11 shows the projection of each of the final datasets, *i.e.* at the completion of the runs, from the simulation experiments. We have already seen a suggestion of what a two dimensional projection from a wall-following task looks like, in the simple example shown in Figure 3.4. The projections in Figure 3.11 indeed look similar: one cluster is the intersection of two apparent 'lines', which is the area of low (or weak) wall-detection, *i.e.* 'no wall', and

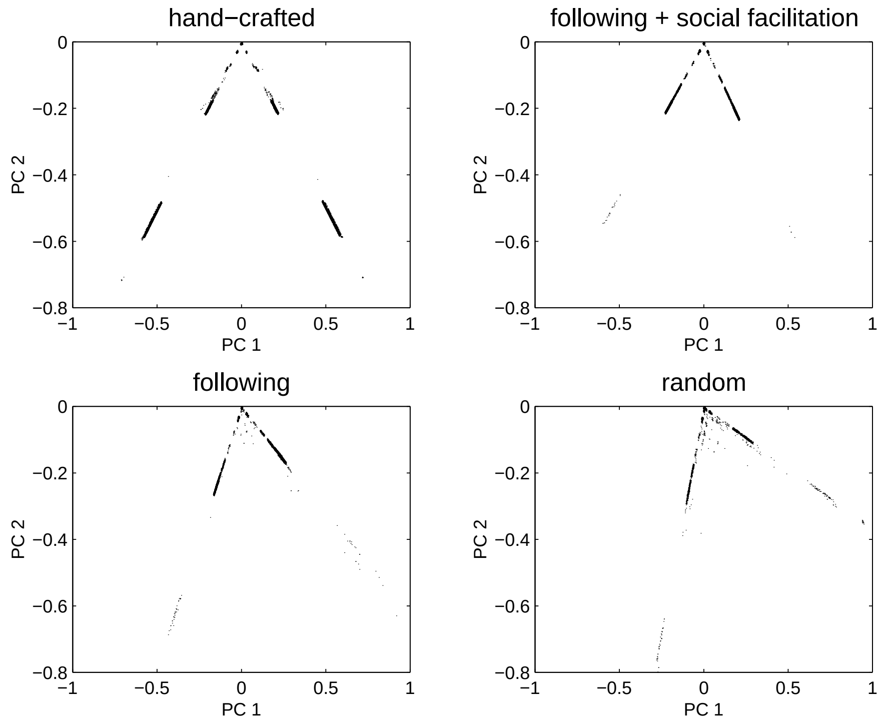


Figure 3.11: Projection of the data from simulation onto the first two PCs for the four scenarios.

as we move away from this intersection in either direction, we see data corresponding to 'right wall' and 'left wall', at different distances from the wall, clustered in distinct areas.

The hand-crafted and social facilitation scenarios look almost identical, as suggested by Figure 3.10, except that the former is exposed more to extreme values, which means the robot spends more time closer to the walls[6]. The 'following' scenario also looks very similar, but we start to see data points around the 'intersection' that are not on either of the two 'lines', corresponding to perceptions other than wall-detection on the side. This is even more apparent in the dataset from the random behaviour. In fact for the latter dataset we now recall that its two dimensional projection only accounts for approximately 73% of the data, and it is perhaps worth inspecting this dataset in three dimensions.

In Figure 3.12 we compare the three-dimensional projection of the random scenario with that of the hand-crafted scenario (the relative variances of these plots are 0.86 and 0.99 respec-

[6]This happens because sometimes the hand-crafted behaviour needs to do some zig-zags to maintain the distance from the wall; the same hand-crafted behaviour controls the demonstrator in the 'following' scenarios, but recall that the learner does not follow exactly the same path as the demonstrator — it actually cuts corners when following behind the demonstrator, and so it never gets as close to the wall.

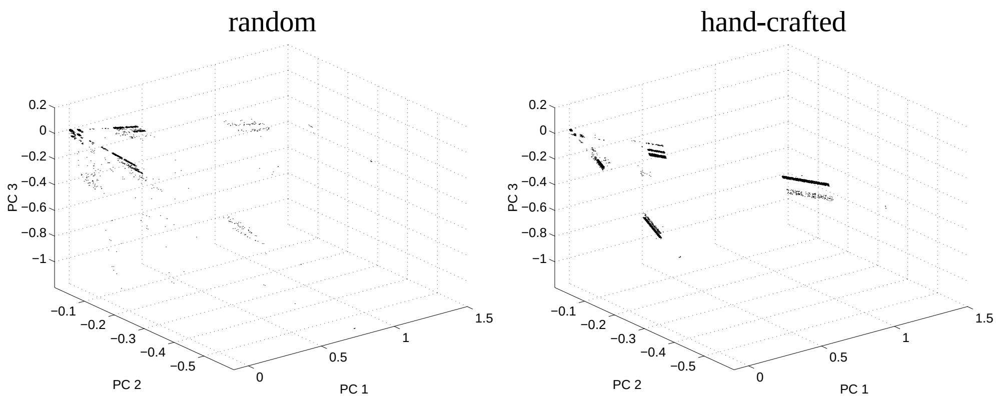


Figure 3.12: Projection of the data from simulation onto the first three PCs for the random and hand-crafted scenarios.

tively). We see that indeed there seems to be another ‘component’ in the data from the random scenario (going down along the third PC from the intersection) that is not related to the wall-following task, which is of course not surprising. A learning system would have great difficulties learning the wall-following task from such a dataset without further assistance, because the data contain too many irrelevant points (in fact, the data contain an additional ‘component’ which is not related to wall-following).

A numerical exploration of the PCs, as performed for the toy example, can provide a confirmation of the above qualitative interpretations of the projections of the data (Figure 3.11 and 3.12), as well as a a better understanding of the perceptual space. Figure 3.13 shows the PCs calculated by PCA for the complete dataset of the hand-crafted scenario; the PCs of all four scenarios are given in Figure A.2. As described for the simple example in the previous section, the six values of each PC correspond to the contributions of each of the six sensors to these calculations (see Figure 3.9 for the locations of the sensors on the robot).

The first PC corresponds to the contrast between left and right, because the values for the left sensors (0–1) have opposite signs to the values for the right sensors (4–5), and they are almost identical in absolute value; sensors 2–3 have no contributions — the robot hardly senses the wall in front; this PC accounts for 79% of the variance. The second PC corresponds to the total signal in the sensors: the values for left and right have the same sign, and again, are almost identical in absolute value. The first two PCs together account for 98% of the variance (see Figure 3.10); the origin (0, 0) in PC-space is the ‘no-wall’ state (see top-left plot in Figure 3.11), the second PC determines the strength of the stimulus, and the first PC

	PC 1	PC 2	PC 3
0:	0.700584	-0.702847	0.122952
1:	0.114031	-0.051821	-0.923956
2:	0.000148	0.000931	-0.027873
3:	0.000004	0.000620	-0.010723
4:	-0.112107	-0.059763	0.351001
5:	-0.695421	-0.706928	-0.084231

Figure 3.13: Three of the principal components calculated for the hand-crafted scenario of the simulation experiment. The components of the PCs correspond to the contributions of the six sensors (see Figure 3.9 for the locations of the sensors on the robot).

determines on which side it is sensed; the third (and any other) PC corresponds to the noise in the sensors.

If we look closely at the projection of the data for the hand-crafted scenario in Figure 3.11, we see that not only are there two straight lines, but also that they meet at the origin and have a slope of 1 or -1!

Let us consider individual points on this projection. First, consider a *single* sensory input, given by the row vector

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} \tag{3.6}$$

where the x_i's correspond to the six sensor values. As described in Section 3.2.1, the projection of $\mathbf{x}$ is a vector, $\mathbf{y}$, given by

$$\begin{aligned}
\mathbf{y} &= \mathbf{x}\begin{bmatrix} PC_1 & PC_2 \end{bmatrix} \\
&= \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}\begin{bmatrix} 0.7 & -0.7 \\ 0.11 & -0.05 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ -0.11 & -0.06 \\ -0.7 & -0.7 \end{bmatrix} \\
&= \begin{bmatrix} 0.7x_0 + 0.11x_1 + 0.00x_2 + 0.00x_3 - 0.11x_4 - 0.7x_5 \\ -0.7x_0 - 0.05x_1 + 0.00x_2 + 0.00x_3 - 0.06x_4 - 0.7x_5 \end{bmatrix}^T \\
&= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}^T
\end{aligned} \tag{3.7}$$

where the values shown in Figure 3.13 have been substituted for PC_1 and PC_2, and T corresponds to matrix transposition. Therefore, *any* projected point that lies on the line with slope 1 satisfies

$$\begin{aligned} y_1 &= y_2 \\ 0.7x_0 + 0.11x_1 - 0.11x_4 - 0.7x_5 &= -0.7x_0 - 0.05x_1 - 0.06x_4 - 0.7x_5 \\ 1.4x_0 + 0.16x_1 - 0.05x_4 &= 0 \end{aligned} \tag{3.8}$$

and any projected point that lies on the line with slope -1 satisfies

$$\begin{aligned} y_1 &= -y_2 \\ 0.7x_0 + 0.11x_1 - 0.11x_4 - 0.7x_5 &= 0.7x_0 + 0.05x_1 + 0.06x_4 + 0.7x_5 \\ 1.4x_5 + 0.17x_4 - 0.06x_1 &= 0 \end{aligned} \tag{3.9}$$

We see from Equation 3.8 that the projection of the points that lie on the line with slope 1, is only influenced by the first two bits of the input, *i.e.* sensors 0 and 1 (plus a slight negative influence from sensor 4). In other words, what characterises the points that are projected on this line is that they correspond to a strong signal from the left side of the robot, with most of it immediately on the side (sensor 0). Similarly, from Equation 3.9 we see that points that lie on the line with slope -1 correspond to a strong signal from the right side of the robot (sensors 4 and 5). Notice the symmetry between left and right: the hand-crafted behaviour exposes the robot equally to both sides.

The PCs can be similarly inspected for the other scenarios (all shown in Figure A.2). They all share similar properties with the hand-crafted scenario, for example they all have an ‘origin’ at (0,0); however notice that the PCs are not as nicely symmetrical, which explains why the corresponding emergent lines in Figure 3.11 have slopes slightly different than 1. Of course, the main difference between the scenarios is the relative variance, as seen in Figure 3.10. For example, in the random scenario, although the first two PCs have a similar structure, here the third PC should be inspected as well, because it accounts for a significant amount of the variance. In fact, the third PC (see Figure A.2(d)) seems to be symmetrical about the center, suggesting that it represents a contrast between sensing the wall in front and on the sides. This third component, also seen on the left of Figure 3.12, is most significant for the random scenario because the robot senses the wall in front very frequently, as opposed to the wall-following scenarios.

Figure 3.14 shows the projection of each of the final datasets from the physical experiments onto two PCs. The structure in the projections of the ‘hand-crafted’ and ‘random’ scenarios

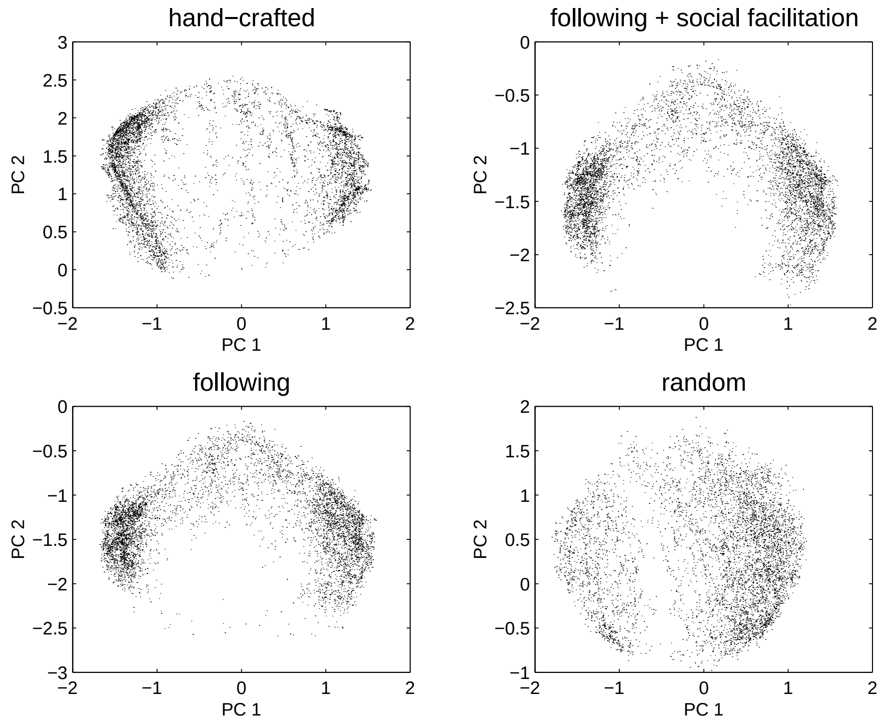


Figure 3.14: Projection of the data from the physical experiment onto the first two PCs for the four scenarios.

seems to be different from the two ‘following’ scenarios. The structure in the hand-crafted and random scenarios is not as observed so far for the wall-following task, namely two straight lines intersecting at the origin in PC-space, whereas the structure in the ‘following’ scenarios *does* match this description. There are two reasons for this: (1) in the hand-crafted and random scenarios the robot perceives the wall straight in front much more frequently than in the ‘following’ scenarios where the robot is always guided to turn towards the wall by the demonstrator; (2) the demonstrator ‘exaggerates’ the time spent in the middle of the arena in order to expose the robot to the ‘no wall’ state, whereas in the hand-crafted case the wandering behaviour controls the robot, so the time spent in the middle of arena is less ‘controlled’: the robot sometimes spends a very short time there before finding a wall again, and consequently there are fewer data points.

The second result mentioned above is evidence that the active demonstrations by the teacher are useful not just to expose the robot to relevant perceptions, but also to help the robot distinguish between the components of the task, by ensuring that the robot is exposed equally to these different components. We will see in the next section the implications this has for an attention system that is attempting to self-organise the robot's perceptions, and in Chapter 4 we will see the implications for learning the task.

The projections in Figure 3.14 can be better explained, and the differences mentioned above between the scenarios can be confirmed, by inspecting what the principal components actually are. The PCs for all four scenarios are shown in Figures A.3 – A.6. The first thing to note is that in all scenarios the most significant component (*i.e.* the first PC) corresponds to the contrast between left and right, as described in the simulation experiment. However in the random scenario this PC only accounts for 39% of the variance, as opposed to approximately 60% in the others. The difference between the hand-crafted and random scenarios, and the two 'following' scenarios, is seen in the second PC, *i.e.* the second most significant component. For the 'following' scenarios, this component is as in the simulation, *i.e.* it corresponds to the total signal — how strongly the wall is sensed; for the other two scenarios this component seems to correspond to a contrast between front and sides, *i.e.* it distinguishes between sensing the wall in front and on *any* side.

As already mentioned, the relative variances accounted for by two PCs are not as high for the physical experiment as for the simulation (see Figure 3.10). A third PC does not improve the situation much — 77% of the variance is accounted for, at best. This suggests that the only useful information is in the first two PCs. However they only account for 73% of the variance at best, so the rest must be due to noise in the sensors — a lot more noise than in the simulation experiment. This is related to the discussion in Section 3.2.1 (see A Simple Example) about the complexity of the task: the physical experiment involves the same task as the simulated experiment, namely the wall-following task, so two dimensions should be able to capture the three main clusters in the data. However, because of the noise in the physical sensors, the data in the physical experiment are less well-structured in these dimensions; adding more dimensions does not improve this significantly.

The principal components calculated in the remaining examples in this section are all shown in Appendix A; however the interpretation of their actual values is omitted and left for the interested reader.

Phototaxis

We will now look at the perceptual data arising from a phototaxis task in the Khepera simulation environment, in which three light sources have been added, as shown in Figure 3.15. Perception of the environment consists of ambient light coming from the light sources. The task involves

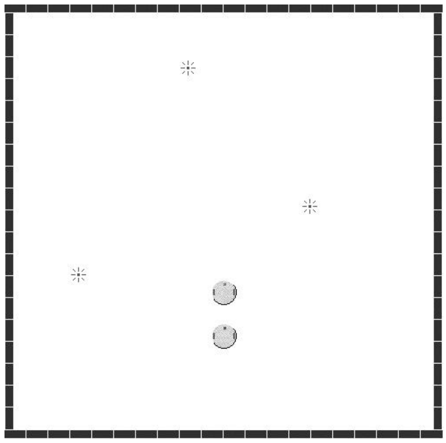


Figure 3.15: The simulated environment consists of a teacher robot followed by a learner robot; the environment used for the phototaxis task contains three light sources.

turning towards and approaching the light sources, but not stopping at any point; the robots can actually pass ‘through’ (or ‘under’) these light sources, and because they do not perceive from the back, upon passing ‘through’ the light source the teacher will simply continue moving until it finds a new light source. If no light is detected the default behaviour is a random wandering behaviour with obstacle avoidance.

It turns out that in the phototaxis experiments the robot never loses the teacher[7], and so, as opposed to the wall-following experiments, only three testing scenarios are required here: ‘hand-crafted’, ‘following’, and ‘random’ — see section 3.1. Figure 3.16 shows the relative variance accounted for by two PCs for these scenarios. Recall that the teacher-following behaviour drives the learner to cut corners. This means that while the hand-crafted behaviour drives the robot to pass directly in front of a light-source, the imprecise teacher-following behaviour drives the learner, as it follows the teacher, to pass at slightly different angles towards the light-source each time. This explains why the perceptual data for the ‘following’ scenario are not as well structured as the ‘hand-crafted’ one. Note that in these experiments

[7]The reason that the learner loses the teacher in the wall-following experiment is related to an implementation difficulty in the Khepera simulator. When executing this task, the teacher sometimes ends up facing the learner, either because it faces a corner in the environment, which forces it to turn around, or the interrupt signal forces it to turn around. The distinction between robot-detection and object-detection is not very robust in the simulator, and therefore when the teacher faces the learner, the teacher will sometimes fail to ‘avoid’ the learner and instead go ‘through’ it; the learner does not detect from its back and therefore the result is that for the learner, the teacher suddenly disappears. In the phototaxis experiments the teacher never ends up facing the learner, firstly because there is no interrupt signal, and secondly because there is much less interaction with the walls in the environment, so it is extremely rare that the teacher faces the corners of the environment.

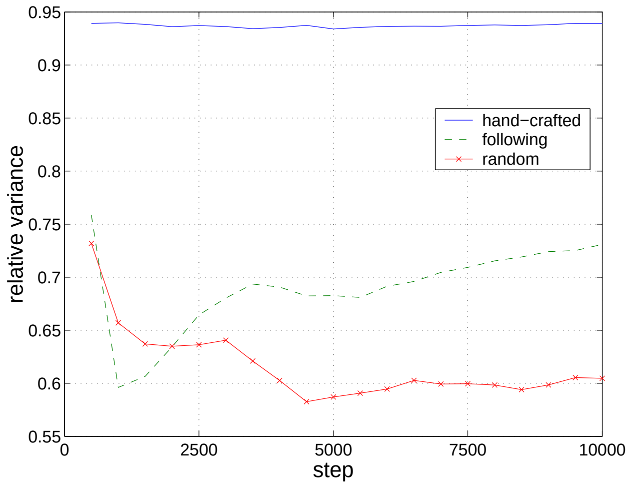


Figure 3.16: Convergence of the relative variance of the first two principal components from the phototaxis experiments.

the demonstrator is passive — it executes the task independently of the learner; a more active demonstrator might, for example, make wider turns to take into account the fact that the learner cuts corners, and the learner would thus be exposed to more correct perceptions, with respect to this task. As expected, the 'random' scenario is least well-structured.

The difference between the first two scenarios can also be seen in the projection of the data onto the first two PCs, shown in Figure 3.17. The first thing to note in all scenarios is that there is only one significant cluster: the 'no stimulation' cluster (*c.f.* 'no-wall' cluster in the wall-following experiments); the reason for this is that the task involves spending the majority of the time looking for light sources, and not spending a significant amount of time at the light sources themselves. In the 'hand-crafted' scenario one can almost observe another cluster (or at least a relatively dense cloud of points) at the right of the plot, corresponding to being directly in front of the light; in between we can see data corresponding to sensing the light at different distances. In fact, it seems that only the first PC is significant: how far the light is sensed straight in front; the second PC seems to be less significant. This is in fact not a desirable outcome: because PCA finds the direction in the input space of maximum variation, the first principal component captures the differences between light-detection and no light-detection, which corresponds to a rather rough level of granularity. A more desirable level of granularity would capture the finer distinctions of detecting the light at different angles, so that the robot could potentially learn how to turn towards the light from these different angles. Perhaps the

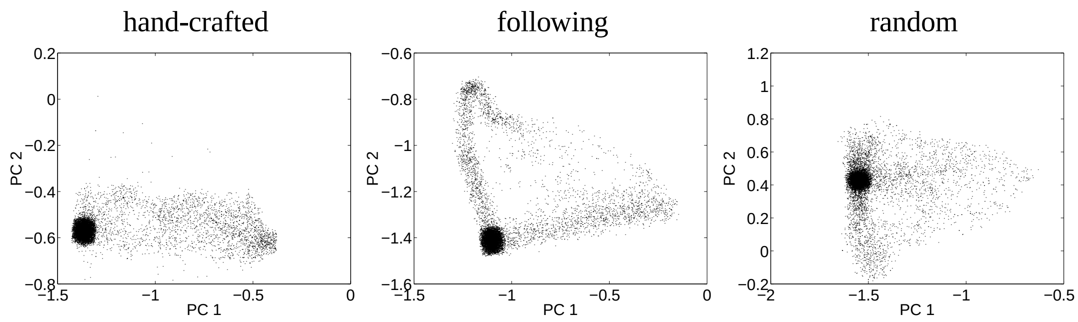


Figure 3.17: Projection of the data from the phototaxis experiments onto the first two PCs for the three scenarios.

best way to represent this task is at two different levels of granularity.

In the ‘following’ scenario we see that there is more complexity in the data: the light is perceived from the left and right at different distances (the two ‘lines’ meeting at the big cluster, as in the wall-following task — see Figure 3.11); the light is also perceived at the front, though this is difficult to see in this plot which only explains approximately 73% of the variance; a three-dimensional projection is therefore provided in Figure 3.18 (which accounts for 89% of the variance), where we can see the ‘approach’ to the light-source from the two sides. So here, a side-effect of the imprecise teacher-following behaviour (driving the learner to cut corners) means that there is more variability in the data, and the distinction of light-detection from different sides is now more significant than in the hand-crafted scenario, at this level of granularity as chosen by PCA.

In the ‘random’ scenario we see that the light is perceived at many different angles and distances from the light-source, as expected since the robot is wandering around the environment and meeting the light-sources at random.

Object-interactions

The experiments presented here[8] involve two 11 degrees of freedom simulated humanoid robots (waist upwards), a demonstrator and an imitator, interacting with an object, as shown in Figure 3.19. Each robot has three degrees of freedom at the neck, three at each shoulder, and one at each elbow; for each degree of freedom a velocity is also measured. The objects are identical and have six degrees of freedom (three for position and three for velocity). The dy-

[8]The object-interaction experiments form part of joint work with George Maistros (see Section 5.3), whose programs are used in producing the data for the object-interaction experiments.

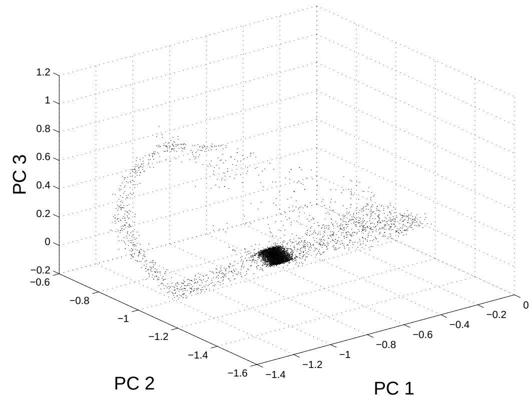


Figure 3.18: Projection of the data from the 'following' phototaxis scenario onto the first three PCs; the relative variance is 89%.

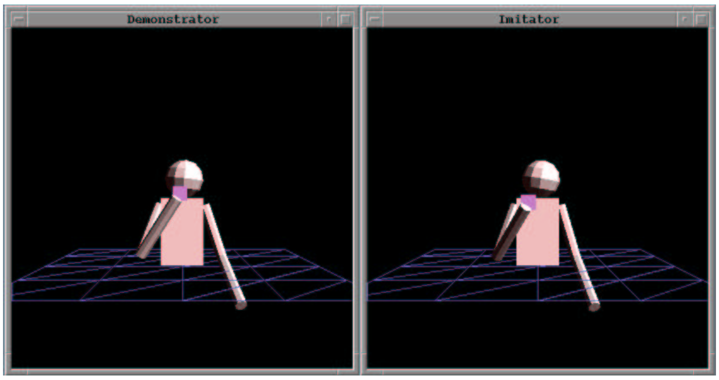


Figure 3.19: The object-interaction experiments consist of an imitator and a demonstrator.

namics of each robot are simulated in DynaMechs (McMillan et al., 1995), a collection of C++ libraries that simulate the physics involved with objects and joint control. The torque for the control of each joint, *i.e.* the input to DynaMechs, is calculated with the aid of a Proportional-Integral-Derivative (PID) controller[9], which converts postural targets (*i.e. via-points* for each joint) into such torque values. The demonstrator is controlled by a sequence of such postural targets to interact with its object, which is lying on a surface at waist level. The absence of fingers, as well as software limitations, lead to a rather crude robot-object interaction: the object is merely attached to (or detached from) the wrist, as long as this is desired and the wrist is close enough.

[9]Written by Yiannis Demiris.

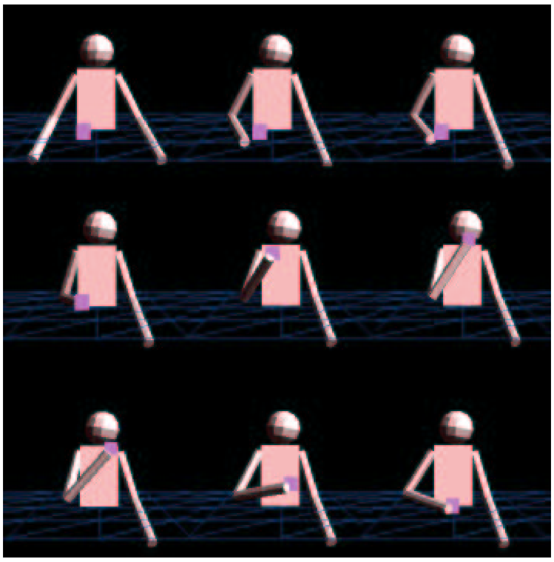


Figure 3.20: A sequence of snapshots of the first object-interaction; left to right, top to bottom.

In Section 5.3 we will see how well the learner can reproduce the actions of the demonstrator, whereas in this chapter we are only concerned with what the learner is perceiving. In the object-interaction experiments the observation of the demonstrated task is passive: the learner does not actually move itself. The data come from a crude approximation to visual perception which consists of the articulation joint angles of the observed demonstrator (11 degrees of freedom), their corresponding joint velocities (another 11), and the position and velocity information of the observed object (6 degrees of freedom) — 28 dimensions in total, where noise is also added to each.

Note that in this type of experiment a 'hand-crafted' scenario as used in the wall-following and phototaxis experiments corresponds to zero noise in the perception, and similarly a 'random' scenario corresponds to significantly high noise. Although analysing such scenarios is interesting, instead we will exploit the complexity available with this humanoid platform to compare sensorimotor data from four different object-interaction tasks, with varying degrees of similarities to each other:

1. The robot approaches the object (glass of beer) with the right hand, 'grasps' it, moves it towards the mouth, 'drinks' its hypothetical contents, and then 'puts' it back on the surface, as shown in Figure 3.20.

2. This interaction is an 'impolite' version of action 1; it differs in that the elbow is pushed further out while the object is moved to the mouth.

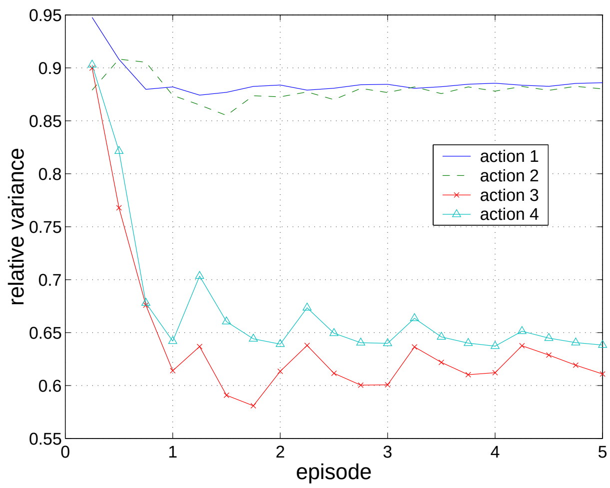


Figure 3.21: Convergence of the relative variance of the first two principal components from the object-interaction experiments.

3. The object is picked up with both hands, moved along the surface to the side, put down, and both hands move to their original positions.

4. The object is picked up with the right hand, transferred to the left hand, and put down; the right hand moves to its original position.

Each of these interactions is repeated five times. We call each repetition an ‘episode’, and as before we collect all the perceptions into one dataset per task. The convergence of the relative variance accounted for by two principal components is shown in Figure 3.21 for all interactions. Note that the length of an episode for each of the four interactions is different (ranging from 2600 to 4000 steps), so the datasets are of different sizes; in Figure 3.21 the lengths have been standardised so that all curves fit on one plot.

Firstly, we see that for all interactions PCA converges after 1–2 episodes. Note that episodes are not identical due to the accumulation of the error between the output of the PID controller and the via (target) points. Nevertheless, 1–2 presentations of the interactions seem to be sufficient for obtaining the structure of the perceptual data.

Secondly we see that actions 1 and 2 are better structured in two dimensions than actions 3 and 4. This is not surprising as the latter two use more degrees of freedom: the first two actions only involve the right hand, the left hand is stationary and therefore does not need to be represented. So the task complexity in actions 3 and 4 is higher, and therefore the salient

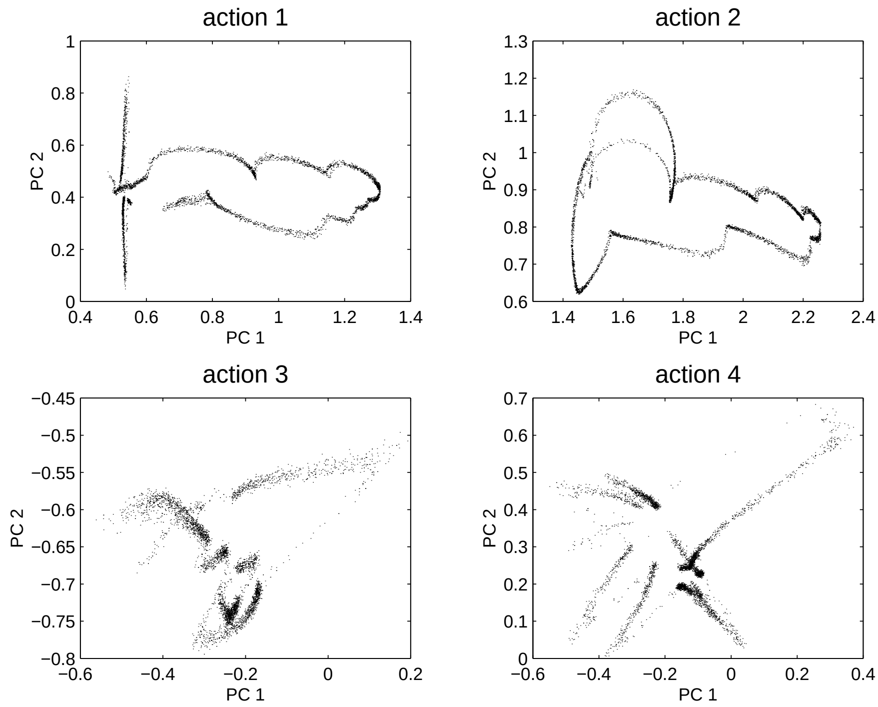


Figure 3.22: Projection of the data from the object-interaction experiment onto the first two PCs.

variations must occur at a finer level of granularity — when three PCs are used, their relative variance increases significantly to 74% and 78%, respectively. Further, the curves for actions 1 and 2 in Figure 3.21 are very similar which suggests that the difference between the first two interactions can be explained by a two dimensional projection, which we will see below.

Figure 3.22 shows the projections of the final datasets, *i.e.* at the completion of all episodes, onto the first two PCs. As before, interpreting these projections is rather difficult. Action 1 has been examined extensively in experiments reported in this chapter (see more in Section 3.3.4) and Section 5.3, which leads to a fairly confident conjecture about how to interpret the corresponding PCA plot. The interpretations of the other actions can only be made relative to that of action 1, but this is not so straightforward for actions 3 and 4. Note that our aim here is only to demonstrate that our method for visualising perceptual data can distinguish between different scenarios and different complexities; Figure 3.22 certainly demonstrates this.

We can distinguish three parts to action 1 in Figure 3.22: one fairly straight curve on the top left side of the figure (going top to mid-height), a loop starting at the mid-height left (going clockwise), and another fairly straight curve at the bottom left (going mid-height to bottom).

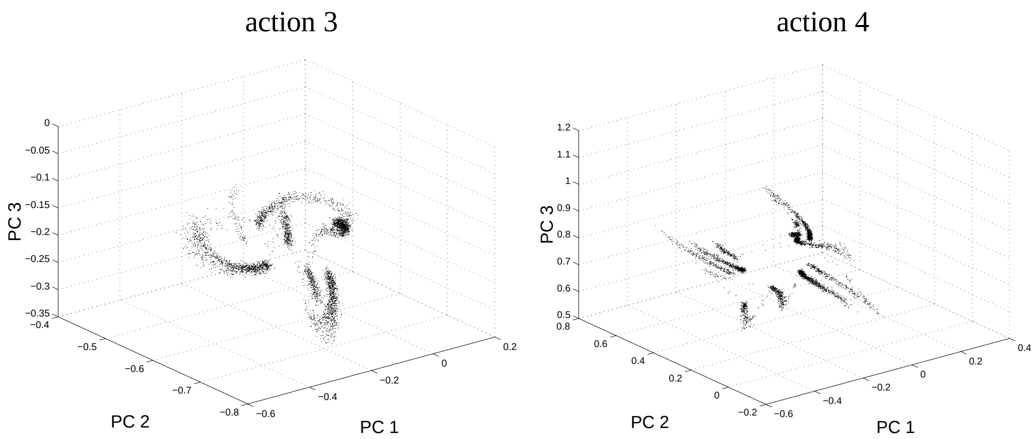


Figure 3.23: Projection of the data from object-interactions 3 and 4 onto the first three PCs; the relative variances are 74% and 78%, respectively.

These respectively correspond to the three major parts of the action: (i) approach the glass, (ii) pick up, drink, and put down (iii) move away from the glass. Note that this plot is *not* of the actual trajectory of the hand, but rather a compressed representation of the joint-angles, joint-velocities (hence parts (i) and (iii) above are separated), and object position and velocity. However, the complexity of the action is low enough that its spatial features can be captured by this two dimensional compressed representation. This representation is also sufficient for expressing the difference between actions 1 and 2: in the latter the elbow is further away from the body, which is clearly visible in the top-left part of the plot for action 2. So although these two actions are different, their salient variations occur at the same level of granularity, at which these actions can therefore be distinguished.

In contrast, since a two-dimensional representation does not seem to be suitable for actions 3 and 4, as suggested by Figure 3.21, these actions might not be distinguishable at this level of granularity. The first *three* PCs account for a significantly higher proportion of the variance — 74% and 78%, respectively — and so the three-dimensional projection of their data, shown in Figure 3.23, might be more suitable for distinguishing between the two actions. However, these actions are not considered further in this thesis, and so their interpretation is left to the interested reader.

Note that, compared to the wall-following experiments, the data in the object-interaction experiments do not contain distinct clusters — the data seem to be equally distributed in the space. This is because these tasks are sequential in nature — the teacher demonstrates continuous movements. In fact, the demonstrations are *almost* continuous — the demonstrator actually pauses briefly just after grasping the glass, half-way between the table and the mouth, and at the mouth; and if one observes closely one can in fact see more data points at these locations (the clusters in Figure 3.22 for actions 1 and 2 are slightly 'darker'). However, these exposures are not significant to show clearly on the plots, and similarly, they might not appear salient enough for the learner, as we will see in the next section. A more active demonstrator would purposely slow down the demonstration at these particular times, in order to provide the learner with plenty of examples of the important parts of the tasks and hence accentuate the salient differences between them.

3.2.3 Conclusion

Many clustering and dimensionality reduction techniques like PCA exist and have been used successfully to model robotic sensory input, and we will see another one in the next section where a short literature review will also be provided. However, PCA as used here also provides a useful tool for inspecting perceptual data. This seems to be lacking in other robotic perception work, where researchers rarely show what the robot is actually perceiving, because of the high dimensionality in the data. Therefore as well as being a useful development tool, this approach to inspecting perceptual data seems to be novel in this context, and could prove valuable in other work involving robotic perception. For example a designer might want to know how long it takes, if at all, for his/her reinforcement learning robot to get exposure to all the environmental conditions and hence rewards necessary to learn a task. This tool can be applied to data from any modality, including motor and proprioceptive data, and is therefore a general tool for inspecting sensorimotor data.

Different activities were suggested in this section that can be used by a designer to learn about the nature of the robot's raw data. These are activities that can be used prior to the design of a learning architecture — they simply involve data inspection[10]. PCA is a widely available off-the-shelf tool that can be used to perform the various activities mentioned in this section, in an increasing 'analysis effort' (as opposed to design effort), as follows. The most straightforward operational use of PCA is to provide it with a dataset, and specify the number of PCs to use. PCA will then give the relative variance accounted for, which by itself is very informative, especially for comparing datasets obtained under different conditions. Then one can also project the data using the PCs onto a space that is more easy to visualise and interpret.

[10] Of course, the nature of these 'data' is pre-imposed by the designer in choosing particular sensors and actuators.

A more analytical inspection involves inspecting the PCs themselves and through them working out how the sensors are correlated and how they contribute to the perception of the task.

In our particular case, the benefit of visualising and analysing the perceptual data is that it gives us an idea of the structure in the data — a valuable empirical investigation of the following:

Level of granularity. PCA gives an indication of the complexity of the task, by reducing the full dimensionality of the data to a compressed representation that is related to the complexity of the task. In effect, this also corresponds to the level of granularity at which significant variations exist in the data; note that PCA always seeks the maximal variations, which is not necessarily desirable; for example, we saw in the phototaxis example that it might be desirable to combine different levels of granularity. We also saw how one task, namely the wall-following task, has the same complexity and hence the same level of granularity, when performed both in simulation and in a physical setting; these experiments provide a very important demonstration that even though the task complexity is the same, the data are much less well-structured when a physical robot is involved. The object-interaction experiments demonstrated that tasks of different complexities, performed on the same platform and with the same robot, can lead to different levels of granularity.

Imprecise teacher-following behaviour. We compared the data that the robot is exposed to through the teacher-following behaviour with an approximation to an ideal teacher-following behaviour (*i.e.* the hand-crafted scenario), and saw the differences in the robot's perceptions. These differences were more significant for the phototaxis task because the fact that the learner cuts corners when following the teacher is more of a problem for this task than the wall-following task. Of course, if the demonstrator is more active, he can influence the learner to take a more desirable path, and even accentuate the variations in the task by forcing high exposure to the parts of the task that differ, as we saw in the physical wall-following experiments. Active demonstrations also minimise the chances that the robot gets lost, and we saw in the simulation wall-following experiments that when the robot is lost, its overall exposure begins to resemble exposure to random experiences, which might be a problem for learning.

Implications for Saliency and Attention. The inspections of the data provide us with an idea of how well an attention system would be able to deal with the different perceptual data sources. The shortcomings mentioned with regards to how well-structured a particular dataset is in a particular dimension, can be directly translated to the ability of an automated system to abstract from the data. One of the roles of attention is to find the structure through self-organisation — to map perceptual data to discrete structures; as argued in Chapter 2, this requires a measure of saliency. The data inspections performed here give an indication of what might constitute saliency, by exposing the level of granularity at which significant variations occur. We have also seen some indication of how the demonstrator can help accentuate the salient variations in the task, through more active demonstrations. Further, the investigations performed here, especially the visualisation of the data, can be used to evaluate the output from the attention system, presented in the next section, by comparing it with the actual data being modelled.

As mentioned at the start of this section, the motivation for carrying out such an investigation is that it provides a designer with a good starting point for designing a learning architecture, and gives an indication of the conditions necessary for learning. In our particular case, the investigation confirms that the imprecise exposure of the learner due to the imprecise teacher-following behaviour might indeed be a problem for learning — although this may be intuitively obvious, it is nevertheless important to demonstrate empirically; this has led to the utilisation of the active demonstrations in the physical experiments. The investigation also suggests that learning with a physical robot might be difficult due to poorly-structured data, and this has led to introducing a further increase in the complexity of the social interactions, in the form of explicit signalling by the demonstrator (see Chapter 4). Finally, the investigation suggests what would be involved in learning from the data in the different tasks; the wall-following tasks contain distinct clusters, whereas the important dynamics in the phototaxis task seem to occur at a rather fine level, and the object-interaction tasks seem to be more sequential in nature.

This investigation only provides a starting point. In the context of the aim of this thesis, the designer still needs to design parameterised saliency mechanisms, identify the important parameters, impose a bias reliably by setting the values of these parameters, and consider how social interactions can be useful in transferring the above effort to the expert involved in the interaction. This section has started to address these issues, and the next section will address them more explicitly by presenting an attention system.

3.3 An Attention System: Abstraction Through Self-Organisation

Chapter 2 promoted a qualitative and quantitative benefit of attention. The qualitative benefit is related to selecting experiences based on the robot's needs, which in this thesis corresponds to learning a task. Section 2.2.1 presented bottom-up and top-down temporal selection mechanisms, and it was argued that top-down mechanisms require structure to be forced by the designer. The interest in this thesis, however, is to examine situations where little effort is needed from the designer. Therefore, bottom-up mechanisms, which find a structure in the raw data and then base selection on this structure, are more suitable. The quantitative benefit of attention is related to dealing with perceptual overload, which in this thesis corresponds to controlling the number of experiences that a learning architecture is exposed to, that is, it corresponds to modulating learning.

A short review of unsupervised learning and self-organising approaches is provided below in Section 3.3.1, leading to the choices for an attention system. Section 3.3.2 presents the algorithm used by the attention system, followed by the identification of the important parameters in Section 3.3.3. Section 3.3.4 shows how the algorithm is implemented on the robotic tasks presented in Section 3.2.2.

3.3.1 Background

Several unsupervised learning approaches are suitable for implementing the kind of mechanisms promoted above. As suggested by Hertz et al. (1991), these approaches can generally be broken down into two categories: competitive and non-competitive learning. Non-competitive unsupervised learning is *not* suitable for clustering or classification, *i.e.* producing a mapping from data to group-membership, but is useful for measuring familiarity and projecting data onto some principal components. Hertz et al. (1991) mention Hebbian-based learning approaches, of which PCA is an example, for measuring the similarity of an input to typical or distributional data seen in the past. Indeed, we have seen in the previous section that while PCA is good for reducing the dimensionality of a dataset (and therefore perhaps measuring familiarity on a lower dimension), it does not actually provide a mapping from data to clusters. In fact, PCA is often applied to data *prior* to the application of a clustering mechanism.

Clustering[11]

Competitive unsupervised learning generally involves a neural network consisting of a set of output units competing to match an input, using some distance measure. The simplest approaches involve updating the 'winning' unit to match the input even better, by moving the unit towards the input by a fraction, called the learning rate, of the distance between them. These approaches use a typical learning rule of the form:

$$\Delta \mathbf{w}_{i^*} = \eta(\mathbf{w}_{i^*} - \xi) \tag{3.10}$$

[11] This part of the review is based on Chapters 8 and 9 of Hertz et al. (1991).

where $\mathbf{w_i}$ is the set of weights used to compute the value of unit i from the input (and hence has the same number of dimensions as the input) , i^* is the winning output unit, ξ is the current input, and η is the learning rate ($0 \leq \eta \leq 1$). The result is that the units position themselves in the input space to represent it as best they can. The problem with these simple learning approaches is that they are not guaranteed to converge, because units can move endlessly in the input space. This can be overcome by reducing the learning rate over time, however that compromises the plasticity of the network, and impairs its ability to adapt to a non-stationary environment. This dilemma is well known and often referred to as the *stability-plasticity dilemma*.

Adaptive Resonance Theory (ART) of Carpenter and Grossberg (1988) addresses this problem by only committing a unit if it is *sufficiently* similar to the input; if it is not, a previously unused unit is recruited for the input, which is only possible while there are unused units still available. Therefore, stability is preferred over plasticity once capacity has been reached. Of course, the measure of similarity is a parameter that is predetermined, and can be used to control the level of granularity in the resulting representation; capacity, *i.e.* the number of units, is also a pre-defined parameter.

Another shortcoming of the simple unsupervised learning approaches is that they do not represent the geometrical topology present in the space. Topology refers to neighbourhood relations, *i.e.* the similarity or dissimilarity between regions in the input space. Topology-preserving approaches provide a formalisation of these relations through the connectivity between output units. One of the most popular of such approaches is Kohonen's *Self Organising Feature Map* (SOFM or SOM) (Kohonen, 1982, 1984). In this approach the neighbours of the winning unit are updated together with the winning unit, where neighbours are determined through a neighbourhood function. The units, often referred to as *nodes*, are fully connected to form an array, or map (usually one- or two-dimensional), which is designed *a priori* together with the neighbourhood function (although the neighbourhood function can vary with time). Nodes representing similar inputs move to similar locations in the input space, while nodes representing dissimilar inputs move to separate locations; therefore the features and topology of the input space are represented by the clustering of nodes, and the connections between them, respectively.

Growing Networks

The approaches described so far, and indeed the majority of supervised and unsupervised learning approaches, rely on the existence of an architecture prior to any learning. The architecture generally specifies how many units (input, output, or hidden) are used, and the relationships between them, *i.e.* their connectivities.

Marsland (2001) reviews a sample of extensions to these approaches, where the learning architecture is modified during learning, and the neural network can therefore be thought of as ‘growing’. He argues that the existing approaches are not always suitable for robotic applications because they are either not topology preserving, such as the ART network described above, or new structures are added periodically rather than in response to incoming data, such as the Growing Neural Gas (GNG) network (Fritzke, 1995) which will be described below. As part of his PhD work on on-line novelty detection for inspection robotics, Marsland developed a new algorithm based on the GNG, called the Grow When Required (GWR) algorithm, which he claims “responds quickly to data, while maintaining a topologically correct set of neighbourhood connections in the map space” (Marsland, 2001, p. 123).

One of the first of such extensions to Kohonen’s SOFM was Fritzke’s Growing Cell Structures and Growing Neural Gas algorithms (Fritzke, 1995). In his approach, the number of nodes is *not* specified *a priori*, and also a neighbourhood function is not specified; instead, nodes and edges are added and deleted during learning. The decision to add nodes is based on the error accumulated by the network in representing the data, which is computed separately for each node. As a node moves in the input space in response to an input (as described above), it measures how much it mismatches the input, and incorporates the mismatch into an accumulated error measure. At regular intervals, a new node is inserted close to the node with the highest error, with the hope that both that node and the new node will better represent that part of the input space. The treatment of edges is based on a technique called Competitive Hebbian Learning: the two nodes that best represent the input (*i.e.* the winning node, and the ‘second-best’ node), are connected with an edge; the rest of the edges connected to the winning node are weakened, and when weak enough are deleted.

Marsland’s criticism of the GNG is firstly that the insertion of nodes is not driven by the immediate input, but rather by misrepresentation *anywhere* in the network; further, because decisions are made at regular intervals, the insertion of a new node does not necessarily occur when the novelty occurs, but possibly some time later. The second criticism is that the actual location of the new node does not reflect the input (it is inserted between the best and second-best nodes), which means that other nodes have to move around to cover the novel location of the input space. In other words, the algorithm does not contain explicit parameters that directly control the selection, or saliency, of the *current* experience. Saliency is modelled indirectly and implicitly through the frequency with which nodes are added. Similarly, the other approaches presented above also model saliency indirectly, through the pre-defined number of nodes (although the ART system by Carpenter and Grossberg (1988) has also an explicit measure of saliency — the similarity measure).

In contrast, Marsland's GWR algorithm contains parameters that model saliency explicitly and directly in terms of the current experiences. This can be useful for designing mechanisms with which the parameters are autonomously tuned, perhaps through their communication to an expert during social interactions, and can therefore be adapted if the measure of saliency is dynamic, as suggested in Section 2.1.2 and will be discussed further in Chapter 6. Further, his algorithm models habituation, which can be used to inhibit a response to familiar experiences, and hence address the quantitative purpose of attention. A number of other examples of related work modelling habituation were presented in Section 2.2.1. However, the incorporation of the complete set of properties desirable for an attention system, as discussed here, makes the GWR algorithm an ideal computational model. It has therefore been adopted and suited to the purposes of the work reported in this thesis. All the details of the algorithm are given below, but for any other information the reader is referred to (Marsland, 2001), where an extensive comparative study between GWR and other algorithms (such as GNG) is also provided.

Towards a Model

The computational system that will be described next consists of a self-organising network, whose nodes and edges grow from experience, using the GWR algorithm which is based in part on the GNG algorithm. More specifically,

- the system involves competitive unsupervised learning, where one node wins and is updated to better represent the input, as are its neighbours to a lesser extent;
- the learning rate is fixed, *i.e.* plasticity is always preferred at the expense of not achieving complete stability; however due to the growing nature of the system, it is hoped that nodes are created and moved to appropriate regions of the input space where they at least remain in those regions, though not necessarily stationary;
- edges are manipulated through competitive Hebbian learning;
- habituation is modelled through an individual activation-frequency measure for each node.

The first three points have already been discussed in this section; the last one regarding habituation will be discussed below in the description of the algorithm. The shorthand 'SOFM' is used throughout the thesis to refer to this growing network, even though strictly speaking the implementation does not correspond to the original Kohonen SOFM algorithm.

3.3.2 The Algorithm

As mentioned above, the algorithm is largely adapted from Marsland's GWR algorithm. The main details will now be presented, starting with the implementation of habituation, and followed by a detailed algorithmic description of the GWR approach.

Habituation

The main asset of the algorithm is that it keeps a *habituation* measure for each node in the SOFM. This measure gives an indication of familiarity, *i.e.* the frequency of that node's activation, which provides a useful heuristic. Each time a node is active, its habituation value decreases exponentially, as shown in Figure 3.24, according to:

$$\tau \frac{dy(t)}{dt} = \alpha[y_0 - y(t)] - 1 \tag{3.11}$$

where y is the current habituation value, y_0 is the initial habituation value, τ determines the habituation rate, and α determines the habituation asymptote. In theory, one can incorporate the strength of the stimulus into this equation such that stimuli of different strengths have different effects on the habituation. This can be done by replacing the constant term (-1) in Equation 3.11 with a function of the stimulus strength over time; further, if one sets the stimulus strength to 0, this has the effect of increasing the habituation value slightly, and over a short period of time introduces 'forgetting'. Marsland (2001) uses this type of forgetting for nodes that are not associated with the current input; in the implementation here we will use a constant stimulus strength term as shown in Equation 3.11 and model forgetting in a different way.

For the algorithm presented below, two thresholds are required: a 'minimal' habituation threshold (the top dashed line in Figure 3.24), used to determine if a stimulus is completely unfamiliar to a node, and a 'full' habituation threshold (the bottom dashed line in Figure 3.24), used to determined if a stimulus is very familiar.

In order to eliminate the need to tune many parameters, we set up a typical scenario, as depicted in Figure 3.24. Habituation starts at 1 ($y_0 = 1$), always converges on the same asymptote (fix α, in our case to 1.05), and the habituation thresholds are fixed (to 0.7 and 0.1); the only

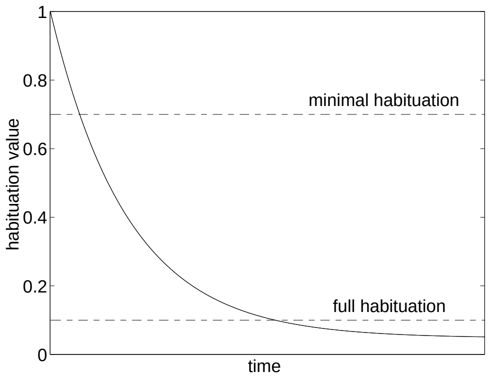


Figure 3.24: Habituation as exponential decay. Thresholds are used for minimal habituation (top dashed line) and full habituation (bottom line)

component that therefore needs to be controlled is how fast habituation occurs, which is given by the habituation rate τ. This parameter is a 'time' parameter, and depends on the time-scale of the particular implementation; it can be used to control how quickly a stimulus becomes more and more familiar until becoming 'completely' familiar. For example, for $\tau = 50$ it takes 19 steps to exceed the minimal threshold, and 138 steps to exceed the full threshold, and for $\tau = 100$ it takes 37 and 276 steps, respectively.

The Grow When Required Algorithm

Below is an algorithmic description of how the SOFM is built up from data, including how the above implementation of habituation is incorporated; see also Figure 3.25. The algorithm accepts instances drawn from an m-dimensional input space. At each step the current input (instance) ξ is used to update the SOFM, which consists of a set of n nodes with weights $\mathbf{w_i}$, $i \in 1 \dots n$ (where each $\mathbf{w_i}$ is m-dimensional), and a set of edges connecting the nodes; each node has a habituation value associated with it, and each edge has an 'age' associated with it.

The algorithm starts with an initialisation of two random connected nodes, and proceeds as follows, for each input:

1. Calculate the activity of each node in terms of the Euclidean distance[12] between the node and the input, as follows

$$a_i = exp^{-\|\mathbf{w}_i - \xi\|} \qquad i \in \{1 \dots n\} \tag{3.12}$$

[12]The Euclidean distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ of length m, is calculated as $\sqrt{\sum_{i=1}^{m}(a_i - b_i)^2}$, and is denoted by $\|\mathbf{a} - \mathbf{b}\|$.

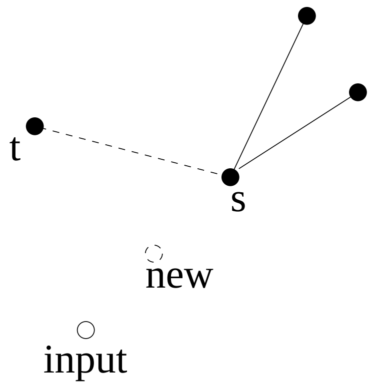


Figure 3.25: A schematic diagram of the Grow When Required (GWR) network. The network of nodes and edges is a topology-preserving representation of the input space. The nodes s and t are the ones that match the input best and second-best, respectively; an edge is inserted between these nodes, and represents a Hebbian connection between them; other nodes emanating from s are 'weakened'; a new node is added if required, and represents novelty; very weak edges are deleted, and disconnected nodes are deleted.

The node with the highest activation is referred to as the 'active', or 'winning' node, denoted by s in Figure 3.25.

2. Connect the winning node with the second-best node t, or if already connected reset the edge-age to 0; increment the age of all other edges emanating from s. This corresponds to Competitive Hebbian Learning.

3. Decide whether the input matches the winning node well: the activation, a_s, is thresholded for novelty detection, where high values (close to 1) correspond to familiarity and low ones (close to 0) to novelty. We refer to this threshold as the *novelty threshold* — it signals the first form of saliency, corresponding to novelty detection.

4a. If the input matches the winning node well, it moves towards the input by some fraction of the distance between them, similarly to the general update rule given in Equation 3.10, as follows

$$\mathbf{w}_s \leftarrow \mathbf{w}_s + \eta(\mathbf{w}_s - \xi) \tag{3.13}$$

where η is the learning rate ($0 \leq \eta \leq 1$). The neighbours of the winning node similarly move towards the input, but more slowly, *i.e.* using a smaller value of η. This is where existing SOFM nodes are updated and move in the input space to better represent the current input.

4b. If the input doesn't match the winning node well, it is *potentially* novel — the habituation value of the node is used to decide whether it is actually novel or not, as follows:

- if the node has only recently been added, its habituation value will be higher than the minimal threshold, and it is still being positioned in the input space, so don't add a new node, but rather update it as described in step 4a.
- otherwise, the node has fired a number of times, and has probably settled in the desired part of the input space, so a mismatch means novelty is detected, and a new node is needed; insert it half-way between the input and the winning node (see Figure 3.25), as follows

$$\mathbf{w}_{new} \leftarrow (\mathbf{w_s} + \xi)/2 \tag{3.14}$$

5a. Habituate the winning node and its neighbours according to Equation 3.11 as follows:

$$y_s \leftarrow y_s + \frac{\alpha(1-y_s)-1}{\tau} \tag{3.15}$$

where y_s is the habituation value of the winning node. The winning node habituates faster than its neighbours (lower value of τ).

5b. If a node is fully habituated (its habituation value is less than the full habituation threshold), 'freeze' the node: the node does not move from where it is, and cannot be deleted. Once a node has been frozen for a specific length of time, 'un-freeze' it by setting its habituation value to half the starting value, thus introducing 'forgetting'. We refer to this length of time as the *full-habituation* time, T; it signals the second form of saliency, corresponding to familiarity.

6. Remove edges older than the maximum allowable age; remove disconnected nodes.

To summarise, the system handles attention as follows. Nodes in the network respond and habituate to their respective stimuli; when fully habituated, nodes ignore further stimulation and hence do not get updated. Thus a stimulus is judged as salient if it is novel and until it becomes familiar enough, or if it is too familiar which forces a forgetting mechanism which makes the stimulus salient again until it becomes familiar enough, again. The former causes a new SOFM node to be created, while the latter causes the best-matching node to dishabituate. The psychological equivalent of this process, as described in Section 2.6, is that the orienting response is gradually inhibited to familiar experiences, and is then reinstated either due to novelty or forgetting.

parameter	step	effect
novelty threshold	3	mainly determines the number of nodes used, where low threshold values result in fewer nodes, but also in faster convergence (see toy example)
update rate (η)	4a	• if too low, network takes long to converge, because nodes move slowly towards the data; consequently more nodes are used. • if too high, also takes long to converge, because of erratic behaviour due to sensitivity to noise. • otherwise converges fast, using few nodes.
habituation (τ, T)	4b 5b	• if habituation too slow (high τ), desired nodes take longer to be added, resulting in late convergence. • if habituation too fast, nodes settle prematurely: - recoverable if full-habituation time is low - otherwise more nodes are added than needed, resulting in late convergence • otherwise algorithm insensitive to τ and T
maximum edge-age	6	if too low then nodes are removed too quickly, resulting in erratic behaviour: desired nodes are removed and inserted repeatedly; if high enough, the actual value is not critical.

Table 3.1: Summary of the GWR parameters and their effect on the algorithm.

3.3.3 Identifying Important Saliency Parameters

Altogether the algorithm is quite heuristic and involves several parameters, some of which arise from the choice to use the SOFM as an unsupervised learning tool, others are added to explicitly model the characteristics of an attention system. A summary of the parameters is given in Table 3.1, together with brief descriptions of the overall effect they have on the algorithm, where these are based on observations made from simple toy examples (see below) as well as task-related experiments (Section 3.3.4). The values used for these parameters in the various experiments in this chapter, and in Chapters 4 and 5, are shown in Table 3.2.

As mentioned in Chapters 1 and 2, it is important to identify the important saliency parameters and set their values carefully, so that the abstracted representation is faithful to the robot's experiences. Two of the parameters of the algorithm have been identified as the most sensitive and important for the attention system — the novelty detection threshold, and the full-habituation time; they are treated explicitly. The values of the remaining parameters are set through trial-and-error; however, this is not believed to introduce much ad-hoc bias to saliency, because the respective influences of all the parameters in the algorithm are dependent on each other — the two important parameters carry most of the responsibility of the attention system for detecting saliency, as explained below.

The novelty detection parameter is most influential for the qualitative purpose of attention mentioned in Chapter 2 and again at the start of this section — organising the structures used to represent the input usefully for learning; its properties are demonstrated in Figure 3.26 with reference to the Decision Space of Detection Theory mentioned in Section 2.6, together with some useful terminology for describing these properties. It measures the saliency of an input as its deviation from (dissimilarity to) the existing structures that represent previous inputs; this measure is used to control the update of existing structures and the creation of new ones. Therefore, care must be taken when setting the value of this parameter because the resulting SOFM corresponds to a compressed representation of the robot's experiences — its level of granularity determines how useful it is.

The full-habituation time parameter is most influential for the quantitative purpose of attention mentioned at the start of this section — modulating a response to familiar experiences. It measures saliency of an input as its familiarity — how often its corresponding node in the SOFM has been active. It does not affect the shape of the SOFM as much as the novelty detection parameter, but it does affect the number of inputs processed. It therefore deals with perceptual load associated with attending to data (and learning from data, as we will see in the next two chapters). Note that a quantitative benefit can also be associated with the novelty detection parameter, but of a different kind. For example, a sensitive novelty threshold produces many nodes in the SOFM — this might result in a desirable representation, but it requires more memory (to store the nodes) and computation (because at each step, the input is compared with all the existing nodes). The setting of this parameter might therefore be tailored not just to achieve a useful representation, but also to ensure memory usage and computation are within limits. This kind of perceptual load is discussed further in Chapter 5, but is not tested explicitly.

The two important parameters are tested explicitly in this chapter and the following two

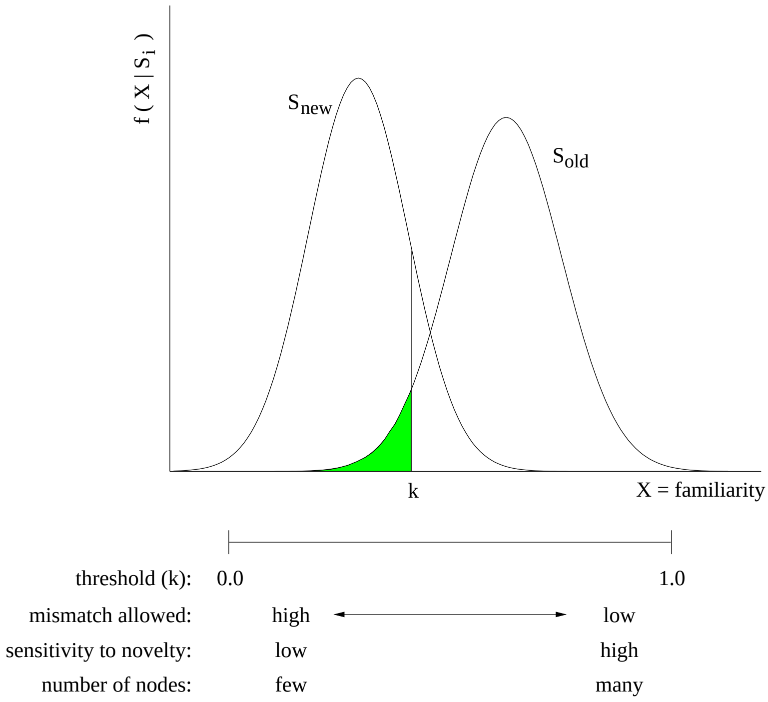


Figure 3.26: The properties of the novelty threshold: as the threshold increases the requirement for similarity between the input and the winning node increases (see Step 3 of the GWR algorithm), therefore only relatively smaller mismatches are allowed, the system is more sensitive to novelty, and consequently the system produces more nodes.

chapters. Of the parameters set ad-hoc, some have been identified as being dependent on the particular application and therefore requiring more careful tuning: the parameters η and τ depend quite heavily on the time-scale of the particular application, and therefore need to be re-tuned to fit that application. The remaining parameters have been identified as being more generic, and they have been set values that would work across different implementations: the choices of α (the parameter that affects the habituation asymptote) and the habituation thresholds were already discussed; the maximum edge-age is set just high enough to prevent erratic behaviour, and low enough to be usable by implementations with a low time-scale (100 steps).

In the remainder of this section the performance of the GWR algorithm, and the properties of the important parameters, will be demonstrated with the use of a small two dimensional toy example; the following section will present the implementation of the algorithm for the robotic tasks discussed in Section 3.2.2.

parameter		toy example	simulation wall-following	physical wall-following	object-interactions
novelty threshold		0.775 – 0.925	0.6 – 0.975	0.1 – 0.35	0.7 – 0.93
full-habituation time (T)		0 – 5000	5000 – 50000	0 – 10000	250 – 2500
update rate (μ) :	winning node	0.15	0.01	0.01	0.016
	neighbours	0.02	0.0006	0.0006	0.0002
habituation rate (τ):	winning node	50	300	50	200
	neighbours	150	600	150	400
habituation asymptote parameter (α)		1.05			
minimal habituation		0.7			
full habituation		0.1			
maximum edge-age		100			

Table 3.2: Parameter values used in the experiments reported in Chapters 3–5. The top section contains the parameters of interest that are examined extensively in the thesis; the middle section contains task-specific parameters that depend on the time scale of the particular implementation; the bottom section contains parameters that have been fixed to work across the different implementations.

Toy Example

We will now test the GWR algorithm using a simple two-dimensional toy example, where the input is drawn randomly from the three squares shown in Figure 3.27. The squares at the corners cover a larger area and are also more dense than the third, smaller one, but within each square the data are uniformly distributed. Recall that the data are fed to the algorithm one at a time; a total of 2500 points are used in this example. All parameter values are shown in Table 3.2.

Let us first examine the novelty threshold. Figure 3.28 shows SOFMs produced using various thresholds, chosen to demonstrate how they can affect the representational outcome:

1. choosing a very low (insensitive) threshold (0.775) results in modelling the data as one cluster with three nodes, attempting to approximate the space as best as possible;

2. choosing a slightly higher (more sensitive) threshold (0.81) results in treating the data as having two main clusters, where the small square simply forms a cluster with the big square at the top right corner;

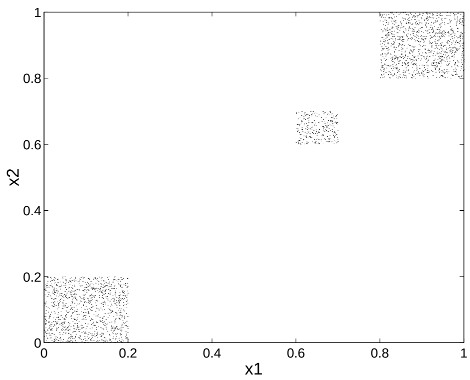


Figure 3.27: Data from the two dimensional toy example.

3. by increasing the threshold even higher (0.85) the small square is now represented better: it is now a component of the big cluster at the top-right corner; it does not quite belong to the cluster but is nevertheless similar enough to be attached to it;

4. with a high enough threshold (0.875) the data are treated as three distinct clusters; notice also that more nodes are now used in each cluster, and this is because of the more sensitive threshold: the slight dissimilarities within the clusters are being exposed, and with them the topology in the data;

5. a very sensitive threshold (0.9) results in plenty of nodes to represent much of the variation within the clusters;

6. as the threshold is increased further (0.925), the squares are almost entirely covered with nodes.

Deciding which of the thresholds is suitable will depend on how one wants to model these data: does the small square form a distinct, salient part of the space, is it an extension of the big square on the top-right corner, or is it simply noise? Should the data be regarded as coming from one, two, or three sources? These questions depend on the desired level of granularity, which is argued in this thesis to be task-dependent; that is, it depends on the purposes for which the representation will be used. The representations produced by the two extremes of Figure 3.28 are known as under-fitting versus over-fitting in the machine learning literature.

It was noted above that the novelty threshold influences how many nodes are produced by the algorithm. Figure 3.29 shows this relationship for different threshold values, where each

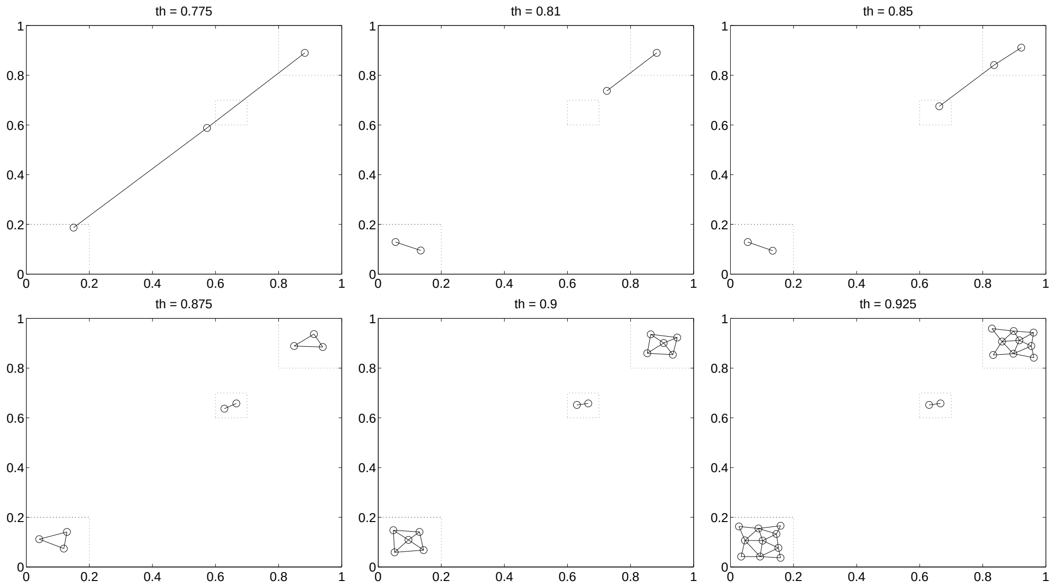


Figure 3.28: SOFMs produced for the toy example by different novelty thresholds, shown above each plot.

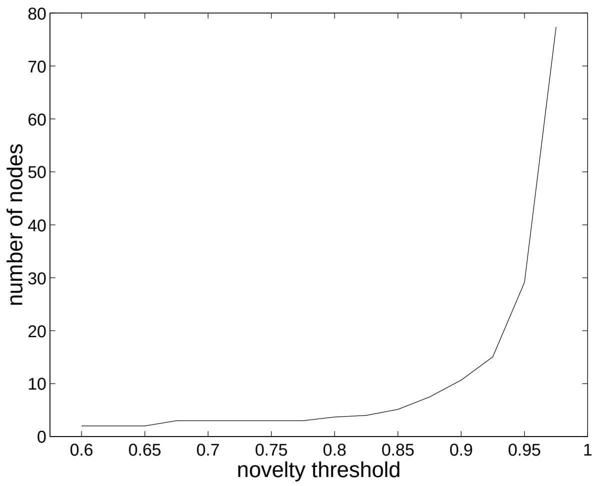


Figure 3.29: Number of nodes produced by GWR as a function of novelty threshold

is repeated 50 times (size of error bars is negligible and therefore omitted from the plot). The exponential shape of this relationship is related to the fact that the decision to add a node is based on a comparison of a similarity measure with the novelty threshold, where this similarity measure is an exponential function (Equation 3.12) — see step 3 of the algorithm.

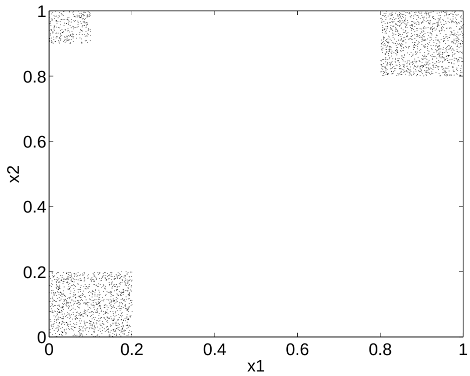


Figure 3.30: Data from the second part of the extended two dimensional toy example.

We will next examine the full-habituation time. For this we extend the dataset by another 2500 points, where now the small square moves to a different location, as shown in Figure 3.30. The algorithm thus receives 5000 points, the first 2500 being drawn randomly from the three squares as before, and the next 2500 from the same big squares but now from the new small square instead of the old one. One might consider that there is now a new state, and an old one is no longer present.

Figure 3.31 shows SOFMs produced with different full-habituation times (novelty threshold is 0.9 in all). The leftmost plot demonstrates the case where, once fully habituated, nodes do not dishabituate (since the full-habituation time is 5000, which is the length of the run). We see that the SOFM was able to produce a representation for the new square, however it also maintained the representation for the old one: there is no forgetting. Further, the old and new squares appear to be similar in the space (using this particular novelty threshold) as they are connected. Due to the topology preserving nature of the algorithm, one would therefore expect the nodes of the old square to move towards the new one because data are only appearing now for the new square. However recall that the nodes are fully-habituated and therefore cannot move. In the middle plot, where the full-habituation time is half the length of the dataset, the SOFM is now able to recruit the nodes used to represent the old square for the new one; one of these nodes in fact now contributes to the representation of the new square. Finally, as the full-habituation time drops to a very low value (500), the SOFM is able to completely forget about the old square, and adequately represent the new one; notice that the final node is not actually required in the final representation and is therefore deleted.

What might be a good choice of full-habituation time when we consider that with a full-

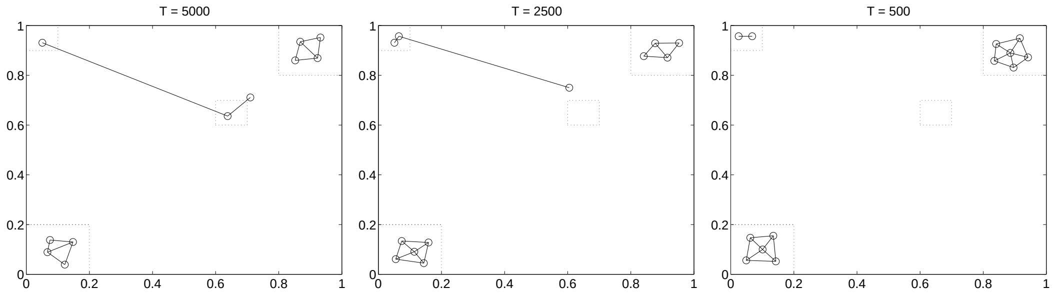


Figure 3.31: SOFMs produced for the toy example by different full-habituation times, shown above each plot.

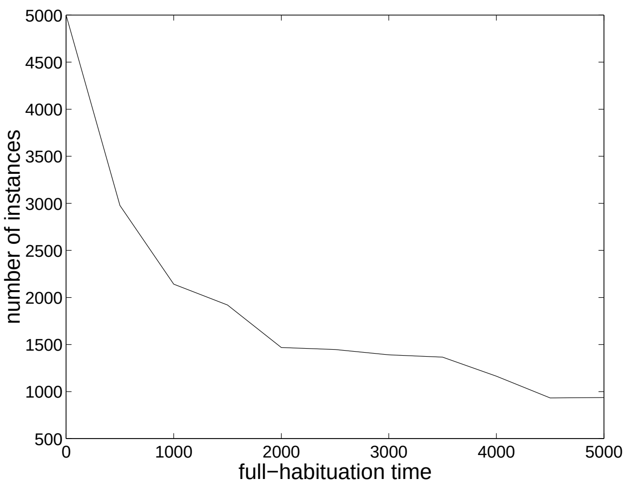


Figure 3.32: Number of instances attended to as a function of full-habituation time

habituation time of 0 the use of habituation does not add much to the algorithm, while on the other hand a maximal full-habituation time takes away the plasticity of the network? Habituation serves to inhibit a response to familiar stimuli; a full-habituation time of 0 fails to achieve this, while a maximal full-habituation time over-achieves it at the expense of being adaptive to non-stationary data sources; there is a trade-off here which must be resolved. As for the decision of the novelty detection threshold, a desirable value for the full-habituation time is task-dependent.

To see more clearly the benefit of modulating repetitive data, we look at the number of instances attended to by the SOFM; recall that the SOFM attends to input unless the winning node is fully habituated (step 5b of the algorithm). Figure 3.32 shows the total number of instances attended to by SOFMs using different full-habituation times, where for each value the experiment is repeated 50 times (error bars are negligible and therefore omitted from the plot). We see that even a low full-habituation time such as the one used in the right-most plot of Figure 3.31 reduces the number of instances to almost half. In other words, modulation does not need to be at the expense of plasticity. We will also see in the next chapter that modulation does not need to be at the expense of learning performance.

As a final note on plasticity, note that the fact that the update rate η does not decrease over time (as in the classical SOFM algorithm) is crucial to the ability of the network to forget and adapt to new situations[13]. Also, one should note that the habituation 'freezing' mechanism imposes almost arbitrarily on the natural plasticity of the network, and the result is that sometimes nodes do not settle in their 'natural' positions. For example in the left-most plot of Figure 3.31, the top-right node of the cluster for the old square has settled outside the range of the data. The fact that the SOFM never completely naturally converges is only a problem if one relies on the nodes to accurately reflect the salient parts of the input space. In this thesis, we will see that this is not a problem because high accuracy is not needed as the attention system serves to abstract raw data for a separate learning architecture, which then generalises from these abstractions.

3.3.4 Implementation on Robotic Tasks

In this section we will consider the tasks mentioned in Section 3.2, where the data presented in that section are exactly the data used to train the SOFMs shown in this section. In contrast to the method (PCA) used to analyse these data in Section 3.2, here the SOFM takes continuous input directly from the sensors in an *on-line* manner, *i.e.* one at a time in the order they are perceived. Because the PCA approach gave us an idea of what the data look like, we can observe the output of the GWR algorithm critically with respect to the data. As noted, the dimensionality of the SOFM is the same as the data, and so in order to visualise it we must project it onto a lower dimension. In the plots shown below, the SOFMs are projected onto two dimensions using the principal components found in the corresponding datasets in Section 3.2, and therefore the spatial locations of the nodes and edges are directly comparable to the data points. Therefore, in the interpretation of the SOFMs below we will repeatedly refer to the plots in Section 3.2. All parameter values are shown in Table 3.2 for each task.

[13] Note that in theory, even when using a decreasing learning rate one is guaranteed to reach *any* solution, however this guarantee only applies for an infinite number of learning steps.

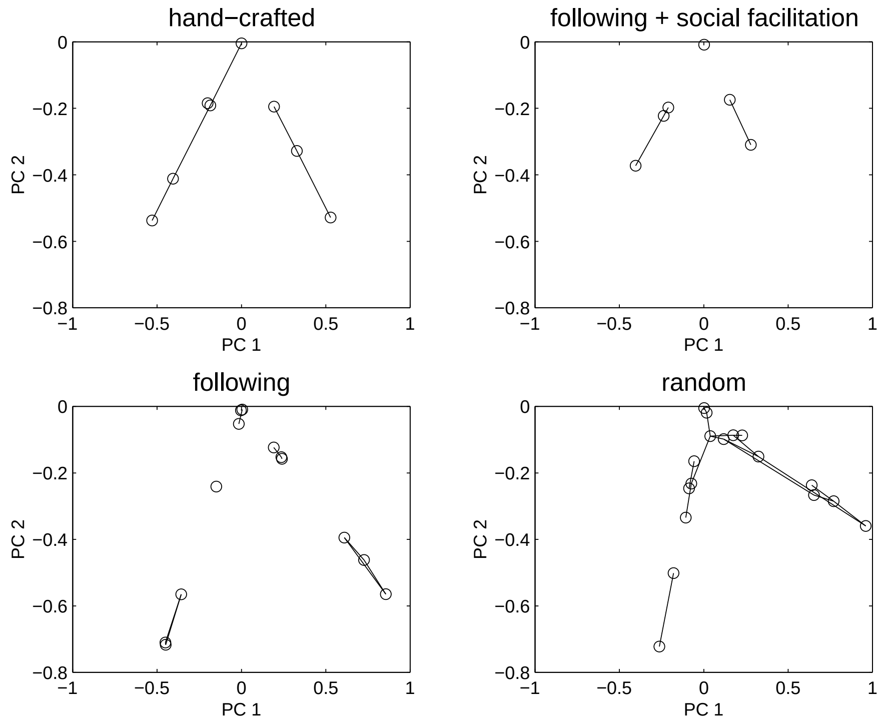


Figure 3.33: SOFMs produced for wall-following in simulation, for the four scenarios; for visualisation purposes the SOFMs are projected onto the first two principal components found in the data by PCA as in Figure 3.11.

Wall-Following

Figure 3.33 shows the SOFMs produced for the simulation wall-following task, with the four different scenarios described in Section 3.2, using a novelty threshold of 0.75 and a full-habituation time of 10000 (the remaining parameter values are shown in Table 3.2). The number of nodes in these SOFMs is 8, 6, 14, and 17, respectively.

The first point to note from Figure 3.33 is that the clusters in the data, as shown in Figure 3.11, are captured by the nodes and edges of the various SOFMs. The most immediate distinction between the four SOFMs is that they have a different number of nodes, even though the parameter values are the same in all. The reason for this is that the distribution of the data is different, as we have seen in Figure 3.11, therefore we expect them to be modelled differently. Specifically, the hand-crafted and social-facilitation scenarios are exposed to a smaller variety of data, and so one would expect these to have fewer nodes; as explained in Section 3.2, in the social-facilitation scenario the learner does not perceive the wall at very close distances as in

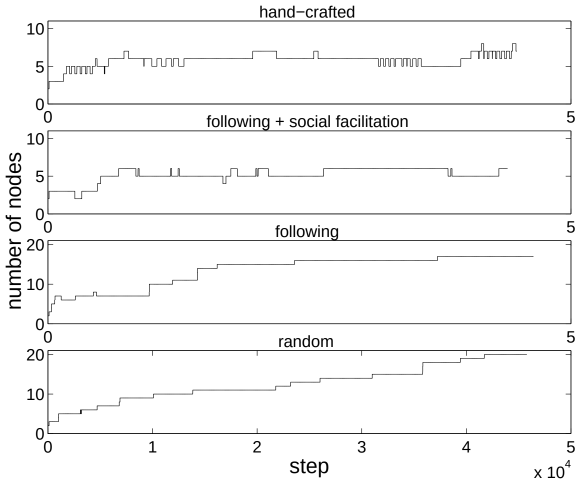


Figure 3.34: SOFM convergence for wall-following in simulation, for the four scenarios, ordered from top to bottom.

the hand-crafted scenario, and hence no nodes are produced for those extremes; and finally, the distribution of the data in the following scenario is closer to that of the random scenario, and in fact in very unlucky situations where the learner loses the demonstrator frequently the SOFM indeed looks more similar to that of the random scenario.

To get an idea of the progression of the SOFM during each run, we look at the number of nodes at each step during the run for the four different scenarios, in Figure 3.34 (plots are ordered from top to bottom). We see that the hand-crafted and social-facilitation scenarios converge to a smaller number of nodes, and also converge earlier; the following scenario initially seems to converge to a few nodes as in the social-facilitation scenario, but then at some point it needs more nodes to represent the data perceived when the robot loses the demonstrator; the random scenario is exposed to the wall at many different angles so it takes longer to converge, however one would expect it to converge eventually. Overall convergence is not precise, for example after the hand-crafted and social-facilitation scenarios seem to converge one or two nodes are added and deleted for the remainder of the run. This is not surprising when we recall that the GWR algorithm prefers plasticity over stability, as explained in the previous section (the update rate η is fixed).

For the physical experiments involving the robot Gillespie and a human demonstrator, the SOFMs produced for the four scenarios mentioned in Section 3.2 are shown in Figure 3.35; here the novelty threshold is 0.225, and the full-habituation time is 1000 (the remaining parameter

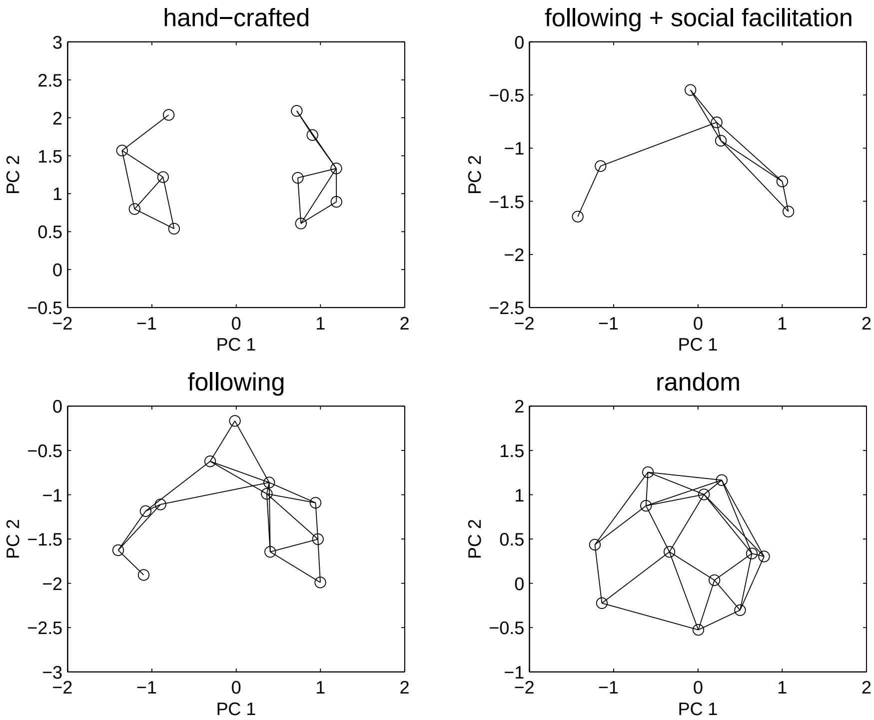


Figure 3.35: SOFMs produced for wall-following in the physical experiment, for the four scenarios; for visualisation purposes the SOFMs are projected onto the first two principal components found in the data by PCA, as in Figure 3.14.

values are shown in Table 3.2); the number of nodes in these SOFMs is 11, 7, 12, and 12, respectively. As in the simulation, we see that the perceptual clusters shown in Figure 3.14 are captured by the various SOFMs. Specifically, notice that the lack of sufficient data for the 'no wall' cluster in the hand-crafted scenario is confirmed by a lack of SOFM nodes. This shows clearly the benefit of active demonstrations: at this particular level of granularity — specified through the novelty threshold — the demonstrator forces the no-wall experiences to be salient through high exposure. Of course we could force the creation of these nodes by using a finer level of granularity — by increasing the novelty threshold — but this also would force more nodes elsewhere, which could be redundant and increase the computational requirements, as discussed in the previous section.

The SOFMs produced for the social-facilitation and following scenarios are similar to those produced in the simulation (Figure 3.33), which is encouraging as they are modelling the same task. What is interesting is that we have had to use a considerably lower novelty threshold in

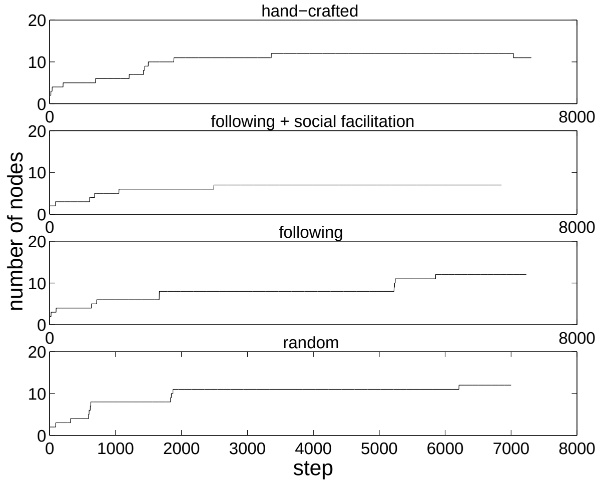


Figure 3.36: SOFM convergence for wall-following in the physical experiment, for the four scenarios, ordered from top to bottom.

order to achieve similar representations in the simulation and physical scenarios. The reason for this is that the physical data are much noisier and less well-structured, and therefore one must be a lot stricter when detecting novelty (*i.e.* less sensitive or equivalently willing to accept higher mismatches when judging similarities — see Figure 3.26) in order to avoid modelling the noise. Of course, this means that each node is therefore a very noisy representation of the data it is modelling. This was also demonstrated in Figure 3.10 where we saw that a two dimensional projection on the complete datasets was much poorer in the physical experiments than the simulated ones, in accounting for the variance in the data.

As for the simulation experiment, let us now look at the convergence of the SOFMs, in Figure 3.36 (again, the plots are ordered from top to bottom). It seems that the hand-crafted and social-facilitation scenarios converge earlier, as in the simulation.

Phototaxis

Figure 3.37 shows the SOFMs produced for the phototaxis task with the three different scenarios described in Section 3.2, and Figure 3.38 shows the convergence of these SOFMs (ordered from top to bottom). All but one of the parameter values used to generate the SOFMs are exactly the same as for the simulation wall-following task. Since the platform (Khepera simulator) and the sampling rate are the same in both tasks, the same parameter values should work reasonably well here; of course, the actual tasks, and hence the distributions of the data,

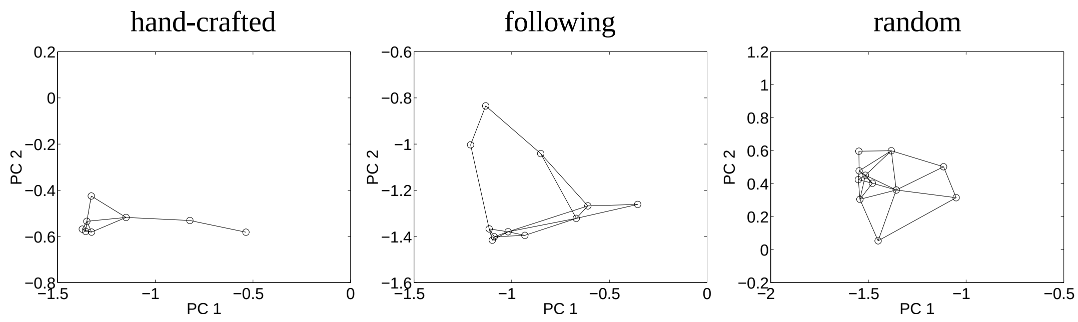


Figure 3.37: SOFMs produced for phototaxis, for the four scenarios; SOFMs are projected onto the first two principal components found in the data by PCA as in Figure 3.17.

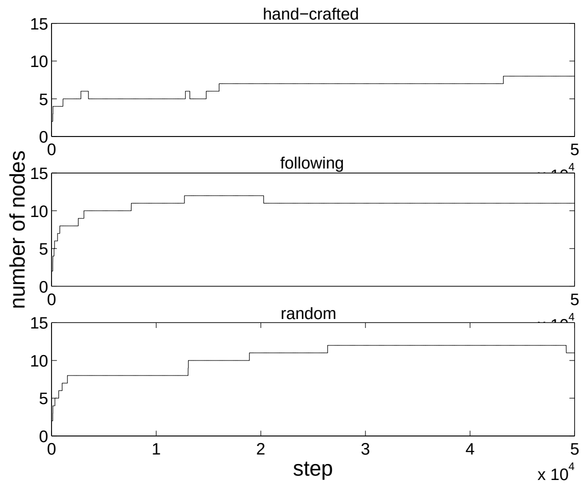


Figure 3.38: SOFM convergence for phototaxis, for the three scenarios ordered from top to bottom.

are completely different, and therefore one parameter must certainly be modified: the novelty threshold; the value used here is 0.55.

As explained in Section 3.2, and demonstrated in Figure 3.17, there is only one distinct cluster of points for this task, surrounded by many points corresponding to transitions to and from the distinct ‘no stimulation’ state. This is also demonstrated by the SOFMs in Figure 3.37: there is a higher density of nodes for the distinct cluster, and overall there are no distinct SOFM clusters, instead all the nodes are connected in some way to the high-density cluster. As suggested by the data inspections in Section 3.2.2, in order to see the more interesting differences of detecting the light at different angles, a finer level of granularity would be required, that is, a higher novelty threshold.

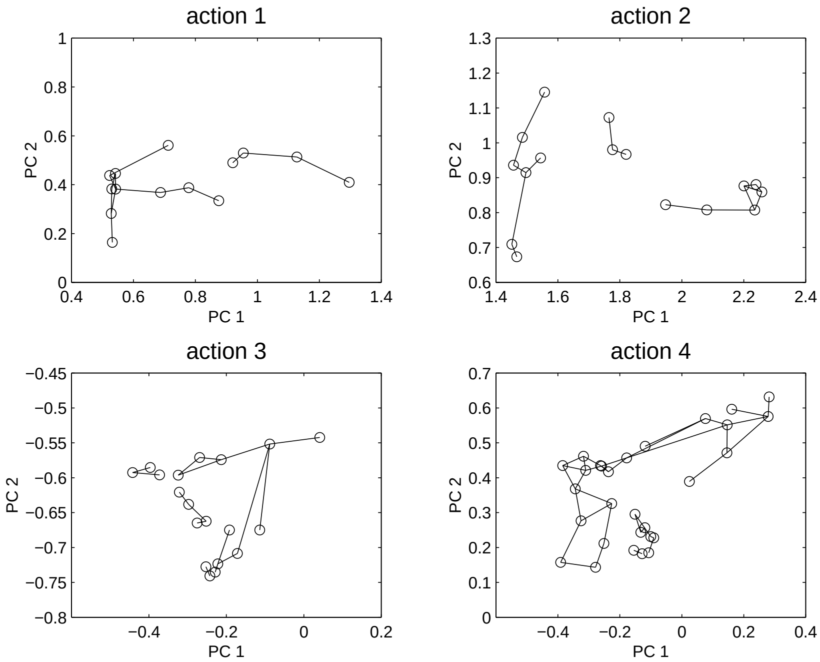


Figure 3.39: SOFMs produced for the four different object-interactions; SOFMs are projected onto the first two principal components found in the data by PCA as in Figure 3.22.

Object Interactions

Figure 3.39 shows the SOFMs produced for the four different object-interaction tasks described in Section 3.2, and Figure 3.40 shows the convergence of these SOFMs (ordered from top to bottom); the SOFMs are produced with a novelty threshold of 0.89, and a full-habituation time of 1000 (the remaining parameter values are listed in Table 3.2).

As in the other robotic tasks presented in this section, here too the GWR algorithm captures the structure of the data, in all four object-interactions, and in particular the slight difference between actions 1 and 2 (see Figure 3.22). As in the phototaxis task, distinct clusters of SOFM nodes are not easily distinguishable, and this is due to the sequential nature of these tasks: data are presented as a temporal sequence of postural instances, so at each particular location in the input space the density of data is relatively small, unless the demonstration slows down in which case there is more sampling. This point was mentioned in Section 3.2.2, where it was also pointed out that in fact there are sub-components in the first and second interactions where slowing down of the action does exist: when the hand reaches the object, half way between reaching the mouth, and of course at the mouth itself (where there is also a change in direction).

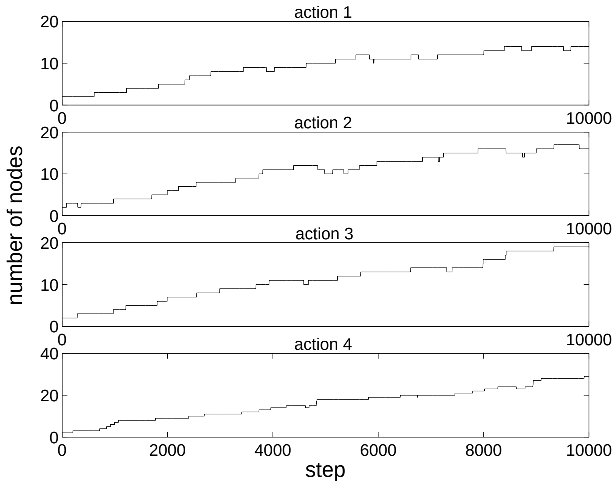


Figure 3.40: SOFM convergence for the four different object-interactions, ordered from top to bottom.

We can see clusters in the SOFMs of actions 1 and 2 that correspond to such components, but they are not always so clear. For example the middle cluster of three nodes in action 2 is not distinct for action 1, although the nodes do exist; a possible explanation for this is that because in action 2 the top part of the input space is further away (because the elbow is further away from the body — see Figure 3.22), it drags the nodes from the middle part enough to separate them from the rest of the map; this also seems to affect the remaining parts of the map.

From Figure 3.40 we see that the SOFMs do not have enough time to converge, although actions 1 and 2 do seem to begin to converge. The instability of the algorithm is more noticeable in these tasks than in the previous ones we have seen; this is again due to the strong sequential nature of the data. Nodes are pulled in one direction as a particular part of the action is demonstrated, and then pulled back again as the same part is demonstrated again later (new 'episode'). Figure 3.40 shows that with the particular parameter values used, more than (approximately) five 'episodes' are required for the nodes to settle. However Figure 3.21 suggests that convergence after two episodes *should* be possible, perhaps with different settings of the parameters.

3.3.5 Conclusion

In this section we have looked at different SOFMs produced by the adapted GWR algorithm for the different tasks and scenarios, and compared them to the actual perceptual data being modelled, where these were presented in Section 3.2. These comparisons confirmed the conclusions from the data inspections (see Section 3.2.3), namely that the different tasks and scenarios involve different levels of granularity and therefore require different measures of saliency; and the benefit of social interactions were confirmed, firstly by showing that the imprecise teacher-following behaviour results in the representation of undesirable experiences, and secondly that the active demonstrations are helpful in accentuating salient parts of the task and therefore in their representation.

3.4 Discussion

It was claimed in the beginning of this thesis, in Chapter 1, that saliency must be biased by an external source. The aim of this chapter has been to demonstrate what is involved in biasing the detection of saliency, and to give an empirical backing to the above claim by showing that saliency is relative to the particular task and environment. In Section 3.2, perceptual data from various tasks and environments were analysed statistically and significant variations were exposed in the data at different levels of granularity. Section 3.3 showed that an attention system, responsible for saliency detection, requires its saliency parameters to be set differently when the same task is implemented on different platforms (that is, in different environments), and when different tasks are implemented on the same platform (that is, in the same environment). The chapter has also started to address the issue of balancing designer effort in setting saliency with social interactions, by showing that the external source of bias for saliency can be the designer who sets saliency parameters, and it can be the demonstrator who accentuates the salient parts of the task.

The activities performed in Section 3.2 contribute to achieving the purposes of this chapter mentioned above, but are also proposed as useful general methodologies. As discussed in Section 3.2.3, the basic analysis of the data using PCA is relatively straightforward but very informative. It is good practice to inspect the data that the robot is exposed to, and have an idea of the statistical nature of these data, because this gives some indication of the ability to learn from them.

Section 3.3 presented the kind of methodology that is claimed in this thesis to be required in order to balance designer effort through social interactions. First, a parameterised attention system was implemented for detecting saliency in raw perceptual data, and self-organising these data. Then, the important parameters were identified, and their role in biasing the representation of the data was tested on a small toy example, and on the robotic tasks. These parameters were isolated from the others, as being most influential in detecting saliency subjectively for the particular task and environment. Therefore, having identified them enables the designer to bias the robot's notion of saliency reliably, as summarised below. It also potentially enables the design of mechanisms with which the values of the parameters can be tuned through explicit signalling and communication with the demonstrator, and therefore be adapted to different levels of granularity as required for specific parts of the task; this is described as an extension to the work in this thesis, in Section 6.4.3. The design of such mechanisms is possible because the parameters model saliency explicitly and directly in terms of the the robot's *current* experience, which motivates the choice of the GWR algorithm for the implementation of the attention system (see Section 3.3.1).

The identification of the parameters in Section 3.3.3 included a discussion of the effort required by the designer in usefully biasing the detection of saliency. Namely, it was mentioned that care must be taken by the designer when setting the novelty detection parameter so that the resulting representation in the SOFM is a useful compressed representation of the raw data. In Chapter 4 we will see a learning setup where this effort is not essential because the learning occurs on the raw data, and therefore the representation in the SOFM is not crucial. Instead, the role of the habituation parameter will be stressed, for modulating learning in the face of limited learning resources. In Chapter 5 we will see examples where the learning occurs at a high level of abstraction, based on the structures discovered by attention, therefore involving more careful designer effort. Recall that attention in this thesis only applies to perceptual data from the robot's sensors (see Section 2.4.2); the examples in Chapter 5 examine designer effort beyond setting the attention parameters, by identifying different amounts of effort involved in abstracting the robot's *motor* data.

A demonstration of how social interactions can be used to balance designer effort (the aim of this thesis) will be possible when showing the *usefulness* of the saliency bias for learning. Such a demonstration will be given collectively through the experiments in the following two chapters, in terms of learning performance. This collection of experiments and their results is summarised in Figure 3.41 using the space identified in Chapters 1 and 2. The 'alone & social' type of social interactions refers to the learning without social facilitation scenario discussed in this chapter (see Section 3.1), that is, learning even when the demonstrator is lost; the 'alone' type refers to the random scenario, that is, learning through a random wandering behaviour; the 'explicit signalling' scenario was not discussed in this chapter — it is introduced in the next chapter. The bottom section of the designer effort axis corresponds to learning from raw data, where the main role of attention is to modulate learning; the middle section corresponds

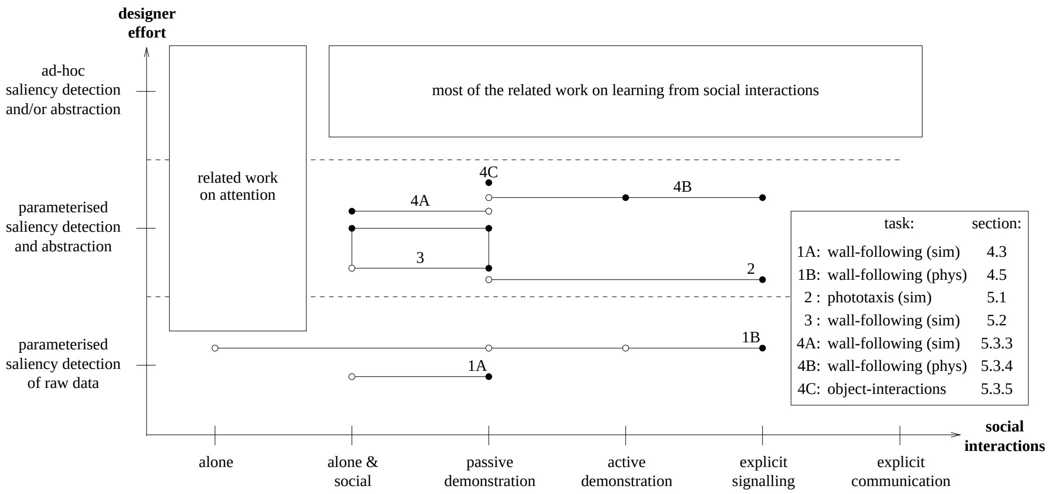


Figure 3.41: The experiments reported in the thesis examine social interactions of different complexities, and learning at different levels of abstractions requiring different amounts of designer effort. The experiments that are actually performed are marked with circles, with a solid circle marking the best performance achieved. Within the second category on the vertical axis, the vertical distinction relates to different amount of effort required by the designer, except that the distinctions within experiment set 4 are purely for visual purposes. Similarly, the vertical distinctions within experiment set 1 are for visual purposes only.

to learning from structured data, where the main role of attention is to structure or abstract the raw data.

Four sets of experiments are characterised — the first is presented in Chapter 4, and the remaining three in Chapter 5. The figure shows all the experiments that have been performed (marked with circles), and where the best performances were achieved in each experimental set (solid circles). A detailed discussion of the results will be given in Chapter 6 after all the experiments have been presented. In brief, this collection of results shows that it is possible to improve performance by moving up the vertical axis or across the horizontal axis; that is, it shows that the designer effort can be balanced by the social interactions. As will be shown, there is an ordering in the amount of effort by the designer from the first set of experiments to the fourth one, and thus the best results achieved through more effort by the designer can be matched by stronger social interactions and less effort by the designer. Note that the experiments will be presented in 'reverse' order to the argument above: they will show that when little effort is given by the designer, there is a need for stronger social interactions, and this need is diminished as more effort is given by the designer.

Chapter 4

Learning at a Low Level of Abstraction

This chapter presents experiments where the learning setup involves the least amount of designer effort compared to the other experiments in this thesis, presented in the next chapter. The learning here occurs on the raw perceptual data of the learner robot; therefore, as opposed to the experiments in the next chapter, there is no effort required by the designer in abstracting the data prior to learning. This means that the robot can potentially learn from *all* its experiences, depending on whether the learning architecture can generalise well from such noisy and unstructured data. However, even if this is the case, there is still an argument for utilising attention in order to select certain experiences over others and reduce the load on the learning system, so long as this does not significantly reduce the learner's ability to generalise. Such selectivity would be useful if, for example, there is a limited availability of resources for processing experiences by the learning architecture.

Utilising an attention system means that *some* designer effort is required in determining how to select experiences, that is, determining the saliency of experiences[1]. However, because the learning architecture generalises from the *raw* data, the issue of setting saliency reliably is not so crucial, as opposed to when learning occurs on data abstracted through saliency, as mentioned in the previous chapter.

The experiments presented in this chapter correspond to 1A and 1B in Figure 4.1. They address learning from the raw data, where the designer effort is expressed as setting saliency parameters for modulating the amount of learning usefully; they will compare such effort with *no* effort, which corresponds to no modulation, and show the implications for learning resources, as mentioned above. The two experiments address different types of social interactions, as will

[1] See start of Section 1.1 on page 3 for the definition of 'saliency'.

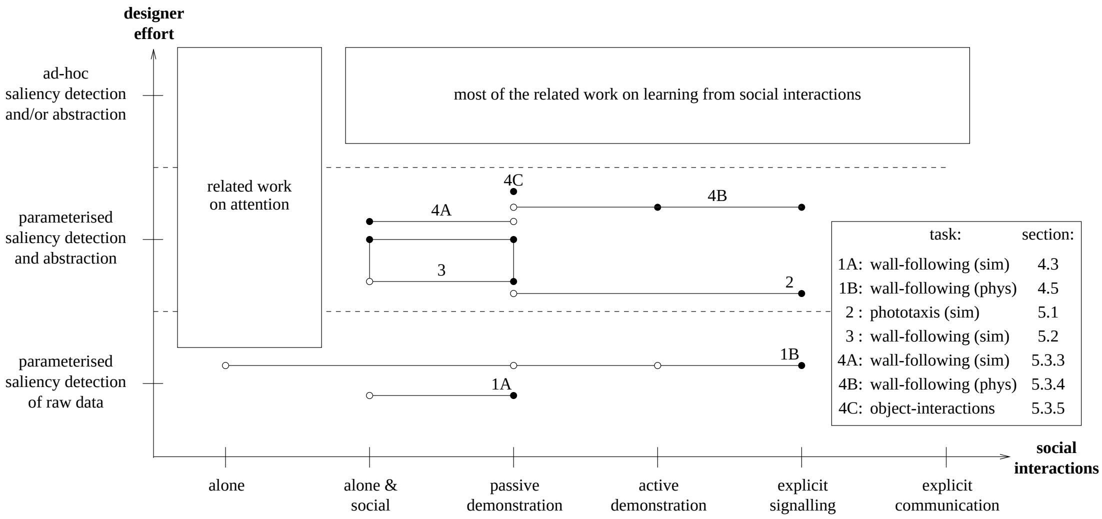


Figure 4.1: This chapter presents experiments 1A and 1B shown in the figure. They involve learning at a low level of abstraction — learning from the raw data. The designer effort is expressed here as abstracting from the raw data for the purpose of modulating the amount of learning, where the abstraction is performed for the purpose of deciding when experiences are familiar. The attention system presented in the previous chapter is used for this purpose — it measures familiarity through its habituation parameter. The experiments address the different types of social interactions that are needed to balance this low amount of designer effort compared with the experiments in the next chapter.

be discussed in the respective sections. This chapter will show that because little effort is given by the designer, more effort is required from the expert compared to the experiments in the next chapter — see experiments 2–4 in Figure 4.1.

We will use attention to modulate learning by selecting experiences based on their familiarity. This requires abstraction to be performed reliably to some extent, but the amount of attention that is re-given to familiar experiences is the more crucial factor — we will test its full range. The attention system presented in the previous chapter will be used here to control the amount of attention given to familiar experiences through its habituation parameter, and thus modulate learning by triggering a separate learning system whenever the SOFM node activated by the current input is not fully-habituated, as shown in Figure 4.2. Recall that a node has a high habituation value either if the input is novel, or if the input has been perceived many times and forgetting is introduced by dishabituating the node.

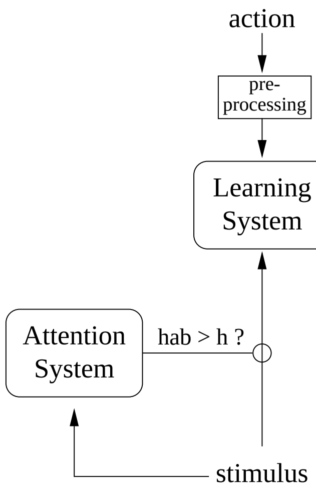


Figure 4.2: Attention-triggered learning. The attention system described in the previous chapter is used here merely as a trigger mechanism: it signals when a separate learning system should learn to associate raw perceptual input with action. Learning occurs when the SOFM node activated by the input is not fully-habituated (the habituation value of the node, *hab*, is greater than *h*, the full-habituation threshold).

The experiments presented here correspond to learning the wall-following task in the simulation and physical environments introduced in Section 3.2.2. The chapter starts with Section 4.1 presenting the learning setup and describing exactly what is being learned; Section 4.2 discusses the design choices specifically for the learning setup here, and more generally in the thesis; Section 4.3 presents the implementation of this setup in the simulated experiments; Section 4.4 shows the modulation benefit of attention in the face of limited learning resources; Section 4.5 presents the physical experiments; and Section 4.6 concludes the chapter.

4.1 Learning Setup

In order to test the modulation benefit of attention, a very simple learning setup is devised. The system is trained to predict *one* value, which corresponds to a 'tendency' to turn. The desirable outcome for this task is that the network learns to output a low turn-tendency (*i.e.* a forward move) when the robot is parallel to a wall, and a high turn-tendency (*i.e.* moving around randomly with turning) when not next to a wall. The network is *not* expected to learn how to control the robot to turn towards or away from a wall. Instead, as we will explain in

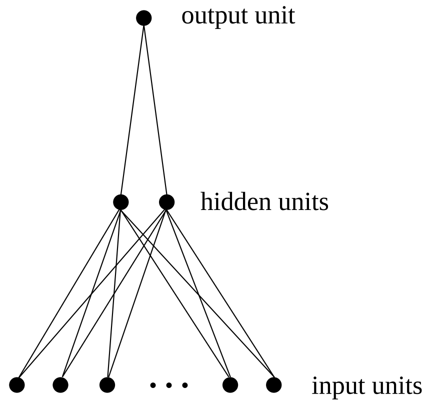


Figure 4.3: The learning architecture is a multi-layer feed-forward neural network.

Section 4.3 for the simulated experiment and Section 4.5 for the physical experiment, in the recall phase the robot is equipped with an obstacle-avoidance behaviour that turns it when it faces a wall; a correctly trained network will then control the robot to move forward when it is parallel to the wall, whereas in incorrectly trained network will result in the robot moving away from the wall.

The learning architecture is a multi-layer, fully-connected, feed-forward neural network (a multi-layer perceptron), shown in Figure 4.3 (henceforth referred to as the feed-forward network), trained with the back-propagation algorithm (see, for example, Hertz et al., 1991). The input layer consists of units representing the perceptual sensors (six units in the simulation and 20 units in the physical robot), the hidden layer consists of two units (chosen empirically), and the output layer consists of one unit corresponding to the tendency to turn as described above. The weights connecting these layers are randomly initialised to values between -0.5 and +0.5.

Motor Pre-processing

The target value for the output unit that is used to train the network is obtained through some pre-processing on the motor values of the learner robot as follows: the encoded motor values are 0, -1, and 1, corresponding to a forward move, right turn, and left turn, respectively; the current motor value is saved into a short-term memory window on which a moving average is calculated; the target-value y used in the output unit is the absolute value of this average, and is therefore a real number between 0 and 1, as shown in Equation 4.1:

$$y(t) = \left| \frac{1}{k} \sum_{t'=t-k}^{t} a(t') \right| \qquad a(\cdot) \in \{-1, 0, 1\}, \tag{4.1}$$

	input-hidden	hidden-output
simulation experiment	0.9	0.5
physical experiment	0.6	0.01

Table 4.1: Weight-update learning rates used in the simulation and physical experiments, shown separately for weights connecting the input layer to the hidden layer, and weights connecting the hidden layer to the output layer.

where $y(t)$ is the target output value calculated at time t, k is the size of the moving window (15 steps in the experiments), and $a(t)$ is the encoded motor value at time t. The absolute value is used since we are not interested in the direction of turn, only in 'tendency to turn'; this has the effect of smoothing (averaging) out any zigzagging exhibited by the robot as it follows the demonstrator, which is especially useful with the physical robot Gillespie.

The Operation of the Learning System

In the learning phase, whenever the learner is attentive its raw sensor values and (processed) motor values are used to make up a supervised learning pattern for the feed-forward network, as depicted in Figure 4.2. The input is used in a feed-forward manner to compute an output using the current values of the weights, where both the hidden and output layers use a sigmoidal transfer function[2] (*i.e.* the computed value of a unit in each layer is a sigmoidal function of the weighted sum of the units in the layer below it). The computed output is compared with the motor target-value presented to the network ($y(t)$ in Equation 4.1), and the error is back-propagated in the usual way from the output unit to the hidden units and then to the input units, resulting in weight updates. The role of the weight updates is to improve the predictive power of the network; the weights are adjusted in the weight space in a gradient-descent manner, controlled by learning rates. The learning rates used in the experiments are shown in Table 4.1.

In the recall phase the robot is placed in the environment on its own, the weights of the feed-forward network are fixed, and the sensor values are fed to the network *at each time step*, therefore the attention system is *not* used, as depicted in Figure 4.4. The input is then used in a feed-forward manner as in the learning phase to compute a value for the output unit, which is then translated into an action; the output translation procedure is described in Section 4.3. One *could* in fact use the attention system in the recall phase and choose to ignore stimuli

[2]The sigmoid function, f, is a continuous approximation to a step function. It is given by $f(x) = \frac{1}{1+e^{-Dx}}$ where D is a constant (0.9 in the experiments).

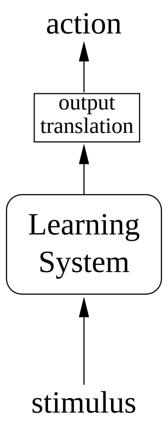


Figure 4.4: Testing the learning system in the recall phase does not involve the attention system.

that do not activate a SOFM node well enough; such situations could arise even though the environment in the recall phase is exactly the same as in the learning phase, because the robot will have some perceptions that it could not have had when following behind the demonstrator in the learning phase, such as sensing a wall straight in front. Therefore, with the setup shown in Figure 4.4 the system is not expected to deal well with unseen situations, but is nevertheless allowed to attempt to produce a response as best it can for *all* situations in the recall phase; additional hand-coded behaviours are used to safe-guard the robot from dangerous situations, as described above.

To summarise, the learning system is set up to achieve the following. The network is trained to predict one value, and this is aided in the learning phase with pre-processing on the motor values before presentation to the network. All the robot learns is to move forward or not, it does not learn how to get itself into a situation where a forward move might be predicted. Therefore in the recall phase, 'wall-following' can only occur if the output from the neural network is bootstrapped with hand-coded behaviours that move the robot to and away from the wall.

4.2 Explanation of Design Choices

The setup of the learning system described above might be criticised for being rather simplistic, and that on its own, the learning system does not achieve very much. Such criticism might in fact be made for all the learning setups in this thesis. The motivation for designing a simple learning setup is as follows. The objective in this thesis is to examine what learning performance can be achieved and how it varies and can be improved, as a function of designer effort and social interactions given *any* learning setup. The objective is *not* to design an optimal, self-contained learning system for the particular tasks covered. In fact, simplifying the learning system as much as possible is crucial to having the ability to infer the role of designer effort and social interactions, because there are fewer factors to control for. It is easier to understand and explain the behaviour of a simpler learning system, and therefore expose the effect of the designer effort and social interactions that bootstrap it. In particular, it is easier and more correct to draw conclusions regarding the attention parameters when one can control and dissociate the side-effects of the parameters of the learning system from these conclusions (such as learning rates and number of hidden units).

These justifications also hold for the choices of the actual learning *architecture* itself. That is, not only is the learning *setup* simplified, but also the architecture is chosen to be relatively simple with few parameters. The actual choice of architecture is rather arbitrary in the sense that it is not chosen to serve a particular function better than some other architecture. Learning architectures are chosen which are standard and have proven to work well in similar situations. The MLP with back-propagation error training is one such example.

Thus simple learning setups and architectures are desirable for the investigations in this thesis. However, it will be argued that the conclusions from these investigations are nevertheless applicable to more complex learning scenarios. We will see in Section 5.3 another learning approach where an emergent outcome of more designer effort is that the robot learns also to turn towards the wall; this outcome has implications for more effort from the expert demonstrating the task. A more complex learning scenario simply means that the learning is more difficult and so more effort is required in training the robot, where this effort can be balanced between the designer and the expert. For example, the complexity of the learning setup in this chapter might be increased by adding another output unit to represent direction of turn; the learning network would then need to learn a more complicated function, which would be more difficult, and might require more careful setting of the learning weights by the designer, or more care from the expert in demonstrating the salient parts of the task.

4.3 Implementation on the Simulated Wall-Following Task

This section presents the set of experiments depicted in Figure 4.1 as 1A. The experimental setup was described in Section 3.2.2. The simulation environment is shown again in Figure 4.5; it consists of a learner robot following behind a demonstrator, who is executing a hand-crafted wall-following task. To remind the reader, a run of the learning phase in the simulation consists

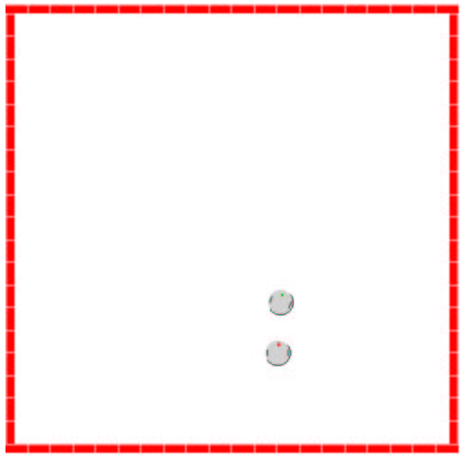


Figure 4.5: The simulated environment used in the wall-following experiments.

of 50000 steps with an interrupt every 5000 step, which forces the demonstrator to move away from the wall and hence wander in the environment. The interrupt is used to ensure that the demonstrator exposes the learner to the full complexity of the task, rather than follow the wall only on one side for the duration of the run.

Recall that due to the imprecise teacher-following behaviour, the learner can sometimes lose the demonstrator. In such cases the learner can either ignore its perceptions or not. The latter corresponds to the 'alone & social' category in Figure 4.1 because the robot learns when it is on its own as well as when it follows the demonstrator. The former corresponds to a 'social facilitation' scenario (see Section 3.1) where the learner only considers its experiences for learning when it follows the teacher — this corresponds to the 'passive demonstrations' category in Figure 4.1. Active demonstrations are not tested in the simulated experiments, as explained in Section 3.1. Note that Section 3.1 discussed two sources of imprecision in the exposure to sensorimotor data arising from the imprecise teacher-following behaviour, and that the social facilitation scenario tests the second of them, concerning the learner losing the teacher. The first source of imprecision, concerning the learner not following exactly the same path as the teacher, is *not* tested here. In Chapter 3 a perfect teacher-following behaviour was emulated by controlling the robot with a hand-crafted wall-following behaviour — the same one that controls the demonstrator when it demonstrates the task to the learner. It is expected that such a behaviour will produce better learning here, but it is not as interesting as comparing the learning when the robot loses the teacher.

We have already seen the characteristics of the perceptual part of the wall-following task in the previous chapter (see particularly Section 3.2.2). The perceptual data we saw there were generated by exactly the same procedure that generated the data used as input to the attention and learning systems in the experiments reported here. In the implementation of the attention system here, the novelty threshold is set to 0.8, the full-habitation parameter is the focus in this chapter and is therefore the independent variable, and the remaining parameters are as reported in the previous chapter (see Table 3.2). The novelty threshold is *not* a parameter in the experiments here because the shape of the SOFM is not crucial to the learning system, as described at the start of this chapter.

We will start the investigation of the learning setup described in Section 4.1 by inspecting the relationship between the attention system and the learning system. In particular, we will explore the extreme effects that a very sensitive and a very insensitive attention system has on triggering the learning of raw data. This exploratory investigation will highlight the importance of the designer effort in setting the saliency parameter of the attention system, which will be shown further, together with the importance of the social interactions, through the quantitative results that will follow. In the exploratory investigation experiences are ignored when the learner loses the demonstrator.

Exploratory Investigation of Attention-Triggered Learning

Since the role of attention is to modulate the patterns presented to the learning system, let us observe the nature of these patterns. Recall that learning is triggered when the SOFM is attentive, *i.e.* when the winning node is not fully-habituated, which occurs when a node is added due to novelty or when forgetting occurs, and continues until the node habituates below the full-habituation threshold. The most simplified case is when fully-habituated nodes are not allowed to dishabituate — by using a maximal full-habituation time (*i.e.* the length of the run). In this case the learning network is only exposed to stimulus ‘events’ once, where an ‘event’ refers to a sequence of similar stimuli, detected (and represented) by a cluster of nodes in the SOFM.

This situation is depicted in Figure 4.6, which shows the patterns presented to the learning system through attention-triggering. It is important to note that the horizontal axes in Figure 4.6 do not correspond to real time, but rather to when the learning system is triggered. The time between consecutive steps in this plot can correspond to consecutive time steps in the simulation, or to time steps very far apart in the simulation. In Figure 4.6, approximately 2400 patterns are presented to the learning system, out of a possible 50000 — the length of the learning phase.

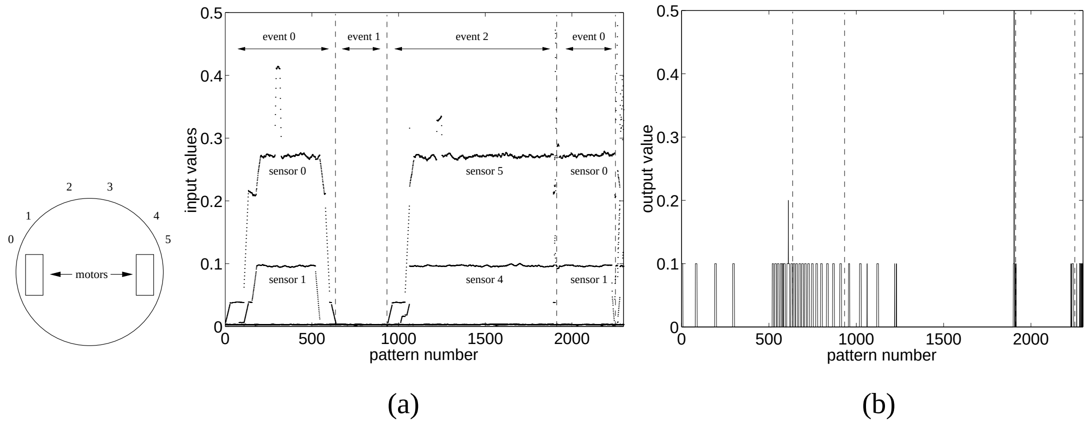


Figure 4.6: The patterns passed to the feed-forward network through attention-triggering. (a) The input units take on the sensor values, see diagram of the robot on the left. (b) The target value used in the output unit is calculated from the motor values, and represents a tendency to turn. Emergent perceptual events are marked by vertical lines. Note that the horizontal axes do not correspond to real time, but rather to when the learning system is triggered. Approximately 2400 patterns are presented in this example, out of a possible 50000 — the length of the learning phase.

Firstly, the left plot shows the values of the input units coming from the learner's six sensors as it is led through the environment. To the left of that plot a diagram of the robot and the location of its sensors is provided. The plot has been labelled and vertical lines have been superimposed on it to mark the emergence of perceptual events, as follows:

- initially the learner is exposed to the wall on the left (event 0); the learning system receives patterns until the SOFM nodes in the cluster for this event — see the SOFM in the top-right of Figure 3.33 on page 96 — have fully-habituated to the input;

- the next time the learning system receives patterns is when a novel event is detected (event 1), which corresponds to not sensing the wall anywhere (recall from Section 3.2.2 that an interrupt forces the demonstrator to turn away from the wall at regular intervals and wander in the middle of the environment); again, the system receives patterns for this event while the corresponding SOFM nodes are habituating;

- another novel event next triggers the learning system (event 2) — the wall being sensed on the right;

- in fact it appears that the SOFM nodes corresponding to event 0 did *not* fully-habituate, because the learning system is triggered by this event again (recall that nodes are not allowed to dishabituate in this example); event 1 must have been detected before the nodes for event 0 were able to fully-habituate;

- once the nodes for event 0 are fully-habituated, the learning system is not triggered again, as all the nodes are now fully habituated, and no novel stimuli are encountered.

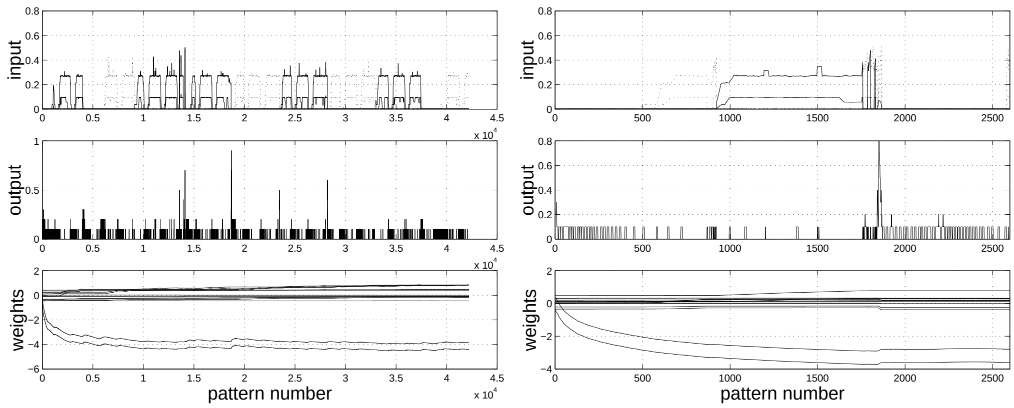
Notice that the front two sensors (sensors 2 and 3) are never active; this is because the learner never perceives the wall in front, as it is always following behind the demonstrator.

The right of Figure 4.6 shows the targets for the output unit that are used to train the feed-forward network. The event markers from the left plot are used here to show how motoric events coincide with perceptual ones. Frequent activations correspond to a higher tendency to turn (hence 'wandering'), which coincide mainly with perceptual event 1, *i.e.* the no-wall event; low frequencies correspond to low tendency to turn, or high tendency to move straight forward, which coincide with the other two events, *i.e.* when the wall is sensed on either side (hence 'wall-following'). It would appear therefore that the neural network should be able to correctly learn the perceptual-motoric mappings for this task in this example.

It is important to highlight at this point an important aspect of the way modulation occurs: it ensures an equal exposure to the different parts (events) of the task. In the example shown in Figure 4.6, the learning is exposed to approximately 2400 patterns; if there were no modulation and instead the system was exposed to the first 2400 patterns, they would not be representative of all the events. The importance of this issue will be stressed again in the discussion of limited resources, in Section 4.4.

Figure 4.6 clearly shows the quantitative benefit of attention: the attention system has reduced the exposure of the learning system to three 'events', using approximately 2400 learning patterns (out of 50000 — the length of a single run). If the learning network can form the correct perception-action mapping, this is also of great qualitative value. Unfortunately, this is not always the case: a single presentation of events is not always sufficient for weight convergence. Also, the learner can be 'unlucky' if at the particular times that the attention system triggers learning, the demonstrator is doing something distracting from the task (such as turning away from the wall because of the interrupt, or even turning away because the learner is in the way!) These kinds of situations would be minimised if the teacher were demonstrating the task actively.

However, if we use the forgetting mechanism by letting nodes dishabituate as described in



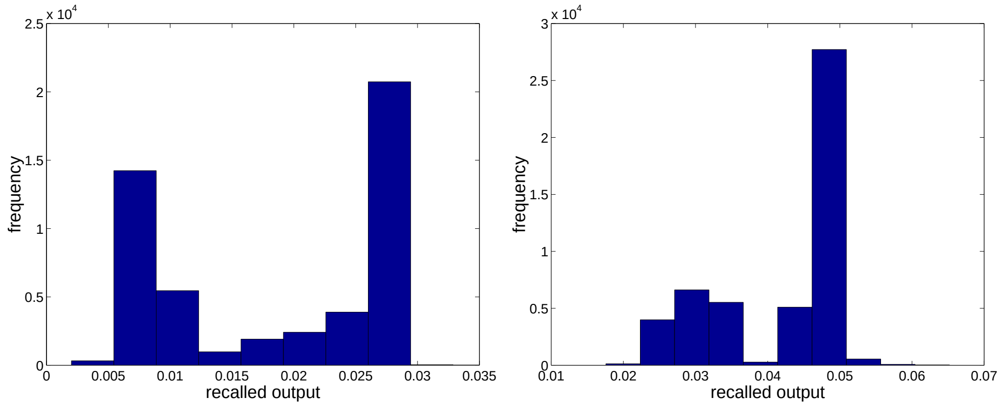
(a) (b)

Figure 4.7: Convergence of the weights of (a) the fully exposed network and (b) the highly selective network, at the learning phase, as input-output patterns are presented through attention-triggering. Perceptual events are distinguished by different line types on the top part of each plot.

Chapter 3, the learning system can be re-exposed to events. The actual number of times that the learning system is exposed to the same event is governed by how long we let a node in the SOFM stay fully habituated before we dishabituate it. As we saw with the toy example in Section 3.3, if we make this length 0, than this is equivalent to no modulation at all, since the attention system is always attentive. In contrast, if we make the full-habituation length very high, the network is exposed to very little information, as we have seen in Figure 4.6, and cannot adapt and recover from premature habituation.

Let us compare the effect of these extremes on the feed-forward network. Figure 4.7 shows the convergence of the network weights due to attention-triggering with full-habituation times 0 and 40000 (henceforth referred to as the ‘fully exposed’ and ‘highly selective’ networks, respectively). Perceptual (input) and motoric (output) events are shown (as in Figure 4.6) together with the weights: 12 input-to-hidden weights (6×2), and 2 hidden-to-output weights (2×1). Note that as before, the horizontal axes in Figure 4.7 do not correspond to real time, but rather to when the learning system is triggered.

Although the highly selective network receives very few patterns, these patterns form a range of experiences as representative as those received by the fully exposed network, where in the latter there is a lot of repetition. The weights of the highly selective network are therefore able to converge, almost to the converged weight values of the fully exposed network (recall



(a) (b)

Figure 4.8: Distributions of the output recalled by (a) the fully exposed network and (b) the highly selective network, after learning is complete; the recall is triggered by input from a wandering behaviour, without attention, such that the full variety of values learned by the network is demonstrated.

that the learning rate is fixed, and therefore the same for both networks). In fact, less selective networks would reach that convergence point because they would be exposed to more patterns. Note that the fully exposed network converges quite early, suggesting that so much repetition is unnecessary.

Let us next examine what these networks have actually learned as a result of being exposed to patterns in the learning phase, at their respective levels of selectivity. To do this we let the robot wander around on its own in the environment randomly (using obstacle-avoidance and wandering behaviours) so that it picks up random perceptions, which are fed to the input layer of the network, and output the variety of different values learned through the weights. Figure 4.8 shows the distributions of the output recalled by the fully exposed and highly selective networks.

The two peaks in the output recalled by the fully exposed network show that this network learned to distinguish between the two types of perceptual experience: one requiring very low turn-tendency (values close to 0), corresponding to moving parallel to a wall, and the other requiring a higher turn-tendency, corresponding to moving randomly when not near a wall. We also see two peaks in the output recalled by the highly selective network, which suggests that it too has learned to distinguish the experiences, but not as distinctly as the fully exposed network. Further, upon inspection of the absolute values of the output we see that the values recalled by the highly selective network are higher than those recalled by the fully exposed network, and the left peak is perhaps not close enough to 0. The reason for these differences is that their weights have not quite converged to the same final values at the end of the learning phase.

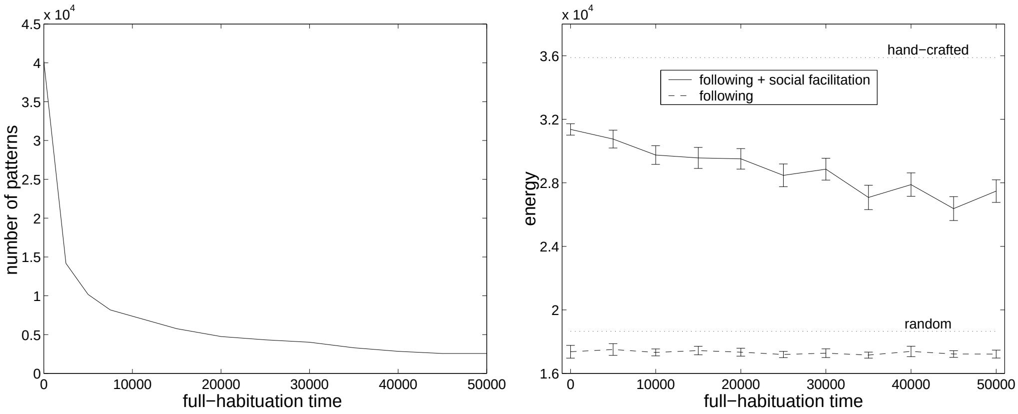
To summarise, a fully exposed network converges quite early, suggesting that exposure to so many patterns is not needed; further, such a network obviously does not make any use of the attention system. On the other extreme, we see from a very selective network that single presentations are perhaps not quite enough, although the network learns reasonably well if the robot does not encounter any 'unlucky' situations. We will next examine the effect of habituation systematically, using values of the full-habituation time in the range between the two extremes used so far.

Results

As mentioned in Section 3.3.3, the full-habituation time does not have a significant effect on the shape of the SOFM, *i.e.* on the perceptual representation. However, we have seen from the two examples above that this parameter is important for learning the task, in the kind of learning setup used in this chapter. We saw from the toy example in Section 3.3 the overall effect of habituation in reducing the amount of information attended to by inhibiting a response to familiar stimuli (see Figure 3.32). In the wall-following task the role of habituation is similar: Figure 4.9(a) shows the number of patterns presented to the network through attention-triggering as a function of full-habituation time; each point on the plot is an average of 100 runs (the size of error bars is negligible and therefore omitted from the plot). As expected, when there is no modulation, the learning is exposed almost all the time (the only times it is not are when the learner loses the demonstrator). Modulation provides a substantial reduction in exposure, and the less forgetting we allow (*i.e.* longer time before dishabituation), the more reduction we get.

To see the effect of habituation on what the network actually learns, we need to devise a measure with which to test the performance of the system across the range of full-habituation times. Two examples of what the recalled output looks like were presented in Figure 4.8, but how does this translate to the ability of the robot to reproduce the wall-following task? To test this we need a way of first translating the output from the learning system into motor commands (see discussion about motor abstraction in Section 2.4.2), and then evaluating the resulting behaviour.

Here there are two abstract motor states: 'move-forward' and 'wander'; the abstraction



(a) (b)

Figure 4.9: Quantitative results from the simulation experiment. (a) The number of patterns presented to the feed-forward network through attention-triggering; (b) 'Energies' acquired at the recall phase by a robot that learned with and without social facilitation in the learning phase; also shown are energies acquired by a robot executing a hand-crafted wall-following behaviour, and a hand-crafted random wandering behaviour. Each point is an average of 100 runs (the error bars in (a) are of negligible lengths and therefore not shown); the length of a single run is 50000.

occurs as follows. Before the recall phase, a calibration dataset is acquired by placing the robot in the middle of the arena for a short length of time, using its perceptions to trigger the feed-forward network, and collecting the output; then, during the recall phase the network output values are statistically compared to the calibration dataset, using the following statistic:

$$t = \frac{y - \bar{y}}{s_c}, \tag{4.2}$$

where y is the current network output, and $\bar{y}$ and s_c are the mean and standard deviation of the calibration dataset, respectively. For 'correctly' trained networks it is expected that the output values would differ significantly when the robot is parallel to a wall, as opposed to when it is not. Setting this measure of significance is where the abstraction is expressed — it determines what constitutes a salient difference in the space of possible values that the network can output.

The actual threshold used as a measure of this significance, placed on the value computed in Equation 4.2, is -150. There are two points to note about this value. Firstly, this value is very high in a statistical testing sense, and in fact corresponds to 150 times the standard deviation of the calibration set; usually in statistical testing, values such as 1.64 and 2.33 are used, corresponding to 5% and 1% significance level t-tests. It was found empirically that such a high significance is required to distinguish the two states, because when the calibration set is collected there is very little stimulation in the robot's sensors, and subsequently any small stimulation, in any of the sensors, produces a significant difference — significant in the usual significance testing sense, but not significant for this task. The typical significance values detect differences at a level of granularity which is too fine. The second point to note is that the threshold is negative; this ensures that output obtained for the 'wall' state must be *lower* than the calibration set. That is, the feed-forward network should learn to distinguish between the two states *correctly*, not simply distinguish between them.

The learner is placed in the environment on its own, its perceptual input is passed to the feed-forward network (no attention is used), where an output is computed and then translated to an action as described above. Further, a built-in 'obstacle-avoidance' is used to turn the robot when it is facing a wall, and prevent it from hitting obstacles. The reason for this is that the network was only trained to control the robot when there is a wall parallel to it on its side; any other competencies would require a higher representational complexity, *i.e.* more output units; further, the robot is never exposed to the wall straight in front of it in the learning phase, because it is always following behind the demonstrator. To account for unsuccessful learning the evaluation is penalised whenever the obstacle-avoidance is triggered. As in the learning phase, an interrupt (see Section 3.2.2) is used in the recall phase to avoid the robot following the wall on one side for the duration of the run, and thus not testing the learned behaviour fully; the interrupt is triggered every 7500 steps.

An 'energy' measure is calculated as the accumulation of the robot's side sensors sensing the wall, where different sensor configurations give different energies. This is used as a measure of the robot's ability to perform the task (higher energies correspond to better wall-following[3]). Figure 4.9(b) shows the different energies acquired with a range of full-habituation times, with and without social facilitation. Superimposed on the plot are two baselines corresponding to energies obtained with the robot executing a hand-crafted wall-following behaviour, and a hand-crafted random wandering behaviour.

The first thing to note is that without social facilitation, the performance is consistently poor. In fact the robot never acquires enough energy to counter penalties, and therefore the

[3]The term 'energy' was originally used for the numerical evaluation devised for the phototaxis task presented in the next chapter, where the robot can be thought of as acquiring energy from the light; it does not actually acquire any such energy, however this term is useful for referring to performance. Similarly, in the wall-following task the robot does not gain any energy from the wall — the choice to use the term for this task as well is made simply to avoid using a new term.

performance is significantly worse than the random behaviour. The learning system is therefore exposed to too many undesirable, non task-related patterns in the learning phase, from which it cannot recover regardless of how much forgetting is allowed. Therefore in Figure 4.1 the ‘passive demonstrations’ category is marked as having a better performance than the ‘alone & social’ category.

The second thing to note from Figure 4.9(b) is that the energy calculations confirm what we have already seen through the two extreme examples in Figure 4.8: networks that are exposed to more data (low full-habituation times) can discriminate better between the ‘wall’ and ‘no wall’ states. Note however that even the selective networks (high full-habituation times) perform well, although there is a significant decrease in performance as the full-habituation time increases. The full benefit of the attention system for modulating learning can be seen by considering Figure 4.9(a) and (b) together, that is, by considering both the qualitative and quantitative benefits of attention: while there is a slight decrease in performance when some modulation is introduced, the amount of data considered is *drastically* reduced. Thus a similar performance is achieved with much less work.

To see this more clearly, Figure 4.10 shows the energies achieved ‘per pattern’, that is, the energy obtained in the recall phase divided by the number of patterns considered in the learning phase, for the social facilitation scenario. The steepest part of the slope in Figure 4.10 can be seen in the first increase of full-habituation time, that is, when some modulation is introduced: as stated above, while a slightly lower energy is achieved, it is achieved by considering much less data (see Figures 4.9 and 4.10 with full-habituation time $0 \rightarrow 5000$). Because the energy and number of patterns level out in Figure 4.9 as the full-habituation time increases, we also expect the energy per pattern to level out, as can be seen in Figure 4.10.

To summarise the simulation results, one can achieve a substantial reduction in the exposure of the learning system through modulation, *and* this does not cause a large decrease in the performance, provided that forgetting is allowed reasonably frequently. In these cases the networks are re-exposed to events just enough times to ensure that the weights converge to a ‘desired’ point, and hence recall the ‘desired’ output and the performance is almost as good as a hand-crafted behaviour. When forgetting is less frequent, the performance drops, but is still much better than a random behaviour. The next section will discuss the quantitative role of attention further, by considering situations where modulation is crucial because there is a limit on the number of patterns that can be considered for learning.

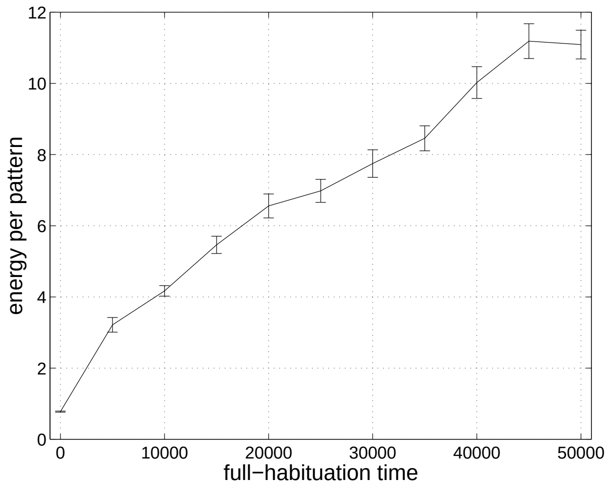


Figure 4.10: Energy per pattern. The energies obtained in the recall phase, shown in Figure 4.9(b) for the social facilitation scenario, are divided by the number of patterns considered in the learning phase, shown in Figure 4.9(a). This plot shows the combined quantitative and qualitative benefits of attention as modulation for learning.

4.4 Modulation for Limited Resources

As mentioned in Chapter 2, the benefit of attention is not only in detecting saliency, but also in dealing with limited resources. In our simplified learning setup here, there is no such limit, because as we can see in Figure 4.9(b), the robot can learn without *any* attention (*i.e.* full-habituation time of 0), and in fact the best generalisation is achieved by the learning system when *all* the experiences are considered. However, in order to be able to generalise these results, one must consider that learning from all the possible experiences might not be possible if one has a limited set of processing resources, which perhaps need to be allocated to activities other than learning (such as tracking the demonstrator, monitoring energy and safety levels, performing other cognitive tasks such as memory consolidation and planning).

We simulate such a scenario by artificially placing a limit on the number of patterns that the learning system can process. Figure 4.11 shows the energies achieved by the robot in the recall phase after learning with social facilitation and different limits. These results show that attention provides an intelligent modulation for learning, by using the available resources to expose the learning equally to the salient events. As the learning system is limited to process fewer patterns, the overall performance decreases, but the best performance is achieved through

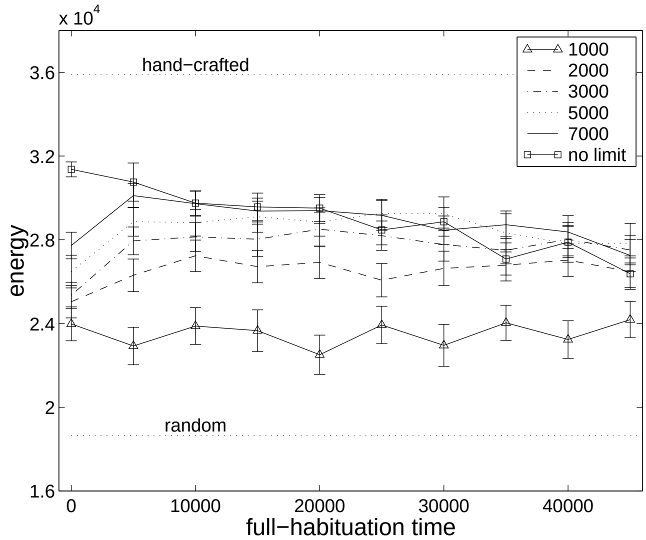


Figure 4.11: Performance in the the recall phase resulting from learning with social facilitation, and with different limits on the number of patterns processed. The legend shows how many patterns the learning systems was allowed to process. Each point is an average of 100 runs; the length of a single run is 50000.

more modulation. When the limit gets too low (1000 patterns), however, modulation does not have a significant effect; this is because even without any modulation, 1000 patterns simply do not provide enough exposure to all the events of the task; we can see from Figure 4.6 that 1000 patterns correspond to at most two of the three events of the task.

4.5 Implementation on the Physical Wall-Following Task

The learning setup described in Section 4.1 has also been implemented in the physical experiments involving the robot Gillespie and a human demonstrator, presented in Chapter 3. This section presents the set of experiments depicted in Figure 4.1 as 1B. The experimental setup was described in Section 3.2.2. The physical environment is shown again in Figure 4.12; it consists of a learner robot following a human demonstrator, who is performing a wall-following task. The data explorations performed in Section 3.2.2 suggested that it might be difficult to learn this task, because the data are poorly structured. So prior to implementing the feed-forward network it was believed that an additional social input would be useful and perhaps necessary: the demonstrator signals to the robot at very specific times during the task; this is

Figure 4.12: The physical wall-following experimental setup, where the demonstrator can explicitly signal to the robot.

achieved with a red glove, which is easily detectable as a contrast to the green shirt used by the robot's teacher-following behaviour.

This signalling by the demonstrator is used to highlight certain experiences over others, and the signal is used directly by the attention system: it forces dishabituation of the winning SOFM node. Therefore, this signalling by the demonstrator in effect takes over the role of the full-habituation parameter. Note that the demonstrator does not just signal a single desirable stimulus, but a desirable *event*, because attention is given to the environment while the node habituates again, as we saw in the previous section. In practice the demonstrator signals when the robot is 'nicely' positioned parallel to the wall and moving forward, and when the robot is in the middle of the arena and the demonstrator is demonstrating frequent turns: the demonstrator is drawing the robot's attention to very specific locations in the environment, at very specific times during the task. These timings were quite crucial, and so care was taken by the demonstrator during the experiments. The aim of this experiment is to evaluate this care by the expert compared to the care taken by the designer in setting the full-habituation saliency parameter.

Recall from Section 3.1 that in the physical experiments passive demonstrations are not practical with the current implementation of the robot's human-tracking system. Therefore, although some preliminary runs were made with passive demonstrations, where the human had his back to Gillespie and was moving without adapting the movements to Gillespie's behaviour, the impracticality was quickly evident by Gillespie frequently failing to track the human, and so there are no numerical results from these preliminary runs (see 'passive demonstrations' category in Figure 4.1). Further, by turning to face Gillespie and demonstrating the task actively, the human ensured not to fall out of Gillespie's field of vision for significant periods, and therefore there is no 'alone & social' scenario as in the simulation experiment.

Active demonstrations are tested firstly by emulating a perfect, noise-free, teacher-following behaviour, by controlling the learner with a hand-crafted wall-following behaviour, thus addressing imprecise exposure to experiences due to an imprecise teacher-following behaviour (see Section 3.1). However, in Section 2.1.2 active demonstrations were proposed not only for ensuring precise exposure to experiences, but also for accentuating the differences between the parts of the task by deviating from the natural demonstrations to 'exaggerate' these differences. Therefore, active demonstrations are tested secondly by providing exactly this kind of demonstration for Gillespie in the experiment here: the demonstrator 'exaggerates' the turning component of the behaviour, *i.e.* purposely performs large turns in both directions when in the middle of the arena, significantly more than the 'wander' component of the hand-crafted wall-following behaviour, and spends more time in the middle of the arena, as we saw in the data inspections in Chapter 3 (see Figure 3.14 in particular). The benefit of explicit signalling is tested by either ignoring the expert's signals during the demonstrations, or not. The experiment will also compare the results from training the robot as it moves around randomly in the environment on its own.

We already saw in the previous chapter (see particularly Section 3.2.2) the characteristics of the perceptual part of the physical wall-following task. In fact, the perceptual datasets seen in the previous chapter (see Figure 3.14) are used as off-board training sets in the experiments reported here. Due to practical constraints arising from experimenting with a real robot, the learning phase is always based on the same run, consisting of 10000 steps corresponding to approximately 40 minutes of real time, with approximately 12 interrupts (the interrupts serve the same purpose as in the simulation experiment: the demonstrator turns towards the middle of the arena to avoid following the wall on one side for the duration of the run). All the information from this run (the sensor data, motor data, and red-glove-detection data) is saved and used as an off-board training set. Stochasticity in the repeated experiments discussed below is achieved through a random initialisation of the feed-forward network weights, a random initialisation of the SOFM nodes, and a random starting point for accessing the datasets.

The learning setup is exactly as in the simulation experiment, and the learning rates used to update the weights of the feed-forward network are shown in Table 4.1. In the attention system the novelty threshold is set to 0.25, the full-habitation parameter is the focus in this chapter and

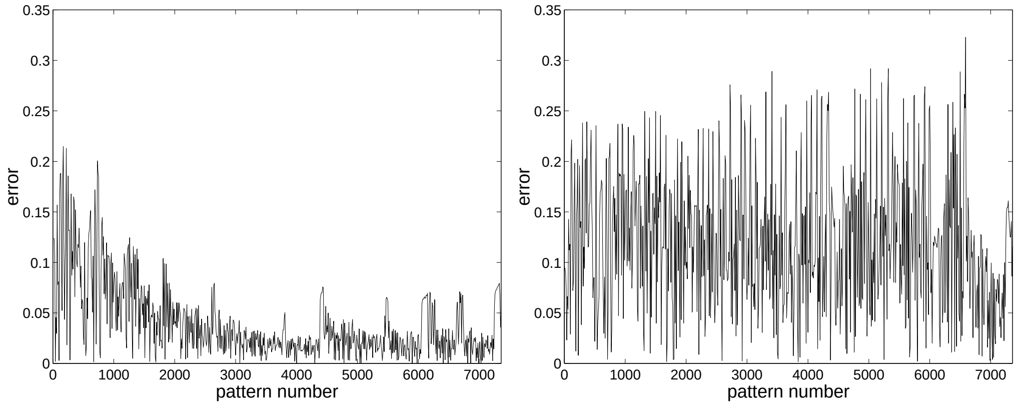


Figure 4.13: Two examples from the physical experiment, demonstrating that the feed-forward network can behave rather differently under different weight initialisations, even though the same training set is used. The plots show the error in the output computed by the networks during the learning phase.

is therefore the independent variable, and the remaining parameters are as reported in the previous chapter (see Table 3.2). Before inspecting the quantitative results from this experiment, we will first investigate the relationship between the attention system and the learning system, as we have done in the simulation experiment.

Exploratory Investigation of Attention-Triggered Learning

Perceptual and motoric events can be identified as in the simulation (Figures 4.6 and 4.7), however it is hard to graphically show the activation of 20 sensors; similarly, it is hard to show the values of 42 weights (20×2 input-to-hidden weights, and 2×1 hidden-to-output weights). Observations of single runs show that the learning is not very stable: the network behaves in different ways for different weight initialisations even though the same training set is used. This is demonstrated in Figure 4.13, which shows the error in the output computed by two networks trained under similar conditions, during the learning phase. A possible explanation as to why the behaviour of the network is inconsistent is that the error space is a complex manifold with many local minima, arising from the high dimensionality and noisy nature of the input space. As was suggested in the investigations of the raw perceptual data in Chapter 3, the data from the physical robot are poorly-structured and difficult to learn from, even in this simplified learning setup.

As before, let us also inspect what the network learns to output. In the simulation this was

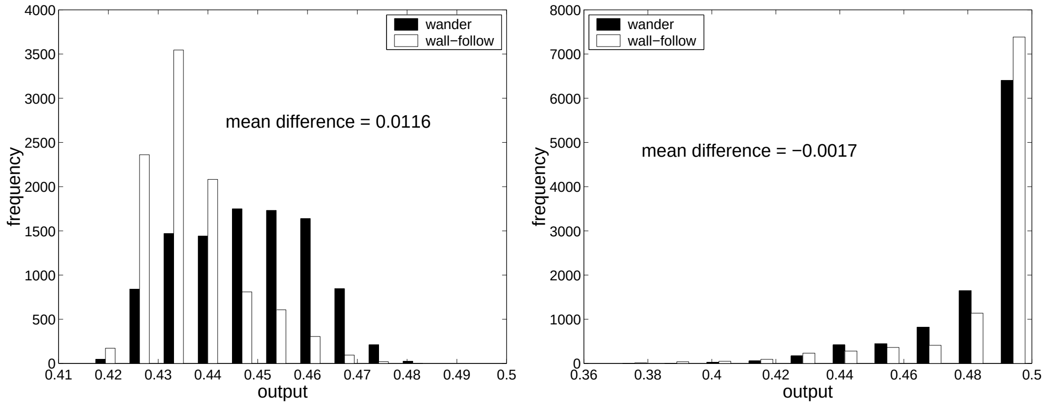


Figure 4.14: Distributions of the output recalled by two different networks, where the input is an off-line testing set of perceptions obtained from wandering and hand-crafted wall-following behaviours, shown separately. The network on the left has learned to separate the two perceptual states better than the one on the right.

achieved using a random wandering behaviour, which provided a representative set of perceptual inputs to the network and was therefore able to trigger all the different output values (see Figure 4.8). The physical robot's wandering behaviour is exposed to the wall much less than in the simulated experiment, so by feeding the feed-forward network input from a wander behaviour alone we would not see two peaks as in Figure 4.8. Therefore in addition to inspecting the output from a wander behaviour, the output resulting from a hand-crafted wall-following behaviour is also inspected.

If the network has learned correctly we expect the output distributions arising from the two behaviours to be different, with lower values being recalled for the wall-following behaviour as it involves the robot being parallel to a wall most of the time. Note that the two behaviours share some perceptions, and so an overlap in their output distribution is expected. Figure 4.14 shows the output distributions separately for the wander behaviour and wall-following behaviour, recalled by two different networks trained under similar conditions; also shown is the difference between the mean of the output recalled for the wander behaviour and the mean of the output recalled for the wall-following behaviour.

The network on the left of Figure 4.14 seems to have learned more correctly than the network on the right. The former was able to distinguish, to some extent, the output for the two different behaviours; the separation is not large, which is due to the common perceptions shared by the two behaviours, and we will see later how this translates to the ability of such a network to reproduce the task. In contrast, the network on the right does not seem to have distinguished the two behaviours at all, and in fact the most frequent output is 0.5, which is the natural 'zero' of the network[4]; a zero output means that the output neuron is neither excited nor inhibited, but it should on average be inhibited, because there are more data for 'wall' than 'no wall', and therefore there are more forward moves than turns[5]. In other words, the second network did not learn anything useful.

Results: Off-line Testing

The examples shown in Figures 4.13 and 4.14 demonstrate that networks trained under similar conditions can behave very differently. As mentioned above, the learning in the physical experiment is not very stable, so any one of the above examples is not representative. The behaviour of the network will now be analysed more generally, by testing the network with different full-habituation times, and with and without the signals from the demonstrator, where each particular combination is repeated 100 times. The variable under inspection is the mean difference between the output distribution due to the 'wander' perceptual input and the output distribution due to the 'wall-following' perceptual input, as demonstrated in Figure 4.14. Since the distributions should ideally be as separable as possible, a high mean difference corresponds to useful learning. The result are shown in Figure 4.15.

For comparison, the network is also *trained* on input from the wander dataset, and on input from the hand-crafted wall-following dataset. The former provides a useful baseline because the robot takes random turns no matter what it perceives, and so the network is not expected to learn to distinguish between the two sensorimotor states. One can then judge the significance of the ability of the network to distinguish between the two states when trained under different conditions, compared to when it is not expected to make this distinction, as we will see below. The latter provides a useful baseline when we recall from Chapter 3 that the hand-crafted wall-following behaviour emulates an ideal teacher-following scenario, as mentioned above, and therefore another form of active demonstrations is evaluated.

The best results are achieved when the signals are used (hence the 'explicit signalling' category in Figure 4.1 is marked as having the best performance), and there is a significant improvement as full-habituation time increases. When the full-habituation time is zero (defined

[4] An output of 0.5 from the sigmoid function corresponds to an input of zero, *i.e.* the output unit receives zero activation (see footnote on page 111: $f(0) = 0.5$)

[5] Compare these output values to the simulation experiment (Figure 4.8), where the output values are much closer to 0 than to 0.5; in the simulation the ratio of forward to turn moves is very high (approximately 20:1), so overall the network learns to inhibit the output neuron (low turn tendency); in Gillespie this ratio is much smaller.

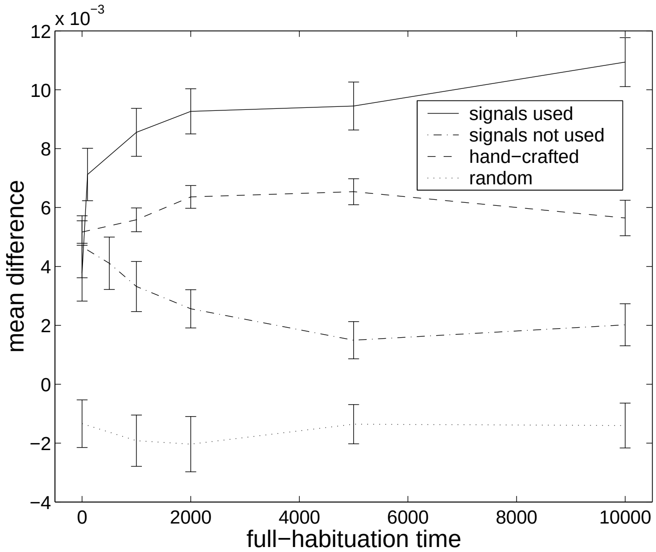


Figure 4.15: Mean difference between output recalled off-line for inputs from a random behaviour and from a wall-following behaviour, repeated 100 times for each full-habituation value. Curves are shown for networks trained through demonstrations where signals are used or not used, and for networks trained through a hand-crafted wall-following behaviour and a random behaviour.

as 'full exposure' in Section 4.3) the signals are ineffective because attention-triggering occurs at every step (*i.e.* there is no modulation); however as soon as some modulation is introduced (full-habituation time = 100) there is a significant improvement: the signals are starting to have an effect. As the full-habituation time increases further, the signals start to dominate over the forgetting mechanism in the attention system, and the absolute best performance is achieved when the full-habituation time is maximal (*i.e.* the length of the run = 10000 steps), which means that dishabituation occurs *only* due to the signals.

The best results outperform those achieved when the network is trained on the hand-crafted wall-following dataset, that is, they outperform the ideal teacher-following scenario. Much of this is due to the signals which help to structure the sensorimotor data, especially when the full-habituation time is high. However, the results are also better when the signals are not so dominant, when the full-habituation time is low, as explained above. Part of the responsibility in providing better-structured data for learning is the active demonstrations from the human demonstrator. As mentioned earlier, the exaggerated demonstrations provide significantly more experiences in the middle of the arena, with significantly larger turns, compared to the robot's experiences when it is controlled with the hand-crafted wall-following behaviour. The network is therefore better able to distinguish between the salient parts of the task — moving forward when parallel to a wall and turning otherwise.

When the signals are not used, we see a situation similar to Figure 4.9(b), namely that the network learns best when all the data are considered, and that when some modulation is used (full-habituation time = 500), the performance is only slightly poorer; if there is too much modulation (full-habituation time $\geq$ 1000), the network is not exposed to enough experiences, and the performance starts to resemble the performance when the network is trained on the random behaviour. Notice that the results from the random wander baseline are significantly negative, whereas one might expect them to not be significantly different from zero. In fact, the sign of the mean difference of this random scenario is not as important as its absolute value[6]. We should thus consider this absolute value as the baseline with which to judge the significance of the other results, and this explains the results reported next, relating to the robot's ability to convert the output from the network to motor commands for executing the task.

Results: Testing the Learned Behaviour

We have seen the values learned by the network, and under what conditions it learns to distinguish well between the parts of the task. Let us now examine the robot's ability to translate this learned output to behaviour, by using a calibration baseline dataset, as in the simulation, and choosing between the two actions using Equation 4.2, and a threshold of -2.9. Note that this threshold is much smaller in absolute value than the one used in the simulation experiment (-150). This is due to the noisy nature of the data: one does not obtain such a clear distinction between the two perceptual states, as we saw in Figure 4.14, and must accept that with such a low value a significant difference is detected spuriously.

As in the simulation, an energy measure is calculated as the accumulation of the robot's side sensors sensing the wall, while the robot is moving around the environment on its own with the aid of a built-in obstacle avoidance behaviour, recalling an output value for *each* perception (*i.e.* attention is not used) and acting on it. The energy is penalised whenever the obstacle-avoidance behaviour is triggered. A recall run consists of 6000 steps, with an interrupt signalled every 1000 steps. The testing of the recalled behaviour is only tested on a select number of the conditions, again due to practical time limitations. Four representative examples are tested:

[6]The negative sign does not necessarily mean that the network learns to turn more when the robot is parallel to the wall, because in fact the robot is hardly exposed to such experiences with the wandering behaviour. A more reasonable explanation to this negative sign could be that there is a stronger activation of the input units when the robot is near a wall — many of its sensors are strongly stimulated — and therefore the output of the network is high.

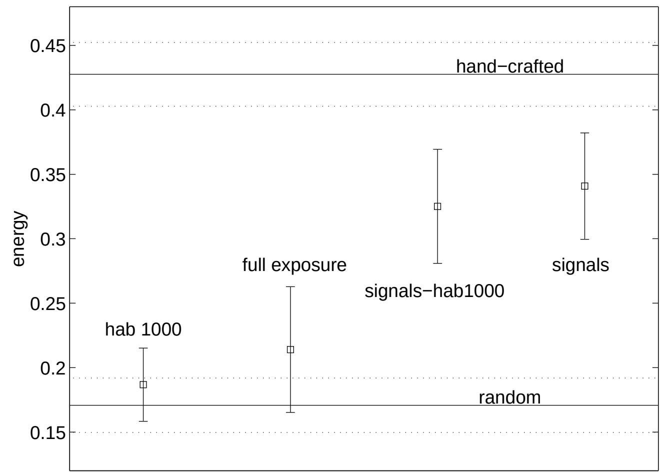


Figure 4.16: Evaluation of the learned behaviour in the physical experiment, where the signals from the demonstrator are used (the two cases on the right), or ignored (the two cases on the left); evaluations of hand-crafted and random behaviours are also shown (the dotted lines show their respective standard errors).

(1) dishabituation due to signals *only* (full-habituation time = 10000); (2) dishabituation due to signals *and* forgetting with full-habituation time 1000; (3) full exposure (full-habituation time = 0); (4) dishabituation due to forgetting *only* (*i.e.* no signals), with full-habituation time 1000. Each of these is repeated 10 times, and the results are shown in Figure 4.16. The energies acquired by the robot executing a hand-crafted wall-following behaviour and a hand-crafted random behaviour are also shown for comparison (note that these are not results obtained from *training* the robot with these hand-crafted behaviours as in Figure 4.15).

The absolute value of these results of course depends on the threshold used to abstract from the raw output values into actions. However, when compared to each other, the results tell us more about how well the network learns under different conditions, in conjunction with the results shown in Figure 4.15. Specifically, we see that when the signals from the demonstrator are used, the conclusion from Figure 4.15 that the network learns well is confirmed here in the robot's execution of the task. The performance is best when *only* the signals are used for dishabituation, but when there is some forgetting, with full-habituation time 1000, the network learns well enough to achieve almost as good a behaviour. When the signals are ignored, the suggestion in Figure 4.15 that the network does not learn as well is confirmed by the fact that the resulting behaviour is not always better than the random behaviour. Further, when the signals are not used, the network does best to consider as much data as possible. We have seen in Section 4.4 how in the simulation experiments the latter might be constrained by the available resources; the same constraint would be expected to apply here if there were limited resources available for learning.

4.6 Conclusion

This chapter has presented experiments where little designer effort is required in the learning setup. The reason that little effort is required is that the learning occurs on the raw data and therefore there is no need to carefully abstract the data. Instead, the effort is expressed in setting the saliency parameter responsible for modulating the learning of raw data, and it was shown that learning is also possible without any modulation at all. It was also shown how setting the parameter might be governed by the available resources, when learning without modulation is not desirable.

The learning system is set up to learn a very simple function, and to achieve the task fully it must be bootstrapped with hand-coded behaviours. The motivation for designing such a simple learning system was that the effect of attention (and hence designer effort) and social interactions could be reliably evaluated, as discussed in Section 4.2. The simple design of the learning system was taken as a first step, and it is believed that in further work the conclusions from this chapter could be tested further by increasing the complexity of the learning system (for example, by adding an output unit to represent direction of turn).

The simulation results in Section 4.3 (summarised as '1A' in Figure 4.1) firstly show that if the robot is allowed to learn when it loses the teacher, the exposure to irrelevant data is significant, and the robot cannot learn the task; it is crucial for the robot only to learn from the demonstrations of the teacher. Secondly, the results show that the learning architecture generalises best when the data are considered all the time, that is, without any modulation. However, the ability to generalise is only slightly reduced when some modulation is introduced, and in contrast, the amount of data considered is drastically reduced. Thus, the quantitative benefit of attention is clear. This benefit is demonstrated further in Section 4.4, where a limit is placed artificially on the number of learning examples that are allowed to be considered, thus emulating a limited-resources scenario. As the limit increases (fewer learning examples are allowed), the overall performance decreases but is maximised through more modulation.

In the physical experiments presented in Section 4.5 (summarised as '1B' in Figure 4.1), we firstly saw the advantage of active demonstrations. We saw their role in addressing imprecise teacher-following by emulating a perfect teacher-following scenario through a hand-crafted wall-following behaviour. We also saw their further role in accentuating the salient differences between the components of the task, with the human demonstrator exaggerating these difference and then signalling their occurrence. It was pointed out in Chapter 3 that the demonstrator actively exposes the learner equally to the parts of the task, but we did not see there the implications for learning the appropriate motor values. Here we saw in Figure 4.15 that due to the nature of the human demonstrations, the learning architecture is better able to separate the sensorimotor regularities of the two salient components of the task. We also saw the benefit of explicit signalling in directly influencing the robot's notion of saliency. The explicit signals from the demonstrator force attention to be (re-)given to the current event; the results show that this signalling has a significant effect — the learning improves as attention is influenced more by the demonstrator, and less by its own parameters. Thus the signalling by the demonstrator takes over the role of the habituation saliency parameter, whose value must be set by the designer. The responsibility of biasing the robot's detection of saliency is transferred from the designer to the demonstrator. However, as long as the signalling is used, attention *can* also be used to select experiences, as long as there is some modulation. This means that the burden on the demonstrator can potentially be reduced, because the learning does have not rely solely on the demonstrator.

Without the explicit signalling the raw data are too poorly-structured for reliable learning of the task, even in this simplified learning setup and with active demonstrations. The fact that learning of the same task is easier in the simulation confirms the comparisons made in Chapter 3, which show that the data from the simulation experiments are much better structured than those from the physical experiments. Active demonstrations and explicit signalling were not attempted in the simulation experiments. The reason for this was discussed in Chapter 2, and is related to the fact that programming such functionalities for a robotic demonstrator is difficult. In the physical experiment the timing of the signals is crucial and depends on the exact position and orientation of the robot. Recall that in the simulation experiment the demonstrator does not actually perceive the learner or know where it is. It is expected however that if active demonstrations, or explicit signals, or both, were present in the simulation, this would significantly improve the performance.

The important conclusion from this chapter is that in both the simulation and physical experiments, the best performance was achieved with the strongest form of social interactions that were attempted, and this is argued to be the case because learning occurs on a low level of abstraction — on the raw data.

Chapter 5

Learning at a High Level of Abstraction

This chapter presents experiments where the learning occurs not on the raw perceptual data, as in the previous chapter, but on the structures discovered and abstracted from the raw data through attention. Therefore, learning occurs at a higher level of abstraction. As argued in Chapters 2 and 3, a lot of care must be taken by the designer in biasing such abstraction usefully for learning, because there is a compression of the raw data and hence a loss of information.

Therefore, as opposed to learning from the raw data, here more effort is required by the designer, and so there is a clear distinction between the experiments here and those in the previous chapter in terms of designer effort. This can clearly be seen in the graphical summary of the experiments, shown again in Figure 5.1 — this chapter involves experiment sets 2–4 while the previous chapter involved experiment set 1. In this chapter the issue of designer effort is analysed further, by identifying an increasing amount of effort between the three sets of experiments (these distinctions are depicted by the vertical placement of experiment sets 2–4 in Figure 5.1). More design effort also means that learning can be more reliable if the designer manages to bias the abstraction usefully and thus present better structured 'data' for learning, as discussed in Chapter 2.

The aim of the experiments in this chapter is to show the implications of these issues regarding designer effort on the effort required from the expert involved in the social interactions. Specifically, they are tailored to addressing the following question: can good performance, achieved through more design effort, be matched by stronger social interactions?

In all three sets of experiments, the designer must bias the abstraction of perceptual data,

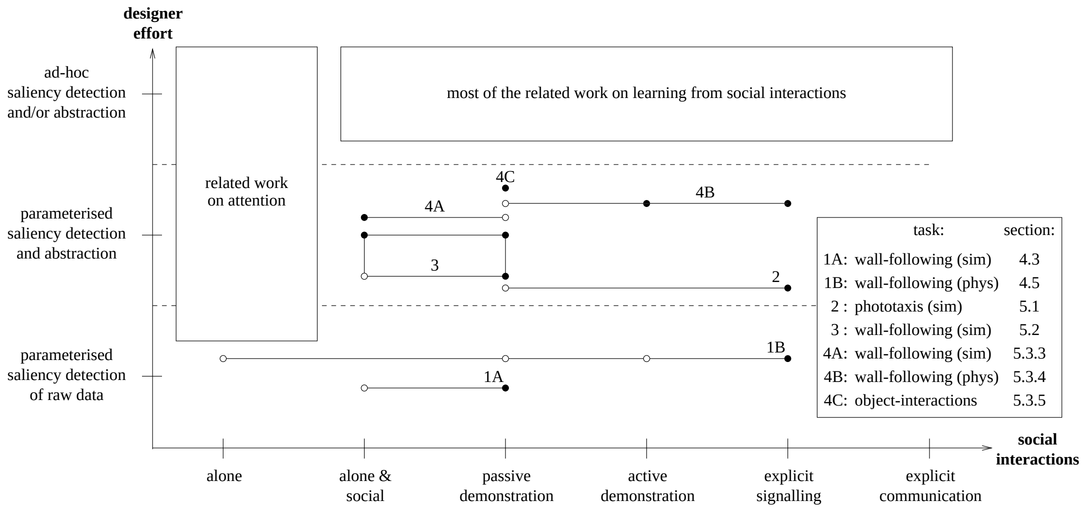


Figure 5.1: This chapter presents experiments 2–4 shown in the figure. They involve learning from perceptual structures discovered and abstracted from the raw data through attention. As opposed to the experiments presented in the previous chapter, where learning occurs on the raw data, here more effort is required from the designer, who must be more careful in biasing this abstraction of the perceptual data usefully for learning, because learning is based on a compression of the raw data. Designer effort is tested further in this chapter through an increasing amount of effort needed for abstracting *motor* data, depicted through the vertical placement of experiments 2–4. The experiments address the implications of different amounts of designer effort on social interactions.

as discussed above. The additional effort that distinguishes between the experiments is related to the effort required in abstracting the *motor* data — see Section 2.4.2 about the need to abstract motor data. In the first set of experiments (Section 5.1), this effort is the least in the experiments in this chapter, and corresponds to setting a saliency parameter, similarly to the abstraction of the perceptual data[1]. The second set of experiments (Section 5.2) also requires setting such a parameter, but it also involves an additional heuristic, which is used to explicitly compare the performance with and without this additional designer effort, and the implications for the social interactions. The third set of experiments (Section 5.3) relies on pre-existing general-purpose sensorimotor skills, where the abstraction of the motor space is incorporated, and it therefore requires the most designer effort. The second and third sets of experiments use the attention system described in Chapter 3 comprising the SOFM, while the first one uses

[1]See start of Section 1.1 on page 3 for the definition of 'saliency'.

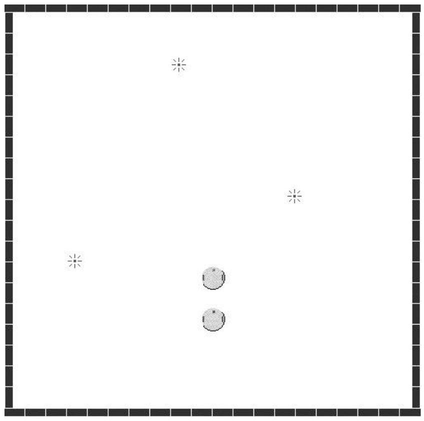


Figure 5.2: The simulated environment used for the phototaxis task contains three light sources.

a different attention system as explained in the next section, corresponding to a perception of change mechanism.

5.1 Learning with a Hopfield Network

The experiments in this section involve the phototaxis task, which was discussed in Section 3.2.2, corresponding to experiment 2 shown in Figure 5.1. The environment used for these experiments is shown again in Figure 5.2. Recall that this is the same platform used in the simulated wall-following experiment in the previous chapter. As in the simulated wall-following experiment, here too active demonstrations are not attempted, due to the difficulty in designing active robotic demonstrators, mentioned in Section 3.1. Further, it was mentioned in Section 3.2.2 that in the phototaxis task the learner never loses the teacher, and so in this experiment there is no 'alone & social' testing scenario as in the previous chapter.

However, there is an additional form of social interactions here — explicit signalling: the demonstrator speeds up just as it starts to turn towards a light source and until it is facing the light source; the learner detects this change in speed through its own motors as it tries to maintain a constant distance from the demonstrator. Thus the demonstrator can potentially draw the attention of the learner to the important changes relevant to the task, and this will be referred to as *stimulus enhancement*, because it resembles the social learning phenomenon mentioned in Section 2.6. Note that a more natural change in speed might be detected without the teacher purposely speeding up — when the teacher turns its speed is momentarily reduced, especially when it makes sharp turns. However, the teacher does not always turn sharply towards the light sources, depending on the angle at which the light first appears; in some cases the turn is rather smooth and therefore does not result in a significant change in speed. Therefore, the explicit change in speed was chosen as a more reliable signal.

It was identified in Chapter 3 that the salient variations in this task occur at a finer level of granularity than a distinction between the ‘light’ and ‘no-light’ clusters (see Figure 3.17). The investigations in Chapter 3 suggest that in order to learn this task, one must represent how to act when the light is perceived by the robot at different configurations with respect to the light. This can be achieved, for example, by utilising a very sensitive novelty threshold in the attention system presented in Chapter 3, as discussed in Section 3.3. However, the experiments reported here were carried out before the attention system presented in Chapter 3 was developed. The choice of the attention mechanism presented here was made specifically to address the representation requirements mentioned above. This choice not only contributes to the overall argument in this thesis, and particularly in this chapter, regarding designer effort, but it in fact strengthens it because it demonstrates that the argument applies to different types of attention and learning mechanisms.

The attention system corresponds to a perception of change mechanism, which looks for a change, separately in each of the robot’s sensors. The learning system associates these changes with relevant actions and thus the robot learns how to act as the light appears in any configuration of its sensors. The representation of the actions used in the learning is achieved by abstracting the motor data in a more ad-hoc manner than the abstraction of the perceptual data using attention, as discussed in Section 2.4.2.

5.1.1 Attention as Perception of Change

The saliency of a stimulus is determined according to how different it is from the last few stimuli, as perceived by each sensor separately. At each time step, the perception of change mechanism performs the following, for each sensor:

1. a short-term memory window of k previous values is created (k=30):

$$\{x(t-k),\ldots,x(t-1)\} \tag{5.1}$$

   where $x(t)$ is the value of the sensor at time t.

2. an average over this window is calculated:

$$m(t) = \frac{1}{k}\sum_{i=1}^{k} x(t-i) \tag{5.2}$$

3. the current sensor value is compared with this average, using the following test statistic:

$$d(t) = \frac{x(t) - m(t)}{s(t)} \tag{5.3}$$

   where $s(t)$ is the standard deviation of the window at time t, that is, the standard deviation of $\{x(t-k), \ldots, x(t-1)\}$. This is equivalent to a t-test, which tests if the current sensor value $x(t)$ is significantly different from a short sample of sensor values taken in the last k steps. Comparing the current value with a *sample* of previous values, rather than simply the previous sensor value, provides some smoothing over noise, and is therefore more robust in detecting the meaningful changes.

4. if the test returns a significant result, signalled when $d(t)$ exceeds a threshold, a change is considered to have been detected by the sensor; the system is said to be attentive overall when a change is detected in any one of the sensors.

The detection threshold plays a very important role in the detection mechanism, and can be thought of as the main saliency parameter. Note that another parameter is the window size, k. However, the respective influences of the two parameters are dependent on each other and so we can fix k and investigate the detection threshold, which is more interesting from the point of view of this thesis because it measures saliency in terms of the *current* experience, as discussed in Section 3.3. The value of k is thus determined empirically.

5.1.2 Learning Setup

The perception of change mechanism described above is used to trigger a learning architecture. The learning architecture is a Hopfield neural network (see, for example, Hertz et al., 1991), which is used to associate the change detected in the sensors with the actions carried out by the motors. Such learning is possible because the phototaxis task only requires the robot to act when its perception is *changing*. For example, when a light appears on the left, the left sensors will detect a change through Equation 5.3, and the correct behaviour is to turn in that direction; or when the light is increasing in intensity straight ahead, the front sensors will detect a change, and the correct behaviour is to move forward. The task does *not* require the robot to stay near a light source when close enough, so the robot does not need a representation of the light being at a particular intensity. Similarly, because the robot is equipped with a default wandering behaviour, this can be triggered whenever the system is not attentive, and hence the robot does not need a representation of not sensing any light.

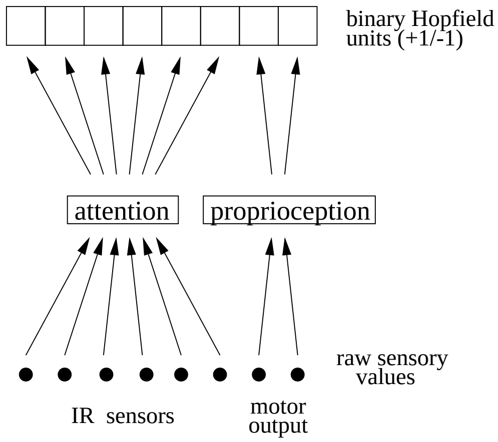


Figure 5.3: The learning setup. The attention module processes information from the IR sensors, and the proprioception module processes information from the motor system. The results from both modules make up the units of an associative learning pattern for the Hopfield network.

The reason a Hopfield Network was chosen as the learning architecture is firstly due to its simplicity — it does not require any parameters (see the discussion in Section 4.2 about simplifying the learning setup). Secondly, it was chosen because it learns in terms of binary patterns, which is appropriate for learning the presence and location of changes, as we will see below.

The learning setup is shown in Figure 5.3. The 'attention' module corresponds to the detection mechanism described above; it operates on each sensor independently, and outputs a value of 1 if the sensor is 'attentive' (*i.e.* if a change has been detected in that sensor), or -1 otherwise. This output is used to make up an associate learning pattern to be presented to the Hopfield network. Thus the first part of the learning pattern can be thought of as the 'perception' part, as it is determined from the perception of the environment through the sensors.

The second part of the learning pattern corresponds to the associated action, and can be thought of as the proprioception of the motors; it is handled similarly, but there is no attention to changes as there is for the sensors, as mentioned earlier. The two motor values (left and right, see Figure 3.9) are compared to determine whether a left turn, right turn, or none, has occurred. Steps 1 and 2 of the detection mechanism, described in the previous section, are applied to the difference between the two motor values. The calculated average is tested statistically for significance using a fixed pre-determined threshold, determined through trial and error such that only significant turns are taken into account, and not ones due to the default wander behaviour.

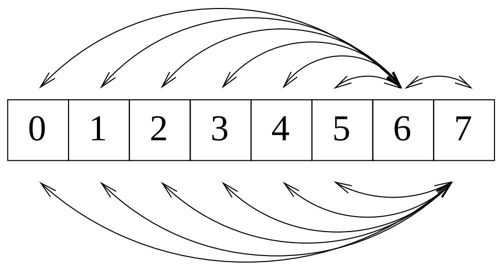


Figure 5.4: The associative pattern used to train the Hopfield network, containing perceptual units (numbered 0–5) and action units (6–7); associative connections between the units are shown.

This is where the abstraction of the motor data occurs. The result of the test then determines if the appropriate Hopfield unit is turned on or not (one unit for left turn, one unit for right turn; a forward move is represented by both units being off or on).

The learning pattern and the associative connections between the units are shown in Figure 5.4. The connections, or weights, are between each perceptual and action unit, and between the two action units. There are no connections interconnecting the perceptual units, the reason being that it was required that the values of the perceptual units are set *purely* by the perception of change mechanism so that the correct action can be learned for this task.

Learning Phase

The learning phase consists of 100000 steps. In each step the associative pattern is used to train the Hopfield network if any or both of the following two conditions hold:

1. stimulus enhancement, *i.e.* the demonstrator speeds up.
2. the system is attentive, *i.e.* one or more of the perceptual units are turned on.

Note that the first is used to test the effect of social interactions, namely the distinction between passive demonstrations and explicit signalling, as shown in Figure 5.1; and the second is used to test the importance of setting the saliency parameter in the attention system, which is attributed to the designer.

The consequence of presenting a pattern to the Hopfield network is a modification of the association strength (weights) between the units. The weight update follows a Hebbian rule as follows:

$$\Delta w_{ij} = \frac{1}{n} a_i a_j, \qquad i, j \in 1 \ldots n, \;\; i \neq j \tag{5.4}$$

where w_{ij} is the weight between unit i and j, and $\mathbf{a}$ is the learning pattern of size n (8 in this case). See Hertz et al. (1991) for more information on the Hopfield network.

Due to the Hebbian learning rule mentioned above, when two units are equal their association is strengthened (the weight connecting them is increased by $1/n$), and similarly when two units are different their association is weakened (the weight is decreased by $1/n$).

Recall Phase

After learning is complete, the robot is placed in the same environment on its own, and its ability to perform the learned task is tested. Additionally it is equipped with default wandering and obstacle avoidance behaviours. The recall phase consists of 100000 steps; at each step, perception through the sensors is handled as before, and again makes up the first part of the associative pattern (left part of Figure 5.3). That is, the perceptual units $a_i, i = 1 \dots m$, where m is the number of perceptual units, are set using the perception of change mechanism. The rest of the pattern is *recalled* by letting the values of the action units be predicted by the network through the weights, as follows:

0. initialise the action units randomly to 1 or -1

1. for each action unit $a_i, m < i \leq n$, calculate a prediction a'_i:

$$a'_i = \begin{cases} 1 & \text{if } \sum_{j=1, i \neq j}^{n} w_{ij} a_j \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{5.5}$$

2. if the prediction of any action unit is different from its actual value (*i.e.* $a_i \neq a'_i$), then change ('flip') it:

$$a_i \leftarrow -a_i \tag{5.6}$$

   and go back to step 1; otherwise finish.

The theory of the Hopfield network guarantees that this procedure will terminate in at most n iterations[2]. The overall effect of this procedure is that the different perceptual units contribute different strengths to the action units, according to the frequency with which each perceptual-unit action-unit pair was active in the learning phase. The perceptual units with most strength (highest weight) will influence most the activation of the action units. Once the activation of

[2]Note that there is a theoretical capacity of how many unique patterns the Hopfield network can reliably recall, which is a small fraction of n, the number of bits in the pattern. Since n in the implementation here is very small (8), we accept that the recall of the learned patterns will not be perfect.

the action units is determined, they are translated into actual motor commands that drive the robot.

5.1.3 Implementation on the Phototaxis Task

In this section, the effect of the two conditions that trigger learning, namely stimulus enhancement and perception of change, are tested. As mentioned above, they respectively test the influence from the teacher and from the designer on the learning.

Experimental Setup

The effect of stimulus enhancement is tested by comparing the results when only 'enhanced' learning patterns are considered by the Hopfield network, as opposed to when *all* learning patterns are considered. The effect of attention is tested by modifying the parameter for perception of change, namely the detection threshold mentioned in step 4 (Section 5.1.1). Since attention is used at both the learning and recall phases, this parameter plays an important role in both, and its effect is therefore tested independently in each phase. The experiments are carried with and without stimulus enhancement, and with different attention thresholds.

The design of the experiment is as follows: the learning phase is repeated 30 times for threshold values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$; at the completion of each run of the learning phase, the recall phase is repeated 20 times for threshold values $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$; these thresholds were found empirically to be the most representative of the different learning performances. This experimental design is depicted in Figure 5.5. Before each run of the learning phase both learner and demonstrator are placed in random locations in the environment, and the Hopfield weights are reset to zero; before each run of the recall phase the learner robot is placed in a random location in the environment on its own.

A one-way analysis of variance (ANOVA) is used to test each combination of learning threshold and recall threshold (dotted square in Figure 5.5); repeating the learning phase ensures that the results are not biased to a particularly successful or unsuccessful training episode, and repeating the recall phase for each learning phase ensures that the evaluation of the results is not limited, or obscured, due to a specific location in the environment (for example, the robot can get 'localised' in a small region of the environment if it learns to turn mostly in one direction). The ANOVA indicates when the results of a particular combination of learning and recall thresholds are consistent, and this can be used to test if the robot has really learned something, or if its performance is dependent on chance, its particular location in the environment, or both;

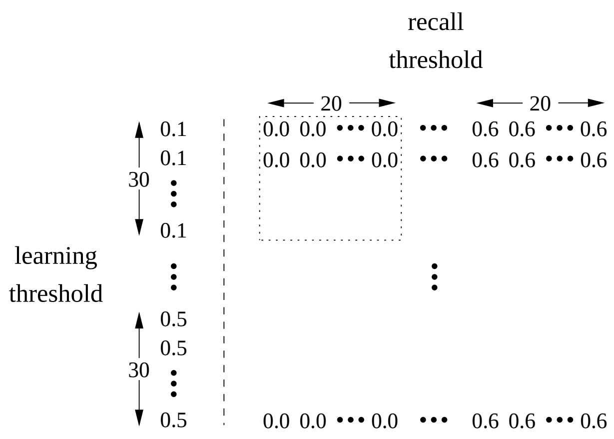


Figure 5.5: Experimental design. The systematic testing of the attention threshold is performed both at the learning phase and the recall phase. The repetition of each threshold is shown, see text for more details. Each combination of learning and recall threshold, denoted by the dotted square, is tested with a one-way ANOVA.

this will be explained further in the discussion of the results.

Note that normally a more correct analysis for this experimental design would be to perform a *single* two-way ANOVA, which would test the overall statistical significance of all threshold combinations, taking into account the actual trends in the thresholds. However, this analysis assumes that the variance is constant across all combinations, and this is suspected not to be the case due to the reasons mentioned above, and is also evident from the results of the one-way analyses that will be shown in Section 5.1.3. In spite of this, a two-way ANOVA has in fact been performed in order to support the conclusions, and it will be mentioned briefly[3].

Results

Let us first examine the quantitative benefit of the perception of change mechanism, by inspecting the number of patterns processed by the Hopfield network during the learning. This is shown in Figure 5.6, as a function of the attention threshold, separately for the cases with and without stimulus enhancement. Since more selective thresholds detect fewer changes the trend seen in Figure 5.6 is expected. Also apparent from Figure 5.6 is that the utilisation of stimulus enhancement reduces the number of patterns substantially (from a maximum of 100000 to less than 2000), but it does not affect the nature of the attention filter, as the two curves in Figure 5.6

[3]Many thanks to Murray and Irit Aitkin for suggesting and helping with this analysis.

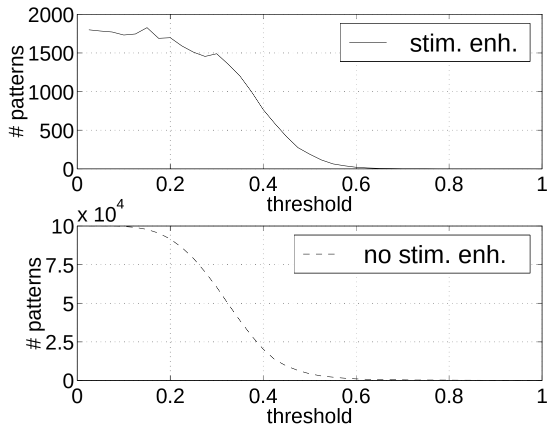


Figure 5.6: Number of patterns processed by the Hopfield network, using different detection thresholds in the learning phase, with and without stimulus enhancement.

are similar in shape. Further, it seems that when the threshold is higher than 0.6, the network is presented with a negligible number of patterns.

Of course, the detection threshold determines not only how many patterns are learned by the Hopfield network, but also how many different *kinds* of patterns are learned. For example, a very sensitive threshold will produce binary patterns whose units are mostly ‘on’, and similarly, a very insensitive threshold will produce binary patterns whose units are mostly ‘off’. Let us now examine the qualitative benefit of the perception of change mechanism in presenting a useful and representative set of binary patterns for learning. This is tested by evaluating the ability of the network to recall patterns correctly in the recall phase, by testing the robot’s ability to perform the task on its own. To do this, an evaluation measure scores the robot’s performance as the ‘energy’ that it acquires from the light sources. This energy is a function of the light intensity the robot senses as it wanders around the environment, using a weighted sum of the sensor readings; the central sensors have more weight than the outer ones, and so passing directly in front of a light source provides more energy than passing to its side.

As mentioned in Section 5.1.2, in the recall phase the robot moves around with a default random wandering behaviour, and whenever the perception of change mechanism detects a change in one of the sensors, it takes over the control of the robot. For comparison, the energy acquired by a robot only controlled with a wandering behaviour is also measured, as well as the energy acquired by a robot controlled by a hand-crafted phototaxis behaviour.

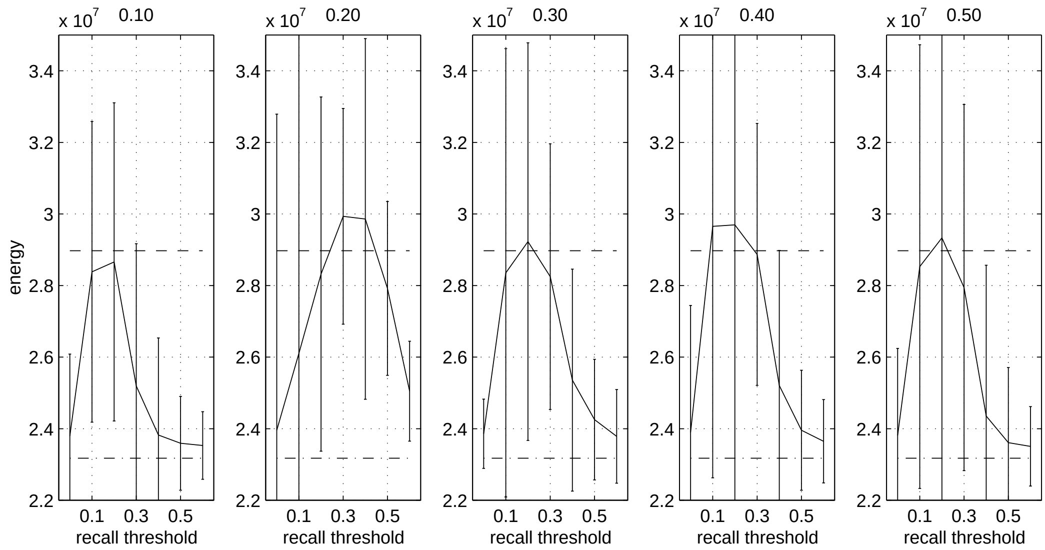
Figure 5.7(a) shows the energy results when stimulus enhancement is used. A separate plot is provided for each learning threshold, and the threshold is indicated at the top of the plot; in each plot the energy is plotted as a function of the recall threshold. Each point in the plot is the overall average energy for the particular combination of learning threshold and recall threshold (*i.e.* the overall average of the dotted box in Figure 5.5), and the error bar is the 95% confidence interval for that average, which is based on the ANOVA calculation of the total sum of squares[4]. Figure 5.7(b) similarly shows the results when stimulus enhancement is not used.

Note that the total sum of squares, SS_t, can be decomposed into the sum of squares within experiments, SS_w, corresponding to the variability of each combination of learning threshold and recall threshold, and the sum of squares between experiments, SS_b, corresponding to the variability in the repetition of each combination. The former, *i.e.* SS_w, can be regarded as capturing the variability due to the recall phase, because a particular combination of learning threshold and recall threshold is tested by training the network once, fixing the weights, and then testing the recall phase 20 times (see Figure 5.5). The effects of re-training the network to re-testing the same combination (30 times) is averaged out by this measure, and is instead captured by the latter component of the decomposition, SS_b. Therefore, SS_b can be regarded as capturing the variability due to the learning phase.

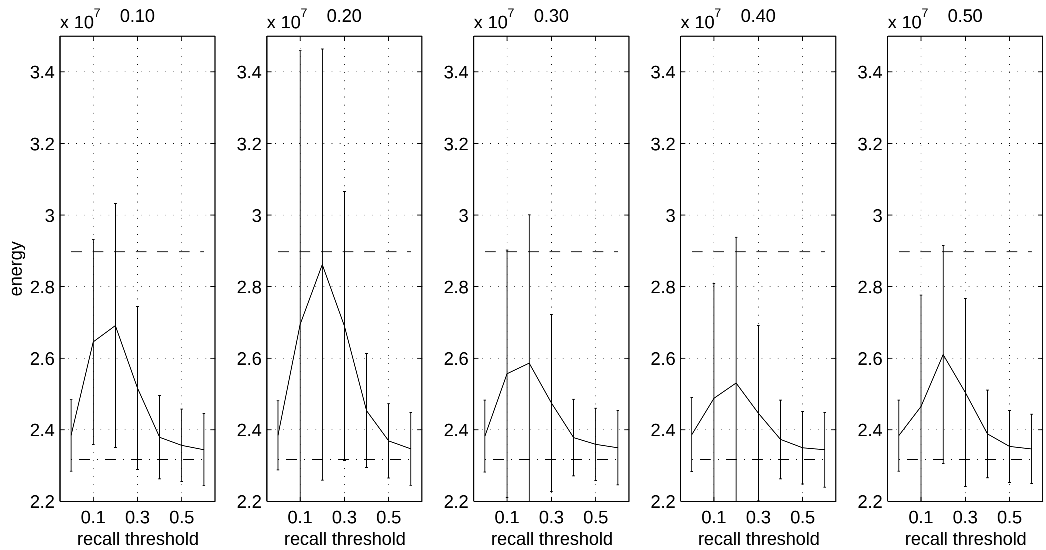
In Figure 5.7 the high variability of the results is clearly visible through the size of the error bars. This high variability is a result of inconsistencies at both the learning phase and the recall phase, as described in Section 5.1.3. The ANOVAs (not shown here) for the different threshold combinations all suggest that the SS_b is very high when compared to the SS_w, indicating that the main source of the inconsistencies is in the learning phase. This is the reason why a two-way ANOVA of all the data is not a suitable analysis, as mentioned in Section 5.1.3. A two-way ANOVA was nevertheless performed (not shown here), and it too suggests that the variance between the different threshold combinations is much higher than within each one, and that most of this variability is *not* due to the different values of the learning threshold, but rather due to the repetition of the same threshold. This indicates an inconsistency under unchanged learning conditions.

Such inconsistency is an indication that the Hopfield network does not generally learn the task well. It has been observed that the main source of inconsistency arises when the network learns a dominant move, for example only left turns or only right turns; in such a situation, the robot will acquire a very high energy if it happens to be near a light source because it will spin on the spot, and acquire a very low energy if placed far from a light source (recall that there is

[4]The value actually used to calculate the confidence interval is the total mean square estimate of the variance, which is obtained by dividing the total sum of squares by its degrees of freedom.



(a)



(b)

Figure 5.7: Energy results obtained (a) with stimulus enhancement, and (b) without stimulus enhancement. A separate plot is shown for each learning threshold, shown at the top of the plot; each plot is a function of the recall threshold; a point on the plot is the overall average energy of the learning threshold and recall threshold combination, and the error bar is the overall standard error of that average, as given by the ANOVA test. The energies acquired by a hand-crafted behaviour and a random behaviour are also shown as upper and lower baselines, respectively.

	learning threshold					overall
	0.1	0.2	0.3	0.4	0.5	
stimulus enhancement	0.64	0.49	0.48	0.68	0.57	0.57
no stimulus enhancement	0.56	0.65	0.52	0.40	0.55	0.53

Table 5.1: Proportions of positive turn-ratio scores for each learning threshold, averaged over the recall thresholds.

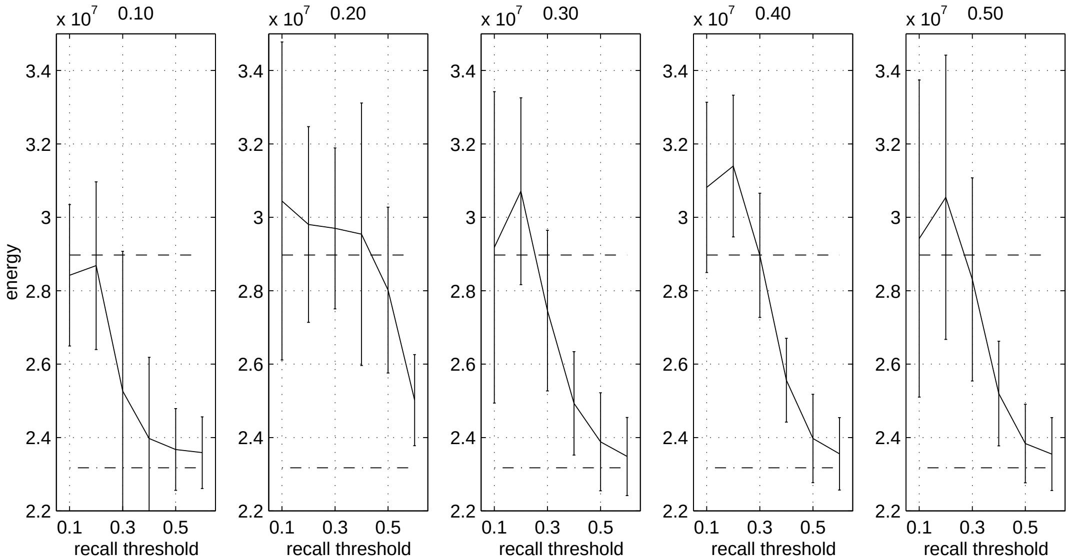
a random re-positioning between recall runs), thus introducing inconsistencies in the results. It seems that the energy measure is not completely informative on its own.

Because it was believed that the above explains most of the variation in the results, another performance measure was devised to both capture and filter out the undesirable cases from the results. This new measure scores the robot's behaviour according to its *turn-ratio*: a negative score is given when a turn in one direction dominates over a turn in the other direction, and when a turn in any direction dominates over a forward move; a zero score is given if there are no turns at all; the score is between -20 and 20, where any positive score is a desirable one, and an increase in positive score is proportional to an increase in the forward-to-turn ratio.

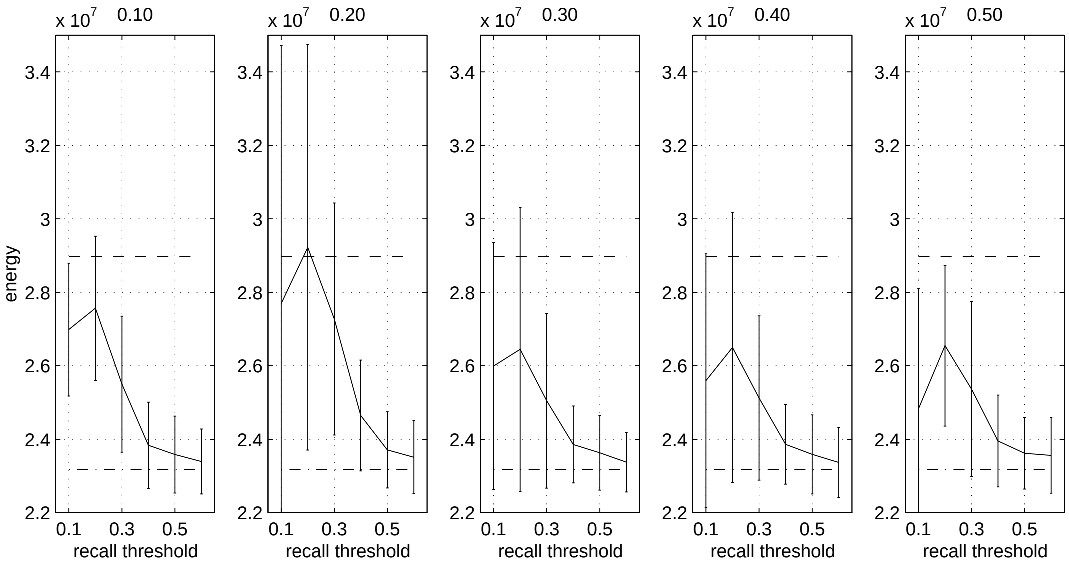
Table 5.1 shows the proportion of positive scores obtained by each model, for each learning threshold (averaged over the recall thresholds). Overall these proportions are not very high for both models (0.57 and 0.53), which suggests that the network does not learn well overall. However, when we look at the results for each learning threshold individually, we can see that the network learns more reliably at particular thresholds.

To demonstrate that most of the inconsistencies in the results can be accounted for by the turn-ratio score, the results are filtered such that only the results where the score is positive are used, and the ANOVA is repeated. That is, 57% of the results using stimulus enhancement are kept, and 53% of the results not using stimulus enhancement are kept. The filtered energy results are shown in Figure 5.8.

The filtered energy results are more consistent, at least when stimulus enhancement is used. There seems to be a similar trend for all learning thresholds: a peak in energy for recall threshold 0.2, with a drop-off of energy to either side. The different learning thresholds seem to result also in similar energy values, all as high as each other, with the exception of the lowest learning threshold (0.1), where the energies are not as high. Note that for all learning thresholds, the filtered results do not contain any data for recall threshold 0 — when no attention is used in



(a)



(b)

Figure 5.8: Filtered energy results (a) with stimulus enhancement, and (b) without stimulus enhancement, obtained by removing the data where the turn-ratio score is negative, from the results shown in Figure 5.7.

the recall phase, the robot cannot recall the desired behaviour, even if the learning phase was successful. This suggests that the proportions shown in Table 5.1 would be more indicative

	learning threshold					overall
	0.1	0.2	0.3	0.4	0.5	
stimulus enhancement	0.75	0.57	0.55	0.79	0.66	0.66
no stimulus enhancement	0.66	0.75	0.60	0.46	0.64	0.61

Table 5.2: Proportions of positive turn-ratio scores, disregarding the scores obtained with recall threshold 0.

if the scores obtained when the recall threshold is 0 were omitted from the calculations. The proportions are re-calculated and shown in Table 5.2.

What distinguishes the learning thresholds when stimulus enhancement is used is the inconsistencies in the results: the 0.4 threshold seems to result in the most consistent results (smallest error bars), as suggested also by the proportion of positive turn-ratio scores, shown in Tables 5.1 and 5.2. Further, with the 0.4 learning threshold, the highest energies compare favourably with the energy achieved by a hand-crafted behaviour, that is when the recall threshold is in the range 0.1–0.3.

When stimulus enhancement is not used the results do not appear to change significantly due to the turn-ratio filtering (see Figure 5.8(b) versus Figure 5.7(b)). The results are only slightly more consistent, which suggests that there are inherent inconsistencies due to other sources and hence learning is simply not as reliable with this model. The energy trend itself is similar to when stimulus enhancement is used, but the actual energies are much lower, except when the learning threshold is 0.2 and the energies are almost as good. In terms of the ratio-score, although it is almost as good as when stimulus enhancement is used with learning threshold 0.2 (see Tables 5.1 and 5.2), the variability in the energy with the optimal recall threshold (0.2) is very high, suggesting that the high score is due to other recall thresholds (which do not correspond to energies that are as high).

To summarise, the most consistent and correct learning occurs when both attention and stimulus enhancement are used, and attention works best when the threshold is 0.4 at the learning phase, and 0.1–0.3 in the recall phase. It is interesting that the Hopfield network requires the thresholds to be different at the learning and recall phases; this will be discussed further below.

5.1.4 Conclusion

This section presented the first of three sets of experiments involving learning at a high level of abstraction. The learning ‘data’ here corresponded to binary patterns, and these were obtained by abstracting from the robot’s raw sensory data using a perception of change mechanism. This mechanism performs comparisons between consecutive perceptions, and the detection of change is signalled through a threshold. The detection threshold is in fact a saliency parameter, and it was tested explicitly in this section. The abstraction of the motor data was achieved in an ad-hoc manner by setting a saliency parameter through some trial-and-error.

The perception of change parameter determines both how many patterns are learned, and how many different *kinds* of patterns are learned, thus respectively dealing with the quantitative benefit of attention for modulating learning, and with the qualitative benefit for abstracting raw data usefully for learning. The latter is equivalent to the number of SOFM nodes that result from different novelty thresholds, in the attention system presented in Chapter 3. Note that there is a theoretical capacity limit to how many different kinds of binary patterns a Hopfield network can learn (or rather, how many patterns it can recall); this capacity can be used as a limited-resources argument in favour of attention, as discussed for the number of SOFM nodes in Section 3.3. However, this issue was not tested explicitly.

The results show that there is a desirable region in the detection threshold space, which shows that care must be taken in setting the value of this parameter, so that the representation for learning is useful. An interesting result was that a different detection region was desirable for learning and recall, the reason for which is not immediately clear because the environment is the same for the learning and recall phase. A possible reason is that since in the recall phase the network weights are not being modified, it is not as crucial to be selective as in the learning phase, where mistakes could be costly. Hence in the learning phase the threshold should be higher (0.4), *i.e.* more selective, and in the recall phase it can be lower (0.1–0.3). If this is indeed the case, then it is a further demonstration that care must be taken in abstracting the data for learning.

The results also show that the explicit signalling from the demonstrator significantly improved the performance. Therefore the important conclusion from this experiment (summarised as ‘2’ in Figure 5.1) is that even with designer effort involved in abstracting the sensory data and motor data, the best performance is achieved with the strongest type of social interaction that was attempted. The next two sections will show that when the designer effort increases further, the stronger social interactions are not as significant. As in the simulated wall-following experiment discussed in the previous chapter, here too active demonstrations were not attempted, due to the difficulties in designing active robotic demonstrators.

5.2 Fitting Gaussians to Motor Values

This section provides another demonstration of learning at a high level of abstraction. As discussed in Chapter 3 and again at the start of this chapter, considerable designer effort is required in abstracting the raw perceptual data usefully for learning. We saw a demonstration of this in the previous section, which also demonstrated the effort involved in abstracting the motor data. In the learning setup here, a similar type of design effort is required, corresponding to setting a saliency parameter for the motor data, and here too this is achieved empirically in an ad-hoc manner. However, in this section an additional form of designer effort will be used to explicitly investigate the implications of increasing the design effort on the social interactions. In other words, this section explicitly investigates how increasing the design effort compares with increasing the strength of the social interactions in terms of learning performance (see experiment set 3 in Figure 5.1).

This section presents the first of two approaches that use the attention system presented in Chapter 3 for abstracting the raw data for learning; the learning is achieved by directly coupling perceptual SOFM nodes with motor structures. The second approach will be presented in Section 5.3, where the learning is in terms of high-level perceptual-motor *targets*. In contrast, here the learning is performed on the actual raw motor values of the robot. As motivated in Section 4.2, a simple learning setup is desired in order to test the benefit of designer effort and social interactions. One approach is to keep a record of all the motor values, for each perceptual SOFM node. That is, each node ‘remembers’ the motor information associated with it, such that it is able to recall this information in the recall phase. Note that because the learning involves the raw motor data, there is no design effort needed for abstraction of the data for learning. Instead, this effort is required in the recall phase, in reliably abstracting the output into actions — see the discussion in Section 2.4.2 on the need to abstract actions from motor data for reliable execution.

This approach is tested on the simulated wall-following task, presented in Section 3.2.2 and tested in Chapter 4. As in Chapter 4, the effect of social interactions is tested here through the effect of ‘social facilitation’ (see Section 3.1) — the learning performance when the robot only learns from demonstrated experiences is compared to when the robot also learns when it loses the teacher (‘passive demonstrations’ and ‘alone & social’, respectively, in Figure 5.1).

The effect of designer effort is tested firstly by evaluating the importance of setting the saliency parameter usefully for learning, and secondly by explicitly comparing the learning performance with and without the additional effort.

5.2.1 Learning Setup

The approach mentioned above suggests recording all the motor information for each SOFM node. However, rather than recording all the information, a more concise representation can consist of three statistics: a mean, variance and number of motor values the node was exposed to. In order to calculate these statistics, there is no need to store each motor value — instead, two sums can be updated incrementally, $\sum x_i$ and $\sum x_i^2$, where x_i is the current motor value; these sums can be used to obtain the mean, $\bar{x}$, and variance, s^2, as follows

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad s^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} {x_i}^2 - n\bar{x}^2\right)$$

where n is the number of motor values that the SOFM node was exposed to.

If it is reasonable to assume that the motor values follow a gaussian (Normal) distribution, then one can use the above statistics to fit a gaussian to the motor data, using the following density function:

$$f(x) = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{1}{2s^2}(x-\bar{x})^2} \tag{5.7}$$

which gives a probability of observing a particular value x.[5]

After fitting such a distribution to each SOFM node in the learning phase, an appropriate recall strategy would be to sample from the distribution fitted for the SOFM node that the input activates, as shown in Figure 5.9. Sampling a random value y from a gaussian can be achieved by obtaining a random probability p (between 0 and 1), and then solving the following for y:

$$p = \int_{-\infty}^{y} f(x)dx. \tag{5.8}$$

This corresponds to calculating an inverse cumulative distribution function (cdf). A numerical solution is obtained here using code downloaded with permission from the Internet (Acklam, 2003)[6], however this code assumes that the distribution is a Standard Normal distribution (mean 0, variance 1), and so some standardisation is required, as follows: if x is the motor value that we want to sample from a Normal (gaussian) distribution with mean $\bar{x}$ and variance s^2, then $z = (x - \bar{x})/s$ is the standardised version of x with mean 0 and variance 1; the inverse cdf

[5] Note that gaussians can also be fitted to multi-dimensional data.

[6] The code used here was written by V. Natarajan.

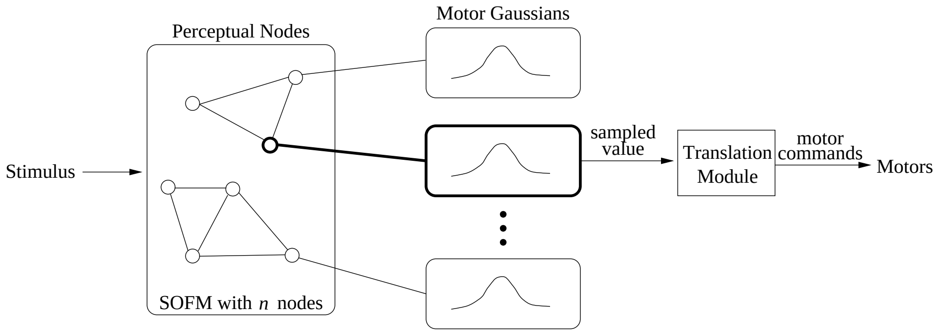


Figure 5.9: Learning at a high level of abstraction is achieved here by fitting a gaussian to the motor values that each SOFM node is exposed to. In the recall phase, motor data are sampled from the gaussians, and then abstracted to actions to produce motor commands.

program takes p as input and returns a z value, which can then be converted back to x using $x = sz + \bar{x}$.

Because this procedure is parametrised by $\bar{x}$ and s^2, it will sample values biased towards the mean of the observed data, with a random spread biased by the variance. If the sampled output is appropriately translated to motor commands, then the learned behaviour can be recalled, with each perceptual input producing a motor output, as shown in Figure 5.9. There is a potential problem with this procedure if a SOFM node was added very late in the learning phase and so did not get much exposure to data, or if it was exposed to outliers; both would result in a very high variance and therefore in a wide gaussian; in the recall phase this would bias the output to undesirably high values. Such nodes are undesirable — they do not represent information that will be useful in the recall of the behaviour. The additional designer effort mentioned above is concerned with dealing with these nodes, as will be discussed below.

5.2.2 Implementation on the Simulated Wall-Following Task

The above approach is implemented in simulation, for the wall-following task. This task was described in Section 3.2.2, and it was used to demonstrate learning in the previous chapter. The environment is shown again in Figure 5.10. Recall that the learning phase consists of 50000 steps in which the learner follows behind the demonstrator who is following the walls on both sides, with an ‘interrupt’ signalled every 5000 steps, which forces the demonstrator to turn towards the middle of the environment. The purpose of the interrupt is to ensure that the demonstrator exposes the learner to the full complexity of the task, rather than follow the wall

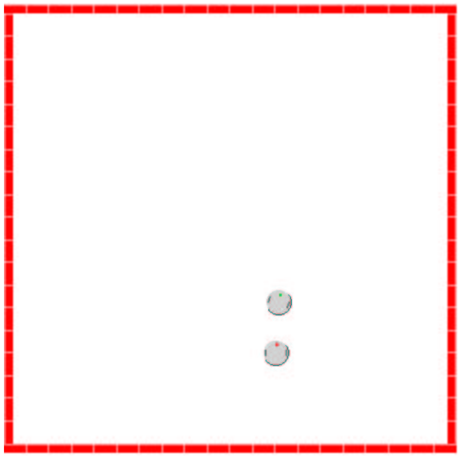


Figure 5.10: The simulated environment used in the wall-following experiments.

only on one side for the duration of the run.

The motor data correspond to the difference between the values of the left and right motors. If the robot performs left turns, right turns, and forward moves equally, the distribution of the motor data can be reasonably assumed to follow a gaussian distribution. Of course, not every SOFM node will be equally exposed to these actions, but an approximately gaussian distribution should nevertheless be observed, as long as the node is exposed to enough data.

As described above, the learning phase involves an incremental update of the two sums needed to calculate the statistics. These sums are kept separately for each node, and they are updated each time a node is active. Note that this means that the approach is expected to generalise well over the whole motoric experience of each node for the duration of the learning phase, which might be unreasonable, because nodes can move considerably in the input space before settling. Nevertheless, it was decided to keep the learning setup as simple as possible (see Section 4.2), and avoid introducing new parameters.

Figure 5.11 shows an example of a trained SOFM together with the motor values that each SOFM was exposed to, and the resulting statistics. We can use the interpretations from Section 3.3.4 (see particularly Figure 3.33) to help interpret the example here. Nodes 0 and 1 correspond to ‘no-wall’ perceptions, particularly node 1 whose motor values have a high standard deviation compared to the other nodes — that node is exposed to many turns in both directions. Nodes 2, 3, 4, and 5 are the ‘wall’ nodes, particularly nodes 3 and 5 because their motor values have lower standard deviation compared to the other nodes — they are exposed to motor data closely distributed around 0; also note that node 4 has a mean motor value quite

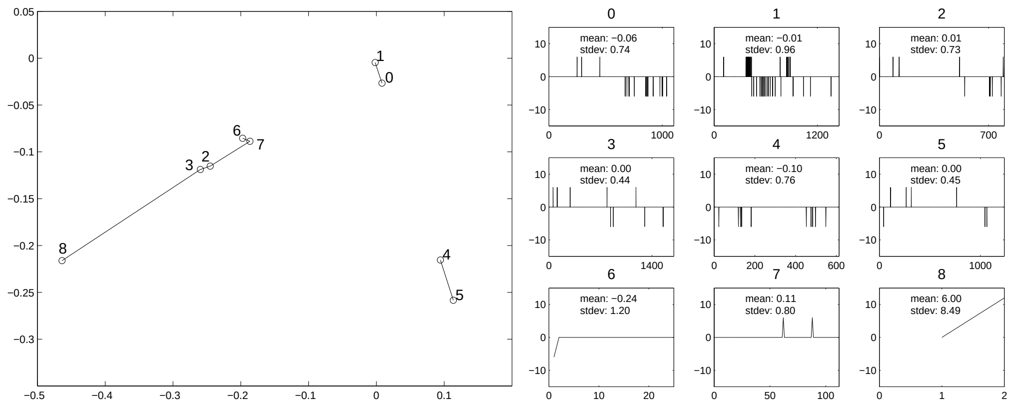


Figure 5.11: An example of fitting gaussians to motor data. Each of the SOFM nodes shown on the left records all the motor values it is exposed to, calculated as the difference between the left and right motors; these values are shown on the right, separately for each node, and including the summary statistics that are used to sample motor values in the recall phase. In each plot on the right, the horizontal axis indexes the data points, and the vertical axis specifies the motor values.

far from 0 compared to the other nodes — that node was exposed to turns only in one direction; this may cause a problem in the recall. Nodes 6, 7, and 8 have motor values with a high mean and/or a high standard deviation; these are undesirable nodes as discussed above, because they are not exposed to enough data to generalise well (25, 115, and 2 data points, respectively).

The recall phase consists of 50000 steps, with the ‘interrupt’ occurring every 7500 steps, to avoid the robot following the wall on one side for the duration of the run, and thus not testing the learned behaviour fully. Further, to deal with unsuccessful learning, as in Chapter 4, a built-in ‘obstacle-avoidance’ behaviour is used to safe-guard the robot, and the evaluation is penalised whenever this occurs.

In the recall phase, two heuristics are used to sample from the motor gaussians. First, nodes are deemed undesirable if they are exposed to fewer than 200 data points, and/or the standard deviation of their motor values is higher than 4. Second, if there are several equally good nodes, grouped using a threshold for similarity, we select between them randomly; this was done because in some situations a perceptual cluster is represented by more than one node (for example clusters 0–1, 2–3, 4–5, and 6–7 in Figure 5.11), and therefore the best generalisation of the motor values is achieved when all the nodes of a particular cluster are considered for recall. These heuristics correspond to additional designer effort — they bias how the learner interprets what it has learned.

Note, however, that in this particular implementation, the heuristics are set fairly intuitively, not arbitrarily. In the first heuristic, the criteria for undesirable nodes are chosen to identify obvious outlier data — recall that the length of the learning phase is 50000, so if a node is exposed to fewer than 200 data, this is indeed a tiny fraction (0.04%); also, most nodes have a standard deviation less than 1, and so data with a standard deviation higher than 4 unquestionably contain an outlier. In the second heuristic the threshold for similarity is set very conservatively, so that nodes have to be very similar to be grouped. These heuristics are nevertheless imposed by the designer. The benefit of the first heuristic is tested explicitly in the experiments, where undesirable nodes are either ignored or not.

Once a value x has been sampled from the gaussian of the node that is activated by the current input, as described in Section 5.2.1, this raw value must be abstracted to produce an action, as shown in Figure 5.9. This corresponds to setting a saliency threshold and, as in the previous section, this is done empirically in an ad-hoc manner. The threshold is used as follows:

- if $|x| < 1.0$, move forward;
- if $x > 1.0$, turn right (left-motor value greater than right-motor value);
- if $x < -1.0$, turn left.

For correct learning, the above procedure is expected to work, because ‘wall’ nodes (see nodes 2, 3, 4, and 5 in Figure 5.11) should be exposed to very few turns, and so their gaussians should be centered around 0, and have a low variance; whereas the ‘no-wall’ nodes (see nodes 0 and 1 in Figure 5.11), should be exposed to many turns, probably in both directions, so although their gaussians will also be centered around 0, their variance should be much higher; therefore sampling from the ‘no-wall’ nodes should output more extreme values than the ‘wall’ nodes who should, on average, sample values closer to 0.

Results

The performance of the system is evaluated numerically, using the same ‘energy’ measure used in Chapter 4. Recall that the energy is a measure that the robot acquires from the wall at particular orientations from it, with the most energy acquired when the robot is exactly parallel to the wall. At the end of the recall episode the accumulated energy measures how well the robot performs the wall-following task. The energy is measured systematically as a function

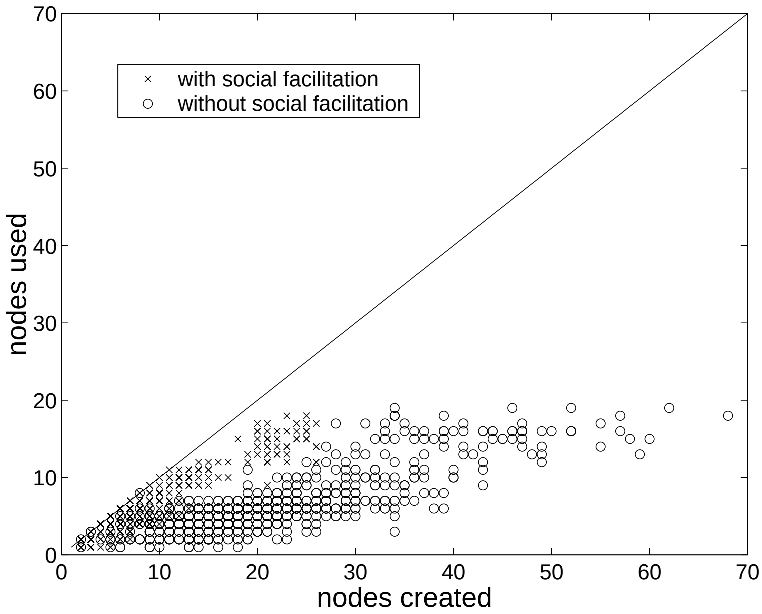
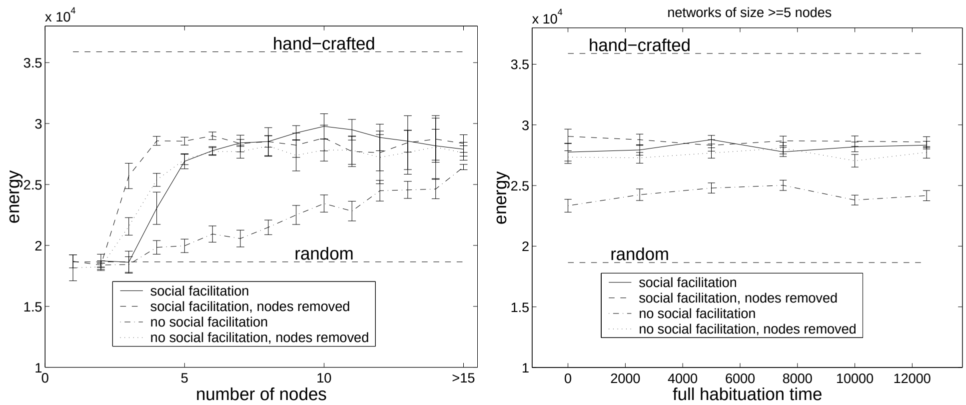


Figure 5.12: Number of nodes used in the recall phase when the condition to ignore undesirable nodes is utilised, versus the number of nodes created during the learning phase. A node is ‘desirable’ if it is exposed to a minimal amount of data (more than 200 observations), and its standard deviation is reasonably low (less than 4).

of the size of the SOFM network and the length of full-habituation time, separately for cases where social facilitation is used or not.

First, to get an idea of how many undesirable nodes are produced, Figure 5.12 shows the number of nodes used when undesirable nodes are ignored, versus the number of nodes created. We can clearly see from Figure 5.12 that the learning of undesirable nodes is only a significant problem when social facilitation is not used. This is not surprising because without this condition the robot learns even when it loses the teacher, and is therefore exposed to non task-relevant sensory-motor data: there are many irrelevant perceptions, resulting in many more SOFM nodes, each exposed to few data. Also we can see from Figure 5.12 that there is an emergent maximum number of *desirable* nodes, approximately 20.

Figure 5.13 shows the energies obtained under the different conditions, as a function of number of nodes and full-habituation time (recall that the number of nodes is controlled by the novelty threshold). Firstly, it appears that SOFMs with less than 5 nodes cannot reproduce the task well in all the conditions; the full-habituation plot therefore does not include results



(a) (b)

Figure 5.13: Evaluation of the recalled behaviour, as a function of (a) network size, and (b) full-habituation time; two conditions are tested: whether social facilitation is used, and whether undesirable nodes are ignored; energy is measured from the sensors at particular configurations from the wall, and compared to energies acquired by a hand-crafted behaviour and a random wandering behaviour.

obtained by these small networks, so that the plot is more informative.

As suggested by Figure 5.12, we see also from Figure 5.13(a) that there is no significant effect from ignoring undesirable nodes when social facilitation is used, except that it reaches a maximum performance with 2–3 fewer nodes. This shows that social facilitation ensures that the learner is exposed to consistent and sufficient data. In contrast, there is a very significant effect to ignoring undesirable nodes when social facilitation is *not* used. It is very interesting that when such nodes are ignored, the performance is as good as with social facilitation! This suggests that the non socially-transmitted data (*i.e.* data perceived when the demonstrator is lost) are outlier data, which are represented by distinct SOFM nodes, and can therefore be detected and dealt with using simple heuristics.

It appears from Figure 5.13(a) that when social facilitation is not used, and undesirable nodes are not removed, the performance is increasing as the number of nodes increases. This is in fact a misleading result: as more nodes are used, each one is exposed to few experiences and therefore the nodes are mostly undesirable; in the recall phase high values are therefore sampled from them because of their high variance, and the resulting actions are mostly turns in both directions; on average the turns cancel each other out and the emerging behaviour is moving forward all the time. Thus the robot acquires high energies when it is close to the walls, but for the wrong reasons. The other three curves in Figure 5.13(a) do in fact correspond to the desirable behaviour, as confirmed by visual inspections.

From Figure 5.13(b) we see that the habituation parameter does not have a significant effect, that is, that any amount of modulation works. This means that similar performances can be achieved by considering fewer experiences.

5.2.3 Conclusion

This set of experiments explicitly compared learning with and without additional effort by the designer. Here the attention system is the SOFM presented in the first experimental chapter, and the learning of the task is achieved by fitting a separate gaussian to the low-level motor data for each SOFM node. In the recall phase, a motor value is sampled from the gaussian of the active SOFM node. For reliable execution of the task, this low-level motor value must first be abstracted to a higher-level action, and this is performed in an ad-hoc manner by the designer by setting a saliency threshold on the raw motor data, similarly to the experiment in the previous section.

The results show that there is a desirable range for the novelty detection threshold, which results in a certain number of nodes, not too many or too few. A limited-resources argument can therefore be made, where there is a limit on the number of nodes allowed (*c.f.* number of different kinds of binary patterns in Section 5.1); the same argument can be made in the next set of experiments. However, as mentioned in Chapter 3, this issue is not tested explicitly in this thesis. The results also show that modulation is not significant, that is, one can achieve similar performances by learning from fewer experiences. This is in contrast with the learning approach used to learn this task in Chapter 4, where the learning on raw unstructured data benefited from as much data to generalise from as possible. Here, because the learning occurs on structured data, generalising from these data is easier and therefore requires less repetition.

As mentioned above, in this section a similar type of designer effort was involved in abstracting motor data as in Section 5.1. The additional effort from the designer here corresponds to a heuristic that can be used in the recall phase. The heuristic is used to remove unreliable gaussians — those that have a high variance and/or contain few data points, and therefore correspond to outlier experiences. The results show that without this heuristic the robot should not learn when it loses the teacher; however, *with* the heuristic the restriction can be relaxed. Thus the heuristic is useful for capturing and ignoring the experiences of the robot when it loses the teacher.

The important conclusion from this section (summarised as '3' in Figure 5.1) is that through more design effort, the problem of the learner losing the demonstrator can be resolved and the performance improved; but alternatively if the stronger form of social interaction is used, where the robot does not learn when it loses the demonstrator, the performance can also be improved and then the additional design effort does not improve the results further.

5.3 A Developmental Approach

This section presents a different approach to the one presented in Section 5.2, of using the attention system presented in Chapter 3 for learning at a high level of abstraction. A biologically-inspired developmental approach is presented here, where rather than abstracting from the low-level motor data *during* the social interactions, the robot has a set of basic sensorimotor skills *prior* to the social interactions. The basic skills are generic, and through the social interactions the robot learns how to use some of them to perform a particular task. With such an approach, the abstraction of motor data is arguably more reliable when it comes to learning the task because the robot already has basic motor control, as discussed in Chapter 2.

The idea here is that during the social interactions, the robot's *perceptual* experiences are self-organised as before, and then in the recall phase, when it comes to executing the learned task, the perceptual structures (SOFM nodes) are used to trigger the appropriate sensorimotor skills, which then take care of the low-level motor control. This setup relies crucially on the fact that all the necessary information needed to trigger the appropriate sensorimotor skills comes *only* from the robot's perceptions — this will be discussed further at the end of this section.

Therefore, as with the other learning approaches presented in this chapter, here too care must be taken by the designer in abstracting the raw perceptual data usefully. However, compared to these other approaches, here the most *additional* effort is required by the designer who must either provide the basic sensorimotor skills directly, or bootstrap some additional learning with which the robot can obtain such skills. As we have done so far in this chapter, we will examine the implications of such design effort on the social interactions (see experiment set 4 in Figure 5.1).

The work reported in this section forms part of collaborative work with George Maistros. With the exception of Section 5.3.5, the experimental content in this section is entirely and solely the contribution of the author of this thesis. The architecture shown in Figure 5.14 was designed in collaboration with George Maistros, as a combination of individual research components: the SOFM appearing on the left of the figure corresponds to the attention system presented in Section 3.3, while the Motor Schemas and Inverse Model are components from the research of George Maistros. The discussion throughout this section (5.3) is mainly the contribution of the author of this thesis. The biological inspirations presented in this section are a result of the research of George Maistros; their discussion is the contribution of the author of this thesis. The presentation of the collaborative work is tailored in this section to support the arguments in this thesis; some unrelated biological and implementational issues have been left out, and the reader is referred to (Marom et al., 2001) for the complete description of this work in progress.

5.3.1 Biological Inspirations

The main inspiration for the approach presented here is a biological finding of the existence of *mirror neurons* in monkeys. Neurophysiological experiments on *macaque* monkeys show that neurons in the rostral part of inferior area 6 (area F5), namely mirror neurons, have both visual and motor properties (Rizzolatti et al., 1988). Brain imaging and transcranial magnetic stimulation studies illustrate the presence of neurons with similar properties in the human brain as well (Decety et al., 1994; Fadiga et al., 1995). Based on their visual and motor properties it is believed that these neurons may form the fundamental basis for imitation in primates (Rizzolatti et al., 2000).

Single neuron studies by Gallese et al. (1996) and Rizzolatti et al. (1996) explored further the properties of F5 neurons and exposed a strong relationship between perception and motor control. For instance, mirror neurons fire both when the monkey performs an action (motor stimulus) and when it observes another monkey or the experimenter perform that same action (perceptual stimulus). Therefore these properties of mirror neurons support the approach mentioned above, whereby some common representation of perception and action can be triggered purely by the perception of a demonstrated task.

Further, one of the most interesting properties of F5 neurons is their high selectivity towards the kind of actions and even finger configurations. In fact, there are different levels of this selectivity: neurons that discharge only to specific finger configurations of specific grasps, others that discharge to a specific grasp regardless of finger configuration, and yet others that discharge to the achievement of a goal (*e.g.* hand apprehension of a specific type of object), regardless of the way this is achieved. Thus there is an inherent level of granularity in what these neurons represent. These properties of the mirror neurons provide further support of their suitability as biological counterpart to the kind of abstraction from raw data discussed in this thesis. The issue of saliency must be somehow incorporated in the activation of mirror neurons.

However, there is no neurophysiological evidence of how mirror neurons are learned or built. Therefore, unfortunately the current research in neurophysiology does not provide any suggestions to the problem addressed in this thesis, regarding the extent to which the abstraction in the mirror neurons is pre-existing or discovered from experience and social interactions. Some believe in the existence of basic knowledge about fundamental motor control in early infancy prior to the development of advanced motor skills (Meltzoff and Moore, 1989). However, it is obvious that mirror neurons reflect an individual's experiences and motoric capabilities, and therefore with the lack of further evidence it seems that a computational implementation where structures are self-organised from experience is suitable.

5.3.2 Learning Setup

As mentioned above, the work reported here is part of collaborative work in progress with George Maistros. In the early work of Maistros and Hayes (2001), they conceptualise the common representation of perception and action, that is, mirror neurons, as a coupling of perceptual and motor *schemas*, inspired from Arbib's Schema Theory (Arbib, 1981). The collection of these couplings are referred to as a *mirror system*. Each perceptual schema recognises some temporal segment of the stimulus, and each motor schema holds a sequence of motor structures that can generate some part of a behaviour, where these structures are at a suitably high level of abstraction in accordance with mirror neurons; in fact, these structures correspond to perceptual-motor *targets*. The schemas alone cannot produce a behaviour — they rely on basic sensorimotor skills, in the form of an *inverse model*, which is a mechanism that, given the robot's current state (perception and proprioception) and the desired state, calculates the motor commands that best achieve the desired state, as shown in Figure 5.14.

Here, perceptual schemas are replaced with SOFM nodes, and in experiments reported in (Marom et al., 2001) it was found that due to the nature of the tasks involved, there is no need to store sequences in the motor schemas, and in fact, there is no need for an actual motoric representation. It was found that the perceptual representation in the SOFM provides a sufficient target for the inverse model, that is, it is sufficient for triggering the appropriate basic sensorimotor skills, and no further information needs to be stored in the motor schemas. In other words, the perceptual representation specifies what the robot should be perceiving for the various parts of the task, and this specification is sufficient for the inverse model to provide motor commands which will enable the robot to have these experiences and hence execute the task. The reason why this is possible for the kind of tasks examined here but not for other

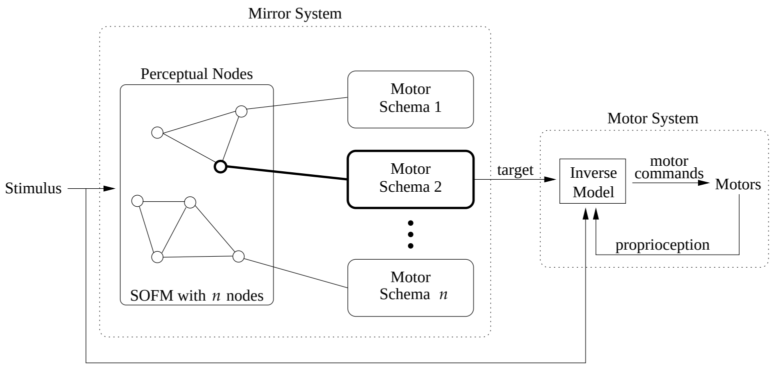


Figure 5.14: The architecture consists of (1) a mirror system, which is a coupling of perceptual and motor structures that are built up from experience; and (2) a motor system, which consists of an inverse model that can convert the output of the mirror system into motor commands.

kinds of tasks will be discussed in more detail after the presentation of the experiments, in Section 5.3.6. In the more general implementation of the architecture, motor schemas can hold important motoric information — this will also be discussed in Section 5.3.6.

Therefore, the learning phase corresponds to simply training the SOFM as described in Chapter 3. In the recall phase, the perceptual input activates one of the SOFM nodes, and this perceptual representation is used as a target to be achieved by the inverse model. The learning setup is tested on three different implementations, all discussed and analysed in Chapter 3: the simulated wall-following experiments, the physical wall-following experiments, and the simulated humanoid object-interactions experiments. In each, the performance is evaluated in terms of the two objectives of the attention system, as governed by the two parameters identified as the most influential in Section 3.3 — the novelty threshold, and the full-habituation time. The former controls the number of nodes, and hence the level of granularity in the representation used for learning; the latter controls the modulation of learning — how much attention is given to familiar experiences.

5.3.3 Implementation on the Simulated Wall-Following Task

The simulated wall-following task was described in Section 3.2.2, and it was used to demonstrate learning in the previous chapter, and in Section 5.2. The experiment here corresponds to 4A in Figure 5.1. Recall that the effect of social interactions is tested through the effect of 'social facilitation' (see Section 3.1) — the learning performance when the robot only learns from

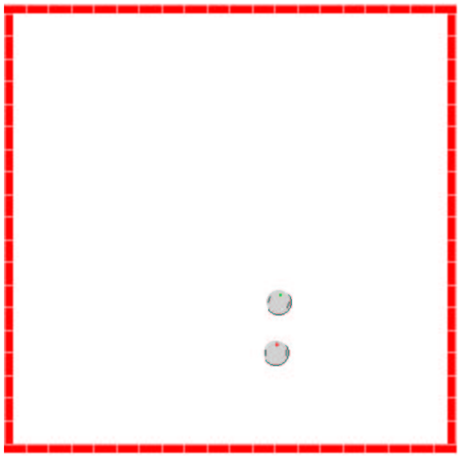


Figure 5.15: The simulated environment used in the wall-following experiments.

demonstrated experiences is compared to when the robot also learns when it loses the teacher ('passive demonstrations' and 'alone & social', respectively, in Figure 5.1). The environment used in the simulated wall-following task is shown again in Figure 5.15. Recall that the learning phase consists of 50000 steps in which the learner follows behind the demonstrator who is following the walls on both sides, with an 'interrupt' signalled every 5000 steps, which forces the demonstrator to turn towards the middle of the environment. The purpose of the interrupt is to ensure that the demonstrator exposes the learner to the full complexity of the task, rather than follow the wall only on one side for the duration of the run.

The interrupt is also used in the recall phase, to avoid the robot following the wall on one side for the duration of the run, and thus not testing the learned behaviour fully. Further, to deal with unsuccessful learning, as in Chapter 4, a built-in 'obstacle-avoidance' behaviour is used to safe-guard the robot, and the evaluation is penalised whenever this occurs. The recall phase consists of 50000 steps, with the interrupt occurring every 7500 steps.

Inverse Model

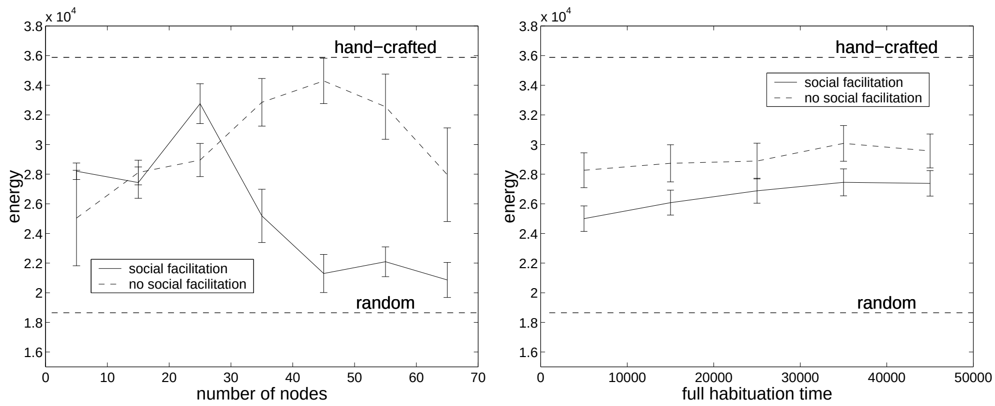
The inverse model should suggest to the robot how to attain a particular *target* state, given the *current* state and the possible actions available to the robot. In this experiment the robot builds the inverse model from experience prior to the start of the experiment, as follows. The robot wanders around the environment on its own, with an added obstacle-avoidance behaviour, and collects information about state-transitions. The action space consists of three actions: straight, left turn, and right turn. The state space is a discretisation of the continuous sensor space of the robot. This discretisation is performed in a rough manner, by grouping continuous instances that are close to each other in the Euclidean space. At the end of this exploration run, the robot has a discrete database of possible states, and a transition matrix is calculated using this database for the number of transitions from a state to any other state, for each of the possible actions (a typical transition matrix is 100×100). The transition matrix effectively corresponds to a Markov Chain, because the next state only depends on the current state, and not on any previous ones.

The inverse model is thus made up of the states database and the transition matrix. It receives as input the current perception of the robot, and a target perception (which comes from the SOFM node); it matches both these instances with the closest available entry in the database, and uses it to find the action that maximises the probability of making the transition. In other words, the inverse model suggests to the robot what action to take in order to sense the wall in a particular configuration, based on what it is currently sensing.

Results

The performance of the system is evaluated numerically, using the same ‘energy’ measure used in Chapter 4 and Section 5.2. Recall that the energy is a measure that the robot acquires from the wall at particular orientations from it, with the most energy acquired when the robot is exactly parallel to the wall. At the end of the recall episode the accumulated energy measures how well the robot performs the wall-following task. The energy is measured systematically as a function of the size of the SOFM network (controlled by the novelty threshold) and the length of full-habituation time, and shown in Figure 5.16(a) and (b), respectively, for cases when the learner follows behind the teacher with and without social facilitation (see Section 3.1). Superimposed on the plots are baselines corresponding to energies acquired by a hand-crafted wall-following behaviour and a random wandering behaviour, as before.

The results are very interesting, because it seems that not only can the robot learn successfully without social facilitation, in contrast to the results in Chapter 4, but in fact with enough SOFM nodes the robot can actually learn better without social facilitation than with it, although only slightly and not very significantly. The reason for this is that the two types of learning in this experiment — the first when the robot acquires the basic sensorimotor skills (the inverse model), and the second when it learns how to use these skills by following the teacher — complement each other very well due to the built-in obstacle avoidance behaviour, as follows.



(a) (b)

Figure 5.16: Evaluation of the recalled behaviour for the simulated experiment, as a function of (a) network size, and (b) full-habituation time. Energy is measured from the sensors at particular configurations from the wall, and compared to energies acquired by a hand-crafted behaviour and a random wandering behaviour.

If the robot is allowed to adapt its SOFM when it loses the teacher (the 'no social facilitation' scenario), because of the obstacle-avoidance behaviour it will experience approaching and turning towards the wall, and assign SOFM nodes to these experiences; if the robot only learns when following behind the teacher (the 'social facilitation' scenario) it does not have these experiences. Then, in the recall phase, the SOFM with the additional nodes is better able, in conjunction with the inverse model, to turn the robot towards the wall and maintain its position parallel to the wall. The perceptual targets specified by the SOFM nodes are attainable by the inverse model, again, due to the obstacle-avoidance behaviour that drives the robot when it acquires the inverse model in the first place. The SOFM without the additional nodes does not have a representation for these experiences, and instead the robot has to use the obstacle-avoidance behaviour, which gets penalised in the recall phase. Further, the robot loses the wall more frequently because it has no representation for sensing the wall as closely.

Of course, this emergent (and unexpected) outcome relies on the fact that the level of granularity in the SOFM is such that the fine differences involving approaching the wall, are deemed salient enough to justify the additional nodes — in contrast to the rougher level of granularity that we have so far been describing for distinguishing between the 'wall' and 'no-wall' perceptions. This requirement for a finer level of granularity is evident in the results, where we can see in Figure 5.16(a) that only when sufficient nodes are used does the 'no social facilitation' scenario start to match and slightly outperform the 'social facilitation' one.

In both scenarios we see that there is a desirable range for how many nodes are used. That is, setting the novelty threshold saliency parameter correctly is crucial for achieving a useful representation in the SOFM for good learning performance. In contrast, we see from Figure 5.16(b) that the habituation parameter does not have such a significant effect, that is, that any amount of modulation works. This means that similar performances can be achieved by considering fewer experiences.

5.3.4 Implementation on the Physical Wall-Following Task

The physical wall-following task was described in Section 3.2.2, and it was used to demonstrate learning in the previous chapter. Here the experiment corresponds to 4B in Figure 5.1. The environment for this task is shown again in Figure 5.17. Recall that in the learning phase the robot follows behind the human demonstrator for 10000 steps, which is approximately 40 minutes of real time; during this run the demonstrator follows the walls on either side, and turns into the middle of the arena approximately 12 times (as with the 'interrupt' in simulation). Recall that due to implementational limitations relating to the human-tracking system, passive demonstrations are not explicitly tested in this environment. Recall also that the demonstrator can explicitly signal to the robot, which is used to draw the robot's attention to particular experiences by forcing the dishabituation of the winning SOFM node. Thus in this experiment, the effect of social interactions is tested by comparing the learning with and without the signals ('explicit signalling' and 'active demonstrations', respectively, in Figure 5.1). Due to hardware and practical limitations, all the information is stored for off-board learning.

The recall phase consists of 6000 steps, which corresponds to around 23 minutes of real time; as in the simulation, an 'interrupt' is signalled at regular intervals — every 1000 steps. The robot is also equipped with a built-in obstacle avoidance behaviour to protect it from unsuccessful learning and also from situations not encountered during the learning phase. For example, when the robot follows behind the demonstrator, it never sees the wall directly in front of it, so it is not expected to know how to handle such a situation in the recall phase, but one does not wish it to drive into the wall! To account for unsuccessful learning the evaluation is penalised whenever the obstacle-avoidance is triggered, as in the simulation experiment.

Figure 5.17: The environment used for the physical wall-following task, where the demonstrator can explicitly signal to the robot.

Inverse Model

Recall that for the simulated experiment the inverse model is acquired by the learner during an exploratory run prior to the social interactions, and it consists of a discretised database of states and transition matrices. From preliminary experiments it was found that in the physical system it is difficult to obtain such a inverse model that is reliable, and this is due to the difficulty we have seen already regarding the poorly-structured sensorimotor data. This problem was overcome by designing a set of innate rules that operate on a small set of states that reliably generalise the robot's state space, and that the robot can use to get from one perceptual state to another.

Results

Recall from Chapter 4 that due to practical time limitations the amount of data available is much smaller here than in the simulation experiment, however it is believed to be sufficient for a reliable evaluation. The dataset obtained in the learning phase was used with different values of the two parameters (novelty threshold and full-habituation time) to obtain various SOFM networks for testing: around 15 different network sizes and 4 different habituation values were used, with each combination run once.

The results are shown in Figure 5.18. We see that overall the results are very good, and this is largely due to the reliable design of the inverse model — it is able to handle well SOFMs of different sizes, as shown in Figure 5.18(a). There is in fact some evidence that very small SOFMs are preferred; that is, very succinct SOFMs provide a more robust representation for

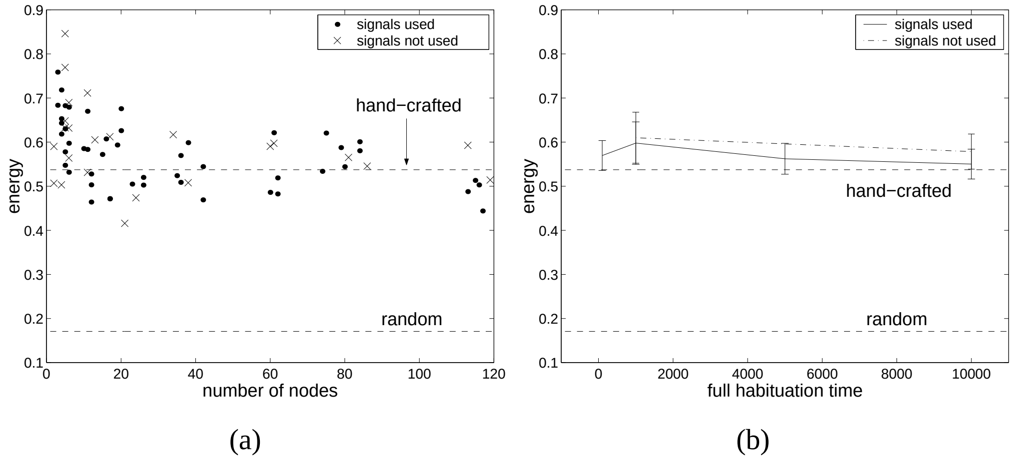


Figure 5.18: Evaluation of the recalled behaviour for the physical experiment, as a function of (a) network size, and (b) full-habituation time. Energy is measured from the sensors at particular configurations from the wall, and compared to energies acquired by a hand-crafted behaviour and a random wandering behaviour. Raw data are plotted in (a), whereas error bars are plotted in (b).

the inverse model, as opposed to the simulation experiments where the inverse model requires more versatility in the representation. We also see that the results are equally good whether the explicit signals from the demonstrator are used or not. These results are in contrast with those in Chapter 4, and the explanation is that because here the learning is at a higher level of abstraction, and it has a very reliable and robust set of existing sensorimotor skills, the role of the explicit signalling from the demonstrator for helping to structure the raw perceptual data is not needed — it does not improve the learning performance.

We see in Figure 5.18(b) that as in the simulation experiment, the role of the habituation parameter is less significant than the novelty detection parameter, which means that similar performances can be achieved by considering fewer experiences.

It is interesting that overall the results are as good as or better than the hand-crafted behaviour. Of course, one could probably design a better hand-crafted behaviour, but the point is that this experiment shows that instead of programming the robot to do wall-following, one can program a set of basic skills that specify the motor commands necessary for a transition from one perceptual state to another, and then demonstrate how to use these skills for wall-following. Whether the second type of programming is easier than the first is of course debatable; we argue that it *is* easier, because one does not have to figure out all the different sensor configurations corresponding to approaching, turning towards, and moving parallel to the wall.

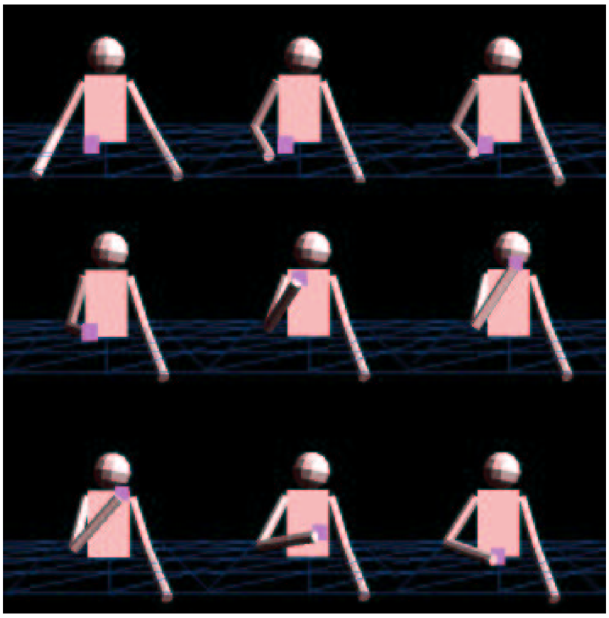


Figure 5.19: A sequence of snapshots of the object-interaction task; left to right, top to bottom.

5.3.5 Implementation on the Object-Interactions Task

The platform for the object-interactions experiment is described in Section 3.2.2. Here only object-interaction 1 is considered, involving approaching the object with the right hand, 'grasping' it, moving it towards the mouth, 'drinking' its hypothetical contents, and then 'putting' it back on the surface, as shown in Figure 5.19. This corresponds to experiment 4C in Figure 5.1. Recall that during the learning phase the imitator passively observes the demonstrator, analysing the visual perception of what the demonstrator is doing; it does not try to replicate the demonstrator's actions. This is the only testing scenario in this experiment, and it corresponds to the 'passive demonstrations' category in Figure 5.1; the discussion in Chapter 6 will suggest other forms of social interactions for this experiment. The learning phase consists of 20 demonstration episodes (approximately 3000 steps each). In the recall phase, the demonstrator performs the object-interaction again, but now the imitator tries to match this behaviour; the interaction is repeated three times (again approximately 3000 steps each).

Inverse Model

The inverse model in this experiment consists of two components, one to control the posture of the robot, the second to control the actual interaction with the object. The first component is a PID controller: postural targets are passed into the PID controller which, together with proprioceptive feedback, calculates the torque (or motor commands) for each limb. The second component consists of a set of boundary conditions for object-interactions, which specify when the wrist is close enough to pick up the glass, when the wrist/glass is close enough to the mouth to ‘drink’, and when the wrist/glass is close enough to the table to put down the glass (these boundary conditions are set to a radius of approximately 4 cm from the centre of the glass, mouth, and table); for example the glass will be ‘grasped’, *i.e.* attached to the robot’s wrist, if the motor targets require this to happen, *and* if the wrist is close enough to the glass. Note that both components of the inverse model are innate and fixed.

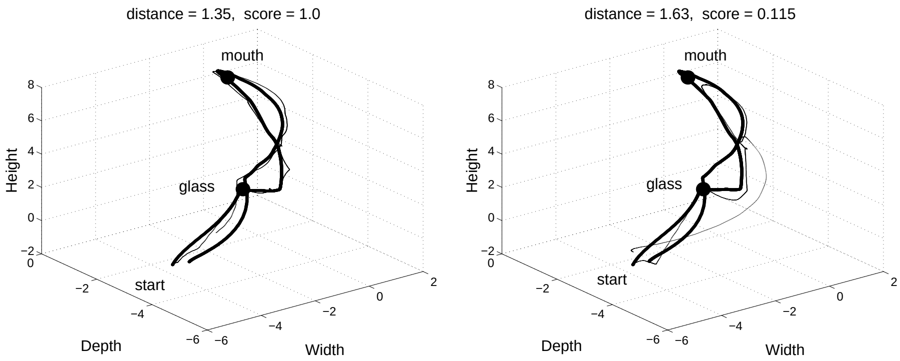
Results

In this experiment two evaluation measures are used in the recall phase: a ‘distance’ measure, which calculates the position of the wrist over time relative to the position of the demonstrator’s wrist, and a ‘score’ for successful execution of the task (*i.e.* picking glass up, drinking, putting down). The measures will be described in more detail below.

Figure 5.20 shows the trajectories of the right-hand wrists of both the demonstrator (thick line) and the imitator (thin line) in a single episode in the recall phase. Notice that Figure 5.20(a) shows a successfully learned behaviour, *i.e.* the trajectories are close to each other, whereas in Figure 5.20(b) the distance between the trajectories is much greater. The distance between the two trajectories is measured by calculating the modulus of the distance between them at each time-step (this simple calculation does not take into account the time-lag between imitator and demonstrator, however the distance has also been calculated using a short-term memory window, and the results were similar).

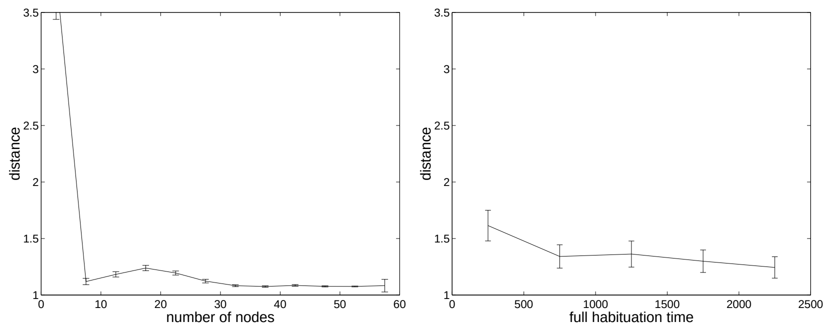
The distance is evaluated systematically using 22 different novelty threshold values (ranging non-uniformly from 0.7 to 0.93, which result in networks of sizes varying from 5 to 60), and 10 different full-habituation times (ranging uniformly from 250 to 2500; recall that the full length of one episode is 3000 steps); each possible combination of these parameters is repeated 5 times. Figure 5.21 shows the distance measure as a function of SOFM network size and full-habituation time. We can see that as long as more than five nodes are used, the performances are equally good independently of the number of nodes and full-habituation time.

Note that the distance calculated is only a measure of the form of the movement; it does not measure how successful the imitator is in achieving the task. In fact it is possible that the trajectory of the imitator is close to the demonstrator’s, but the imitator fails to to pick up the glass etc. This can sometimes happen for small SOFM networks, where the trajectory is good,



(a) (b)

Figure 5.20: Evaluation of the recalled behaviour: the trajectories of the right hand wrists of the demonstrator (thick line) and the imitator (thin line) in a single episode in the recall phase. (a) Trajectories of a successfully learned behaviour; (b) a less successful one. The black spheres on the plots denote the task subgoals, *i.e.* glass and mouth, and their radii denote the corresponding error margin; also indicated above each plot are numerical measures of these behaviours.

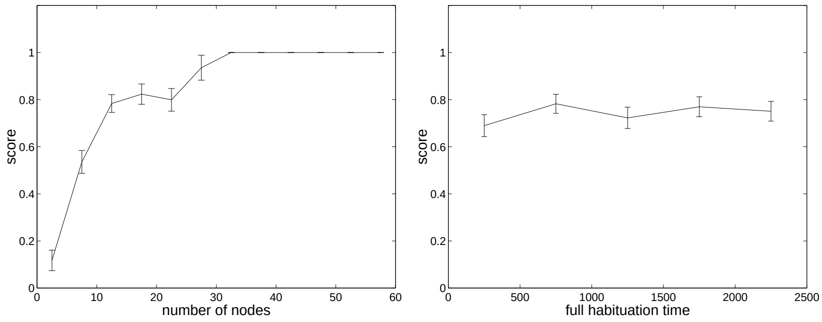


(a) (b)

Figure 5.21: Evaluation of the recalled behaviour, as a function of (a) network size, and (b) full-habituation time. The Euclidean distance between the right-hand wrists of the demonstrator and the imitator is an approximated measure of the form of the movement.

but because there are not many nodes the imitator is actually 'cutting corners' and missing the glass, or missing the mouth, etc.

To test the ability of the recalled behaviour to successfully achieve the task, the second measure was devised to score the behaviour. The imitator can get scores by achieving any combination of the following three goals: (A) picking up the glass, (B) 'drinking' from the glass, and (C) putting the glass back on the table; a bonus is given if all three goals are achieved,



(a) (b)

Figure 5.22: Score obtained at the recall phase, as a function of (a) network size, and (b) full-habituation time. The score is a measure of how well the imitator achieves the task.

which corresponds to a perfect execution of the task. The score is calculated similarly for the demonstrator, and is used to scale the imitator's score, because the imitator can only perform as well as the demonstrator, who occasionally fails in parts of the task (due to noise); therefore the maximum possible score for the imitator is 1. Note that for each of these goals to succeed there must exist a margin of error, *i.e.* a region around the goal (glass, mouth, table) within which the goal is met (*e.g.* a region surrounding the glass where it is allowed to be grasped). This is set as a radius of approximately 4 cm from the centre of gravity of the glass, mouth, or original glass-position on the table (the black spheres in Figure 5.20).

The scores obtained are shown in Figure 5.22, again, as a function of SOFM network size and full-habituation time. We see that while the full-habituation time is insignificant for this measure too, the network size is not. Figure 5.22(a) confirms the suggestion above that smaller networks (size 5–10) that perform well according to the distance measure (see Figure 5.21(a)), in fact do not perform the task well, while larger networks perform well overall (*i.e.* they don't simply copy the form of the movement), and in fact large enough networks obtain a perfect score. This experiment provides an illustration where performance improves as more SOFM nodes are used (up to some point). A limited resources argument might apply here if the computation and/or memory resources needed for handling this learning setup cannot deal with too many nodes, and so one might have to settle for a less than perfect performance.

5.3.6 Conclusion

This section presented a developmental approach to learning at a high level of abstraction, where abstracted perceptual experiences (given by SOFM nodes) are used to trigger existing basic sensorimotor skills that are also at a suitably high level of abstraction. Therefore, as well as designer effort required for abstracting the perceptual experiences by setting saliency parameters, this approach requires the most *additional* designer effort compared to the other experiments in the previous two sections, because the designer must either equip the robot with a reliable set of sensorimotor skills (Sections 5.3.4 and 5.3.5), or bootstrap additional learning with which the robot can acquire these skills (Section 5.3.3).

The fact that perceptual information *alone* is sufficient to trigger appropriate sensorimotor skills is due to the fact that the tasks were all perceptual in nature. Learning the wall-following tasks corresponds only to knowing what are the necessary perceptions; there is no need to know, for example, how long to follow walls, or how fast to move. In the object-interactions task, the actual pick-up action is only virtual, due to software limitation: as long as the wrist is close enough to the object, it automatically attaches itself to it; the object does not have a weight, and no complicated manipulations are required. This setup makes the object-interaction task a 'perceptual' task, like the wall-following tasks, because the robot learns it by 'seeing' the teacher's posture and the location of the object, and this information is sufficient for triggering the learner's own skills. If the full physical properties of the object were modelled, and the learner had different skills to apply different forces to different objects, then in order to learn which skill was appropriate for the task, the learner would need to know the proprioception of the teacher, which would be very difficult, if at all possible, to get just from observing the teacher. In future work the implementation will be extended to apply to other types of tasks that are not purely perceptual. The motor schemas shown in Figure 5.14 would then represent the non-perceptual information that would be used to trigger the inverse model.

The tradeoff of the increase in designer effort mentioned above is that the learning setup is very reliable, and the robot relies less on the social interactions, compared to the other experiments in this chapter, and the experiments in Chapter 4. This is the important conclusion from the experiments in this section, summarised as '4' in Figure 5.1. In fact, we even saw in Section 5.3.3 a desirable side-effect of the imprecise teacher-following behaviour, whereby the learner benefits from learning when it loses the teacher. In the next chapter, we will discuss that this does not go against the argument in this thesis of using social interactions of increasing complexities, and that it actually supports it. The contributions of the object-interaction experiments (Section 5.3.5) are less clear than the wall-following experiments, because the learning of the object-interaction task was not demonstrated with any other approach, or with varying complexities of social interactions. They do however contribute, as we saw in Chapter 3, and as will be discussed further in the next chapter.

Chapter 6

Discussion

The aim of this thesis was to show that the amount of effort required by the designer in biasing the robot's learning of a task can be balanced by the amount of effort from an expert in influencing the learning during the social interactions (see Figure 1.1). Chapter 3 provided an empirical investigation of the robot's experiences as a backing for the claim that motivates this aim, namely that learning does indeed need to be biased by an external source. It showed that the experiences the robot is exposed to are imprecise, and that saliency — the level of granularity at which significant differences occur — depends on the particular task that the robot is supposed to learn in the particular environment.

The aim of the thesis was addressed not only by considering different amounts of effort from the designer and the expert and comparing their effect on the robot's *ability* to learn the task, but also by comparing the learning *performance*. Therefore, as will be discussed in this chapter, the results from this thesis show that balancing effort is beneficial not only for achieving more generality in the learning setup and reducing designer effort, but also for maintaining performance. This is depicted schematically in Figure 6.1 which shows how a typical performance might vary, and that a particular level of performance can be maintained by balancing designer and expert effort.

Section 6.1 discusses how performance is used to characterise the results in the design space identified in this thesis. The results are then interpreted using this characterisation in Section 6.2, and concluded in Section 6.3. The related work presented in Section 2.3 is then revisited in Section 6.4 and characterised in terms of performance in light of the results in this thesis. Section 6.5 provides the overall conclusions from the thesis.

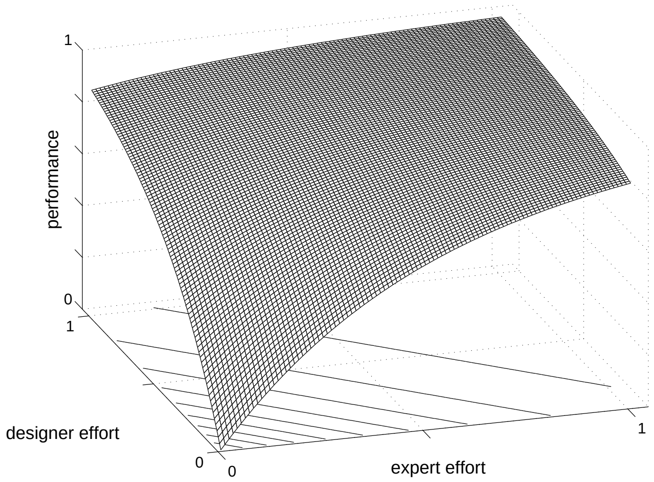


Figure 6.1: This performance surface characterises learning performance within the design space identified in this thesis. The surface shows that performance can be improved by increasing designer effort or expert effort, that at some point the performance levels out, and that a particular level of performance (see projections on the horizontal plane) can be maintained by balancing these two kinds of effort — that is, compensating one for the other. The surface is not intended to depict any features beyond those mentioned, such as the actual functions that describe the increase in performance or how a particular level of performance is maintained by balancing the two types of efforts, and how the performance levels out. Therefore the surface illustrates general performance tendencies — the actual shape chosen here is arbitrary.

6.1 Learning Performance

Figure 6.1 provides a qualitative depiction of how performance varies in the space identified in this thesis, as suggested by the results. It is intended as a general picture of how different systems behave under the different design issues addressed by the two dimensions of the space. Although in the figure performance varies in the same way as a function of each dimension, and the relationship between these two dimensions is linear, the figure is not meant to imply these features. The actual functions are not important, as long as the performance as a function of each dimension is increasing, the performance levels out in each dimension, and one can maintain a particular level of performance by compensating for some decrease in one dimension by some increase in the other. Such a demonstration is very difficult to provide quantitatively because different systems use different performance measures and learn different things, as will be shown in Section 6.4, and this is true also for the experiments reported in this thesis.

In some systems performance is measured in terms of the number of learning steps, and then a comparison between systems is only possible if the learning is implemented on similar robotic platforms with similar control architectures, where a learning step involves similar computational resources. Perhaps a more objective measure is time (CPU or real) for learning convergence, but again, the criteria used to measure satisfactory learning might be subjective to the platform and architecture. With these kinds of measures, performance can be shown to improve as a decrease in the measures, that is, fewer learning steps or less time. Other types of learning performance measures correspond to the correctness of the output of the learning architecture. However, this requires having a source of 'correct' output for evaluating the learned output (such as a set of labelled examples), which might be difficult to obtain. Further, this set of outputs makes the evaluation subjective to the particular experiment. With these kinds of measures, performance can be shown to improve as an increase in the correctness of the learned output.

One can measure learning performance completely independently of the learning architecture, by evaluating it *behaviourally*. This corresponds to a measure of how well the robot executes the learned task, regardless of the time taken for the learning architecture to learn, or of the actual values it has learned. Then the same measure can be used to compare the same task implemented using different architectures, perhaps even by different researchers. The measure can evaluate how closely a learner's behaviour matches the expert's, or how well it achieves the purposes of the task using criteria imposed by the experimenter. Researchers often report qualitatively that their system is able to learn and execute different tasks, but quantitative measures could also be devised, such as the 'energy' measure used in the experiments in this thesis. Using such measures one can more easily infer a pattern of learning performance across different experiments, specifically where different design issues are addressed and where other measures are therefore more difficult to use.

The experiments in this thesis have all been evaluated using such behavioural measures, so that a performance surface such as the one shown in Figure 6.1 could be extrapolated and generalised from the results. The *actual* results are summarised in Figure 6.2. Although the different experiments involve different learning architectures and different tasks, and where different types of of social interactions and designer effort are attempted, the overall interpretation of the results is generalised as the surface shown in Figure 6.1, with the aim of showing that performance can be improved by moving along either of the design dimensions, and that with enough designer and expert effort the performance levels out.

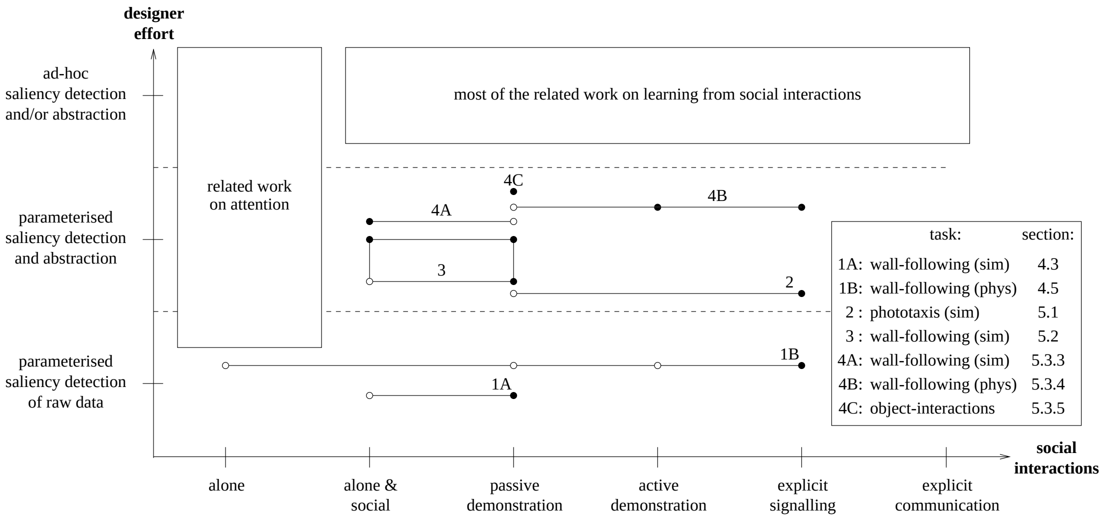


Figure 6.2: The experiments reported in the thesis examine social interactions of different complexities, and learning at different levels of abstractions requiring different amounts of designer effort. The experiments that are actually performed are marked with circles, with a solid circle marking the best performance achieved. Within the second category on the vertical axis, the vertical distinction relates to different amount of effort required by the designer, except that the distinctions within experiment set 4 are purely for visual purposes. Similarly, the vertical distinctions within experiment set 1 are for visual purposes only.

6.2 Summary of the Results

In this section, the results from Chapters 4 and 5 are summarised and compared. A graphical summary of these results was referred to throughout the thesis, and it is shown again in Figure 6.2. The results from Chapters 4 and 5 refer directly to learning performances of different learning approaches; four different learning approaches were presented, and they are labelled in Figure 6.2 as follows:

1. MLP approach (Chapter 4)

2. Hopfield approach (Section 5.1)

3. gaussian-fitting approach (Section 5.2)

4. developmental approach (Section 5.3)

In the figure, the best performance achieved in each experiment is shown with a black circle, and a line connects it with the other results from the same experiment. The figure is intended to depict 'performance' as a third dimension, with two values, marking the best performance or otherwise. Within each learning approach comparisons were made in Chapters 4 and 5 between different learning scenarios, mainly concerning different forms of social interactions; approach 3 (Section 5.2) also compared different amounts of designer effort. All these different comparisons will be summarised again here, and comparisons will also be made *between* the different learning approaches.

As mentioned above, comparisons are easier to make between experiments involving similar implementational platforms, and where the same task is involved. Therefore, the discussion of the results below is organised according to the task and platform. Within each section the discussion will show the implications of less designer effort for the benefit of stronger social interactions, and so the experiments will be discussed in decreasing order of designer effort. The discussion will also highlight which of the social interactions are addressed in each task, how they contribute to the particular part of the design space, and the connection to other parts addressed by other experiments.

6.2.1 Simulated Wall-Following Experiments

The experiments involving the simulated wall-following task (1A, 3, and 4A in Figure 6.2) address the second source of imprecise exposure to sensorimotor data mentioned in Section 3.1, arising from the learner's exposure when it loses the teacher. By doing so, they start to demonstrate the benefit of stronger social interactions (moving from the 'alone & social' category to the 'passive demonstration' category in Figure 6.2), and how this is affected by the designer effort. They also suggest the implications for increasing the complexity in the social interactions further.

The developmental approach (4A in Figure 6.2) requires the most designer effort, because the designer must bootstrap additional learning, prior to the social interactions, such that a useful and reliable set of basic sensorimotor skills is obtained. With this setup, there is no problem with the learner occasionally losing the teacher during the social interactions and learning on its own, that is, there is no decrease in performance. The results in fact show that there is a slight increase, though not very significant (compare peaks between the 'social facilitation' and 'no social facilitation' scenarios in Figure 5.16) — this will be discussed below. In the performance surface (Figure 6.1), the lack of decrease in performance corresponds to being at the extreme end of designer effort, where decreases in the expert's effort have little significance. The reason the performance does not decrease is related to the fact that the learning does not involve low-level motor data. Recall that it is the abstraction of the *perceptual* data that triggers the pre-learned sensorimotor skills. If the designer is careful in setting the saliency parameter (see performance for the ‘no social facilitation’ scenario with > 30 nodes, in Figure 5.16), this abstraction is robust enough to deal with the cases when the robot gets lost, and is thus able to correctly trigger the reliable pre-learned sensorimotor skills. This robustness of the perceptual abstraction is related to the level of granularity in the representation of the perceptual data. When the learner loses the teacher it meets the wall straight ahead at different angles — variations in the perceptual data occur at a fine level of granularity; the saliency of these variations, and hence their effect on the abstraction of the perceptual data, depend on how sensitive the saliency parameters are.

In the gaussian-fitting approach (3 in Figure 6.2) the learning involves the raw motor data, so in that respect there is less designer effort. We can see on the performance surface (Figure 6.1) that this means that there is the possibility of improving the performance through stronger social interactions. The results indeed show that the stronger form of social interactions improves the learning performance — generalising from the raw motor data when the learner loses the teacher is undesirable. However we saw that the designer can impose an *additional* bias (depicted by vertical lines in Figure 6.2) to deal with the undesirable learning when the learner is lost, but this learning performance is matched by learning with the stronger form of social interactions (see Figure 5.13(a)). This corresponds to a location on the performance surface where it levels out. That is, not only can the performance be improved to the same level by either increasing the designer effort or the social interactions in dealing with imprecise exposure, but also no further improvement is achieved by increasing *both* types of effort.

In the MLP approach (1A in Figure 6.2) the learning is at a low level of abstraction, therefore the least amount of effort is given by the designer in abstracting both the perceptual and motor data. Here the stronger form of social interactions is crucial for correct learning (see Figure 4.9(b)), and this corresponds to being at the low extreme of designer effort on the performance surface, where the social interactions have the most significant effect. The responsibility for usefully biasing the robot’s learning is transferred from the designer to the expert. We saw that in fact the robot can learn without *any* designer effort in abstracting the data, because raw data can be considered all the time without any modulation, as long as there are enough learning resources to do so (see Figure 4.11). Further, the performance in the case without any designer effort is comparable to the other learning approaches that do involve designer effort. This corresponds to a different performance surface than the one shown in Figure 6.1, such as

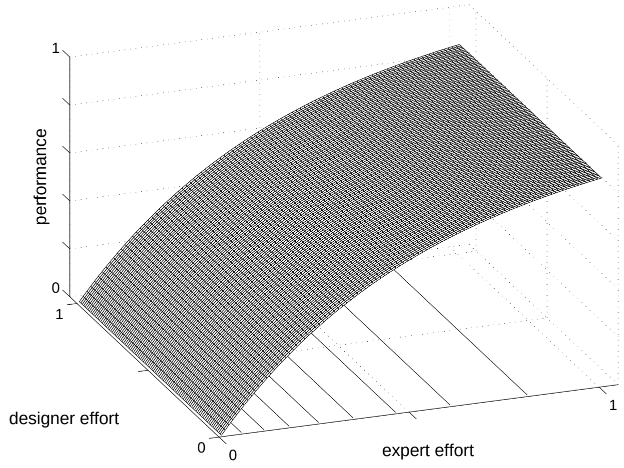


Figure 6.3: The performance surface when there are unlimited learning resources. The experiments in Chapter 4 show that learning from the raw data without any abstraction from the designer yields desirable results when there are unlimited learning resources, suggesting this kind of performance surface where increasing designer effort does not significantly improve the results. As for Figure 6.1, the actual shape shown here for the performance surface is arbitrary — only general features are depicted here, showing that performance improves as a function of expert effort, and that at some point it levels out, whereas designer effort does not improve the performance.

the one shown in Figure 6.3, where increasing designer effort does not significantly improve the performance. This is an example of how the performance surface is affected by available resources. This is discussed further in Section 6.4.

As mentioned above, some of the results from the developmental approach do not seem to support the general performance behaviour depicted in Figure 6.1. The discussion above described why the performance does not significantly worsen with the weaker form of social interactions because the learning can be complemented with some 'self-learning'. But in fact the results show that the performance can actually be slightly improved when the social interactions are complemented with some self-learning. When the level of granularity is fine enough to represent the robot's perceptual experiences when it is on its own, as discussed above, these experiences just happen to complement the experiences obtained from the social interactions. The robot experiences turning to and from the wall at different angles. This was not expected from the robot for learning this task, and in fact, the other learning approaches do not accommodate these finer experiences and therefore cannot capitalise on them.

However, we argue that this result *does* support increasing the social interactions through the utilisation of active demonstrations, and therefore that it fits with the description given by the performance surface in Figure 6.1. It was only a side-effect that the learner's built-in behaviour for moving on its own provided it with useful experiences — this was not planned. In contrast, an active demonstrator could *ensure* that the robot has these kind of experiences, for example by manipulating its movements such that it does not 'cut corners' as the learner usually does when following the demonstrator. This addresses the first source of imprecise exposure to sensorimotor data mentioned in Section 3.1, whereby the learner does not follow the demonstrator's path exactly. The results from the developmental approach therefore provide an even stronger support for increasing the complexity of the social interactions. That is, they show that for the more complicated learning task, the social interactions could have a more significant effect in improving the results, and this would also be evident in the other learning approaches if their complexities were increased appropriately.

6.2.2 Physical Wall-Following Experiments

The experiments involving the physical wall-following task (1B and 4B in Figure 6.2) address higher complexities in the social interactions than the simulation counterparts, firstly due to implementational limitations involving the teacher-following tracking system; and secondly due to the noisier, less well-structured nature of the sensorimotor data compared to the simulated data, which became evident in Chapter 3. The physical robot's tracking system was not designed robustly enough to track a human moving freely in an open environment, and rather than spending more time and effort improving it, active demonstrations were introduced instead in order to ensure the human does not fall out of the robot's field of view. Also, an ideal teacher-following scenario was emulated by evaluating the learning when the robot is exposed to experiences through a hand-crafted wall-following behaviour. Explicit signalling by the demonstrator, whereby the demonstrator waves his hand, was introduced in order to help structure the robot's sensorimotor data. Therefore, these experiments demonstrate the benefit of increasing the complexity of the social interactions further than the simulation experiments (moving further along the horizontal axis from the 'passive demonstration' category in Figure 6.2), and how this is affected by the designer effort.

The physical wall-following task was used to test the developmental and MLP approaches. As with the simulation experiments, the developmental approach in the physical wall-following experiment (4B in Figure 6.2) involves more designer effort compared with the MLP approach, and therefore the social interactions have a less significant effect on the performance, as suggested by the performance surface. Much of this effort is in providing the basic sensorimotor skills, which, unlike in the simulation experiment, are programmed by the designer. This involves a reliable abstraction of both the motor and perceptual spaces, and because the resulting skills are reliable and robust, the abstraction of the perceptual data during learning of the actual task, is less crucial than in the simulation — SOFMs of different sizes all produce a good performance (see Figure 5.18(a)). The explicit signalling from the demonstrator is not needed for structuring the perceptual data for learning the task, because of the abstraction provided by the SOFM and due to the reliability of the sensorimotor skills.

In contrast, the MLP approach (1B in Figure 6.2) *does* benefit from the explicit signalling, because here the learning occurs on the raw perceptual data, and there are no basic sensorimotor skills; here the abstraction of the motor data to motor actions occurs in the recall phase, through the designer setting a saliency threshold on the output of the learning system. Thus compared to the developmental approach, there is little designer effort in abstracting the motor data. As in the simulation experiments, there is also little or no designer effort in abstracting the perceptual data: when the explicit signals are not used, the best performances are achieved when there is little or no modulation (see the ‘signals not used’ scenario in Figure 4.15); when the explicit signals *are* used, the best performances are achieved when the signals start to dominate over the attention system in triggering (and hence modulating) the learning system (see the ‘signals used’ scenario in Figure 4.15).

In fact, the performances achieved with explicit signalling are significantly better than without (see Figures 4.15 and 4.16). Thus, the designer effort relating to setting saliency parameters is not only balanced by the expert’s signalling of saliency, but is actually outperformed. Also, the MLP approach showed the benefit of active demonstrations, firstly by showing the benefit of precise exposure to experiences through an emulation of an ideal teacher-following behaviour (see the ‘hand-crafted’ scenario in Figure 4.15), and secondly by showing that the expert’s demonstrations are favourable to this emulation, even when the explicit signals are not so dominant (see the ‘signals used’ scenario with low full-habituation times in Figure 4.15). The noisy and unstructured nature of the physical robot’s sensorimotor data is handled by the expert who accentuates the salient differences between the parts of the task. The fact that the expert ‘exaggerates’ these differences means that the designer can set the saliency parameters less precisely. Thus we see overall from these results that increasing the expert’s effort by using active demonstrations can balance designer effort and maintain performance, while increasing the expert's effort further through explicit signalling improves the performance. Hence, as in the simulation, at this low level of abstraction, where there is little designer effort compared to the developmental approach, the responsibility of usefully biasing the robot's learning is transferred from the designer to the expert.

However, the results from the developmental approach seem to be better overall than the MLP approach. The tradeoff of more designer effort is that the 'data' available to the robot for learning and executing the task are more reliable and robust. As discussed in Chapter 2, one must however question the generality of such designer effort. Another possible explanation for the favourable results was given above for the simulation experiments, concerning the fact that the learning architecture in the developmental approach is more suited to learning and dealing with the finer parts of the task, which the MLP approach was not designed to deal with. It was suggested at the end of Chapter 4 how the MLP learning architecture could be extended to deal with these parts of the task, namely turning towards the wall, by increasing the complexity of the architecture by adding output units to the neural network.

There is an overall conclusion from the simulated and physical wall-following tasks, with regards to scalability. Although simple learning setups were devised, where the robot was not expected to learn how to turn towards the wall, we see various suggestions that the conclusions can extend to deal with more complicated learning setups, such as turning towards the wall. Specifically, to deal with such an increase in learning complexity, we could replace the simpler MLP approach with the developmental approach, or introduce more learning units in the MLP approach, and in both increase the designer effort; also, we could increase the influence from the social interactions by ensuring that active demonstrations expose the learner to the appropriate experiences necessary for learning to turn towards the wall, as was suggested by the results of the developmental approach in the simulation. The experiments in this thesis also suggest that the conclusions extend to even more complicated tasks, such as ones involving interactions with objects, as discussed below. Thus dealing with more complicated learning scenarios simply means increasing the level of effort from the designer and the expert, in order to reach a desirable level of performance.

6.2.3 The Remaining Experiments

Although the Hopfield approach (2 in Figure 6.2) was implemented on a different task and used a different attention system from the one used in the wall-following experiments, its aim was to address another part of the performance landscape, showing the implications of designer effort on social interactions. It showed that at this high level of abstraction, although the designer effort in abstracting the perceptual data is considerable compared to learning at the lower level of abstraction, the designer effort in abstracting the motor data is comparably low. Therefore, this corresponds to a point in the performance surface where a significant increase can be achieved through stronger social interactions. The results show that the strongest form of social interactions, namely explicit signalling, provides a significant improvement to the learning performance (see Figures 5.7 and 5.8). As mentioned in Section 6.1, comparing performances across implementations where different tasks are involved is difficult, however we can draw the following *qualitative* conclusion: compared to the other approaches for learning at a high level of abstraction, the reduction in designer effort in this approach is balanced by an increase in the complexity in the social interactions.

Similarly, comparing the numerical results of the object-interaction experiments (4C in Figure 6.2) with any of the other experiments is not straightforward. In fact, there are no comparisons to make, because these experiments did not test different levels of social interactions or different types of designer effort. It is worth mentioning however that a source of imprecise exposure to sensorimotor data in fact exists also in these experiments. Visual inspections of the demonstrations showed that the demonstrator occasionally failed to put the object back down on the table, due to an imperfect programming of the demonstrator's actions. This provides undesirable experiences to the learner, and is equivalent to the learner losing the demonstrator in the mobile robot experiments. This issue was not tested explicitly as in the wall-following experiments, but from the results it seems that the learning approach was able to generalise well in spite of these undesirable experiences. A reasonable explanation for this is related to the large amount of designer effort in abstracting the perceptual data and providing the basic sensorimotor skills, as discussed above.

Here the basic sensorimotor skills were provided to the robot by the designer as two components. The first was a PID controller which tells the robot how to achieve a particular posture — this component could be argued to be general purpose and task-independent. However, this certainly cannot be argued for the second component, which deals with the actual interaction with the object — it specifies the boundary conditions that are necessary for grasping the object, 'drinking' from it, and putting it down on the table; these specifications depend on the actual object and the purpose of interacting with it. If instead these boundary conditions were controlled by a saliency parameter that specified, for example, when the hand is deemed close enough to the object, then the designer effort involved in setting this parameter could be balanced by more effort from the expert, through active demonstrations for accentuating the differences between being close enough to the object or not, and even signalling when the hand is close enough. Thus, by reducing the designer effort, it is expected that an effect would be observed in the performance surface when modifying the social interactions.

The object-interaction experiments were useful for showing that the notion of saliency can be different for different tasks, because the level of granularity at which salient differences occur depends on the complexity of the task — Section 3.2.2 presented four different object-interaction tasks, with a varying degree of difference between them. The results related to learning one of these tasks in Section 5.3.5 showed that the learning performance depends on a desirable representation in the abstraction of the data, with a sufficient number of SOFM nodes (see Figure 5.22(a)). One would expect that the learning of one of the more complicated object-interactions presented in Section 3.2.2 would require the representation to be at a finer level of granularity, and therefore more SOFM nodes to represent it. The fact that performance depends on a useful abstraction, which is currently the responsibility of the designer, suggests that designer effort could be reduced by transferring this responsibility to the expert, perhaps by signalling the salient postures.

Therefore although the object-interaction experiments were not tested as extensively as the other experiments, they nevertheless contribute to this thesis. They are important for demonstrating that the arguments and ideas in this thesis apply to different kinds of robots and tasks, not just the kind of mobile robot experiments such as wall-following and phototaxis. Namely, the object-interaction experiments addressed issues of levels of granularity, importance of saliency bias, implications for limited learning resources, imprecise exposure to sensorimotor data suggesting more active demonstrations, active demonstrations for accentuating saliency, and explicit signalling and communication for dealing directly with saliency. While the consideration of how these various issues apply to the object-interactions experiments provides a valuable contribution to this thesis, testing the issues explicitly in future work would strengthen the conclusions and thus provide an even greater contribution. These issues will be discussed further in Section 6.4.

6.3 Overall Conclusions from the Results

A summary of the results was shown in Figure 6.2, where direct comparisons are made within each experiment based on quantitative differences. An overall summary is extrapolated from this figure to reflect the qualitative differences between all the experiments. This produces a performance surface (Figure 6.1) for characterising the work reported here, and, as will be shown in the next section, for characterising related work. The qualitative and quantitative differences between the experiments were summarised and discussed in this section. The important points are listed below:

- Designer effort can be balanced with more effort from an expert during social interactions. More generally, a particular level of performance can be achieved by increasing one type of effort and decreasing the other, as shown by the projections on the horizontal plane in Figure 6.1.

- Performance can be improved by increasing either of the two types of effort, and it is possible to achieve the best performance by increasing one type of effort and not the other. This means that the performance surface shown in Figure 6.1 levels out.

- The points where performance levels out depend on the available learning resources. Specifically, when there are unlimited resources, a comparable performance can be achieved with very little designer effort in abstracting from the raw sensorimotor data, thus eliminating the significant increase in performance due to designer effort. In principle, a similar elimination could be observed from the effect of social interactions, as will be discussed in the next section.

- The points where performance levels out also depend on other design issues not considered by the characterisation of designer effort in this thesis, such as the design of the robot's morphology and the learning architecture. The experimental discussion suggested that performance could be improved by modifying the learning architecture, to suit the task better, and therefore how the arguably simplistic learning setups used in the experiments might scale up to learn more complicated functions. It was argued that the general shape of the performance surface mentioned above applies regardless of the learning architecture.

- There is a desirable balance between the two types of effort, that is, it is not necessarily desirable to completely replace designer effort. The results suggest that *some* designer effort is desirable for a reliable and robust learning setup that requires fewer learning resources. This will also be demonstrated further in the discussion of the implication of the results to related work, in the following section.

These conclusions are in accordance with the general performance surface shown in Figure 6.1. Of course, the empirical evidence for this surface is limited — it only provides a small set of points from which the surface is extrapolated. Nevertheless, the validity of such a general performance surface will be shown further in the next section by discussing how it applies generally to other systems presented in the literature. The utility of the design space identified in this thesis will thus be strengthened further, because it will be shown that the space can be used not only to organise and describe related work in the literature, but also to provide a methodology — governed by performance — to address the tradeoff that the space conceptualises.

This thesis provides empirical evidence for the *general* tendencies described above and shown in Figure 6.1 regarding learning performance. They do not say anything about the specific *shape* of the performance surface. For example, they do not specify a function that characterises the balance between the two dimensions of the space, and they do not specify how the contribution of each dimensions levels out. It is proposed that the design space provides a framework for summarising future research, and that such research will therefore provide further empirical evidence of the general performance surface suggested here, and even some categorical evidence of specific shapes that the surface can have under different conditions.

6.4 Implications of the Results to Related Work

The implications of the results from this thesis are related to balancing designer effort in biasing the learning by a robot with social interactions, and how this affects the learning performance. These implications for the related work presented in Chapter 2 will now be discussed. In particular, the discussion will show how performance in related work can be improved, and thus address the gap in the research that was identified in Chapter 2, as shown again in Figure 6.4.

Three main implications can be identified:

1. through **active demonstrations**, an expert can firstly ensure that the learner is exposed to desirable experiences, and secondly accentuate the salient differences between the experiences;

2. one can reduce the amount of designer effort involved in abstracting the experiences of the robot and thus forcing a **level of abstraction**, by utilising parameterised, bottom-up, self-organising mechanisms for abstracting experiences, and then using the expert to

influence the abstraction through active demonstrations and more explicitly by signalling saliency;

3. one can completely replace the setting of **saliency parameters** at design time with parameter-tuning through explicit signalling and communication between the learner and expert during the social interactions.

The first two implications follow directly from the results in this thesis, while the third is conjectured; they will be discussed in order.

6.4.1 Active Demonstrations

Mobile Robots

In most of the work involving mobile robots, where a learner robot follows behind a robotic or human teacher, a difficulty arises from the fact that the learner does not follow exactly the same path as the teacher, and it is therefore exposed to imprecise sensorimotor data for learning.

In the work of Hayes and Demiris (1994) this is not a significant problem due to the confined maze environment, in which the paths of the teacher and learner are restricted to be almost identical, and also in which saliency is trivial. Performance is measured here qualitatively as the ability of the robot to learn to navigate different mazes. In the simulation experiments there are very few discrete perceptions and actions, and therefore the learner is guaranteed full exposure to them, as necessary for successfully learning the correct mappings. In the physical experiments, active demonstrations could be used to ensure that the robot is well-exposed to the salient parts of the environment, namely the corners of the maze. We saw in Section 5.3.3 the benefit of the learner sensing the wall closely, and similarly in Demiris's maze environment the teacher might want to ensure the learner is exposed closely to the corners, and that it does not cut corners. The benefit of such active demonstrations would be clearer in more open environments, where the path of the learner is less restricted, and where saliency is less trivial.

This distinction between restricted and open environments is demonstrated nicely in the experiments of Billard and Hayes (1999). Although the demonstrations actually have an active component, with which the teacher controls the path of the learner, the performance in the restricted environment is nevertheless better than in the open environment. Performance is based on the correctness of the correlations learned by the neural network, the convergence of the learning weights, and the size of the learned vocabulary. Compared to the results from the restricted environment, in the open environment there are noisier learned correlations, there

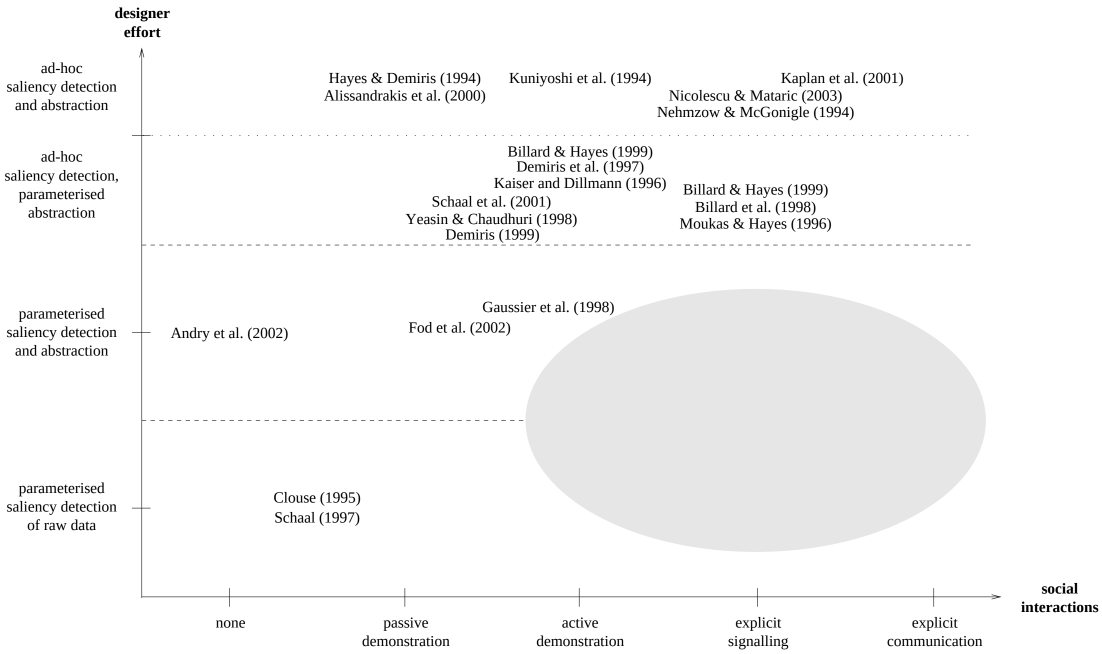


Figure 6.4: The organisation of the literature shown in Figure 2.3 is shown here again. The discussion in this chapter focuses on how these examples in the literature could be re-positioned in the design space towards the gap that is identified in this thesis, marked as the grey region, for the purpose of balancing designer effort with social interactions, and improving learning performance.

is no clear convergence of the weights, and the learned vocabulary is smaller because some perceptions are missed. The first two results suggest that the data are less well-structured, and the third result suggests that the teacher does not ensure full exposure. All these problems could be solved through more active demonstrations, where the teacher exposes the learner to *all* the relevant experiences, and accentuates the salient correlations over the noisy ones through higher exposure, as we saw in the physical wall-following experiments in Section 4.5. The learning will stabilise and converge if the experiences are better structured.

As mentioned in Section 2.3, Gaussier et al. (1998) already recognise a distinction between passive and active demonstrations, and that the latter type improves learning performance. They measure performance as the precision of the imitated trajectory with respect to the teacher's trajectory, in terms of distances and angles. The human teacher adapts his own trajectories so that the learner passes through correct trajectories, and the timings of the learner's actions are more precise. This is similar to the physical wall-following experiments in Section 4.5, where the human demonstrator adapts (exaggerates) his movements so that the robot's actions are clearly distinguished for the different perceptual situations.

In the reinforcement learning experiments by Clouse (1995), the automated trainer can be thought of as providing active demonstrations to the learning agent, because the information exchanged between them is in fact the actions that the learner should take. The performance of the system is measured by the number of trials taken for the learner to teach the goal and the number of actions needed. In this implementation, it is the learner who asks the trainer for advice, at regular intervals, and as mentioned in Section 2.3, it is not beneficial for the learner to ask for advice all the time, because it misses out on valuable negative learning. However, the 'demonstrations' could be made more active if the trainer influenced when advice is given, for example, by giving advice when it (the trainer) believes the learner is going through salient positive experiences which must be distinguished from the negative ones. This would speed up the learning and therefore improve the performance.

Other Robots

The problem of imprecise exposure to sensorimotor data exists also in other types of robotic platforms, not involving mobile robots. In experiments involving teaching a robot how to move its limbs or manipulate objects, the quality of the demonstrations and the robot's ability to copy the demonstrator's actions determine what kind of experiences the robot is exposed to. The human's movements must be segmented usefully.

In the work of Demiris et al. (1997) segmentation is trivial due to the restricted set of movements used — only horizontal rotational head movements are used, where saliency detection trivially corresponds to a change in direction in the horizontal axis. As in Demiris's maze experiments (Hayes and Demiris, 1994), the demonstrator therefore naturally provides active demonstrations due to the nature of the task. However if more complicated movements were to be imitated, for example movements also in the vertical axis, where the differences between parts of the movements are more subtle, and saliency detection is less intuitive, then active demonstrations would be useful for accentuating the differences between the parts of the movements, by providing distinct movements. The performance of this system is evaluated as its ability to learn from demonstrations of varying speeds and durations. The results from the physical experiments in this thesis suggest that active demonstrations could improve the performance of Demiris's system, but this would be more clearly demonstrated if more complicated movements were utilised.

Demiris does in fact use his learning architecture to learn more complicated movements, in the simulated humanoid experiments where a humanoid robot learns various postures from an identical demonstrator (Demiris, 1999). These experiments provide a nice demonstration of the difference between designed and learned behaviours. Performance here is related to how well the robot's behaviours match the observed behaviour. If none of the existing behaviours match the observed behaviour well, a new behaviour is learned; the robot has a set of designed behaviours to start with. The results show that overall the designed behaviours perform better than the learned ones, and Demiris mentions that this is due to the noisy perceptions of the learner. Again, the results from this thesis suggest that this problem could be addressed with active demonstrations for improving the performance, and thus transferring the responsibility of the designer in usefully distinguishing between behaviours prior to learning, to the demonstrator *during* learning.

Note that the object-interaction experiments presented in this thesis were in fact performed using the same simulator as Demiris's experiments, and we saw in Chapter 3 that different behaviours result in different sensorimotor data and are subsequently represented by different SOFMs. It was mentioned in Chapter 3 that these differences occur at different levels of granularity according to their complexity, and that the demonstrator can accentuate these differences by slowing down the demonstration at the critical parts of the task in order to expose the learner to more sensorimotor data, and therefore force the creation of SOFM nodes for these experiences. It is expected that the learning performance in the object-interaction experiments in Section 5.3.5 could be improved, as could Demiris's experiments, by providing active demonstrations, where the demonstrator purposely slows down significantly at the salient parts of the task, and accentuates salient differences by demonstrating distinct trajectories, clearly distinguishable from each other, or even 'exaggerating' the differences between them, as in the physical wall-following experiments, for example by moving the elbow further away to distinguish between object-interaction 1 and object-interaction 2 (see Section 3.2.2).

However, it is argued in this thesis that programming a robotic demonstrator to provide active demonstrations such as described above is very difficult, whereas for human demonstrators this is much more natural and adaptive, especially when the human is situated in the same environment as the robot. Therefore, it is likely that in experiments involving a human demonstrator, such as those reported by Fod et al. (2002), Schaal et al. (2001), and Yeasin and Chaudhuri (1998), the human is (at least) subconsciously providing active demonstrations. If not, then the results from this thesis suggest that learning performance could be improved by demonstrating well-structured, distinct trajectories. In the first two examples above, performance corresponds to how well the simulated or physical robot, respectively, can match the observed movements; in the third example, performance corresponds to how fast the assembly system detects the demonstrations and predicts the components of the corresponding plan. Similarly, in the experiments by Andry et al. (2002), if the imitation by the robotic manipulator of the human movements were used to learn a particular task (recall from Section 2.3 that the robot learns how to move its manipulator to various locations in its visual field in order to imitate a human, but it does not learn anything further), then active demonstrations would be useful for accentuating the relevant sensorimotor skills for the task.

In simulation reinforcement learning systems, where expert advice or demonstration is given, the issue of programming an active expert or demonstrator must inevitably be dealt with, because the learner is given advice or demonstrations which are immediately usable by the system, thus the system is 'exposed' to precise experiences. The example given by Clouse (1995), where advice is provided in the form of the actions that the learner should take was discussed above. In the work by Schaal (1997), the learner is given demonstrations in the form of actual data that simulate the learner's experiences without it having to go through them. As mentioned in Section 2.3, here the system learns from the raw data, and so there is no need to tailor these demonstrations to influence saliency and hence abstraction of the data. However, the demonstrations could be chosen such that the important experiences are highlighted, for example, by giving more examples of the difficult parts of the sensorimotor space, and thus speeding-up the learning. Schaal (1997) mentions that in the cart-and-pole system swinging the pole to the upright position is relatively easy, whereas keeping it balanced is the difficult part. Active demonstrations could ensure sufficient examples of the more difficult parts of the task.

The implications of active demonstrations for the related work presented above are clear. We argue that active demonstrations should be useful for all robotic learning systems that learn from social interactions. However, in some cases the implications of active demonstrations are less clear, because the perceptions of the robot are strongly controlled due to the designer forcing a high level of abstraction, which therefore also means that saliency is trivial, as discussed in Section 2.2.2. As discussed in Section 6.2, this corresponds to being at the extreme end of designer effort in the performance surface (see Figure 6.1), where increasing the social interactions has little significance. Some examples will be discussed in the next section.

6.4.2 Level of Abstraction

The results from this thesis show that through explicit signalling, social interactions can be used to structure the robot's experience usefully for learning and therefore start at a relatively low level of abstraction, rather than force a high level of abstraction at design time. This means that the expert involved in the social interactions can influence the resulting level of granularity from which the robot learns. For example, in the experiments by Billard and Hayes (1999), the teacher sends signals to the learner to associate with its perceptions so that it learns a vocabulary of its experiences. If these signals were also used as a spotlight to highlight the salient experiences and favour the learning of their correlations, as with the signals used in Section 4.5 and the stimulus enhancement used in Section 5.1, then the learning performance would be improved. Similarly, in the experiments of Gaussier et al. (1998), the robot's movements could be more easily segmented if the teacher signalled when salient changes between the edges of the trajectories occur. The same is true for all other systems where segmentation of continuous movements is involved (Moukas and Hayes, 1996; Schaal et al., 2001; Fod et al., 2002; Kaiser and Dillmann, 1996; Yeasin and Chaudhuri, 1998).

Demiris (1999) actually recognises the issue of different levels of granularity for different task complexities. The demonstrations consist either of short movements corresponding to letters of a semaphore code alphabet, or to longer movements corresponding to a sequence of letters, or words. However, whatever is demonstrated, the learner tries to match it with a complete behaviour, that is, the learner cannot break down a complete demonstration of a word into its constituent parts, or letters. So the learner has separate unconnected representations of simple behaviours (letters) and composite ones (words), which it tries to match with the observed behaviours. Knowing the level of granularity of what is being demonstrated would be useful for improving the ability of the learner to match, categorise, and learn new behaviours. With explicit signalling the demonstrator could signal the salient events and hence the level of granularity at which they occur. As mentioned in Section 2.3, Gaussier et al. (1998) and Alissandrakis et al. (2000) explicitly model different levels of granularity, and again, the expert could influence which of the different levels is suitable, which might vary between tasks or even during one particular task.

As mentioned above for active demonstrations, the implications of explicit signalling for related work is only clear where the level of granularity is not rigidly forced at design time. We now refer to cases where increasing the complexity of the social interactions to significantly improve the performance can only be demonstrated if we consider how these systems perform with less designer effort in forcing a high level of abstraction.

Moukas and Hayes (1996) utilise explicit signalling to signal the start and end of trajectories. However, the level of granularity within the trajectories is pre-determined through the number of times data are sampled for each trajectory, and the number of nodes used by the Kohonen map to self-organise these data. They measure performance through correct predictions of the feed-forward neural network, and the speed of convergence of its weights. The combination of pre-defined level of granularity and the segmentation provided by the teacher's signals provide the neural network with well-structured data for good learning. In order to reduce designer effort in forcing the level of granularity, one could use a growing Kohonen map such as the GWR algorithm used in the experiments in this thesis (see Section 3.3); then, active demonstrations would be useful to accentuate the differences between trajectories to influence their representation with different nodes, and more signalling could be used to explicitly signal these salient differences. Further, the sampling rate, used to create the input to the Kohonen map, could be adapted if the map cannot achieve a level of granularity which is fine enough to respond to the signals (see Section 6.4.3).

Similarly, in the work of Nehmzow and McGonigle (1994), where the part of the learning input corresponding to the infra-red sensors is encoded, this pre-defined level of granularity might not be suitable for some tasks. It might be more appropriate to start from the raw data, and then use explicit signalling to help the robot structure these data into the coded data needed for the learning.

As discussed in Section 2.2.2, in some cases forcing a high level of abstraction through design is favourable when it is more reliable and robust, especially when this is related to motor control and specifically to object-manipulations. The developmental approach presented in Chapter 5 was proposed as a way of dealing with the difficult problem of interacting with objects, where perceptual segmentation of an observed movement is complemented with a set of basic sensorimotor skills, as we saw in the object-interaction experiments in Section 5.3.5. These sensorimotor skills exist prior to the social interactions, either due to previous learning or due to additional design. Either way, if these skills are represented at a fine enough level of granularity, then they could be used during the social interactions to develop new skills, whose level of granularity is imposed by the expert involved in the social interactions.

In the work of Kuniyoshi and colleagues (Kuniyoshi et al., 1994; Kuniyoshi and Inoue, 1993) the level of granularity is very firmly specified at design time: object shapes and sizes are specified, and the possible actions are confined to vertical up-down movements. As a consequence, the demonstrations have to be very precise, and hence active. The system performs very well, in that it can recognise the salient parts of the demonstration very fast, in real-time. The design restrictions mentioned above regarding the level of granularity are reasonable and general for the kind of assembly tasks the system was implemented on. However, these design restrictions would not be appropriate for other types of tasks, where the actions are not restricted to vertical movements and where objects either appear ('place' action) or disappear ('pick-up' action), that is, for tasks involving free movements and complicated object-interactions. For these kinds of tasks, the segmentation of the movements would be less trivial, and would benefit from active demonstrations, and the desirable level of abstraction for the object-interactions would benefit from explicit signalling as described above.

Kaplan et al. (2001) and Nicolescu and Matarić (2003) utilise demonstration methods that can be regarded as active and containing explicit signalling, as described in Section 2.3. However, their systems are provided with high-level behaviours that include object-interactions. If they wanted to train their robots to perform tasks that required behaviours at finer levels of granularity, it would be interesting to see if their demonstration methods would be appropriate for learning these behaviours from lower-level raw sensorimotor data. Kaplan's 'luring' is similar to the teacher-following scenarios in the mobile robots experiments such as the ones presented in this thesis. So for example, a 'kick' behaviour could be taught (rather than designed) by luring the robot to move its leg appropriately, and then rewarding this occurrence, that is, explicitly signalling its saliency; this might be complemented with training another behaviour that uses the same leg, and the differences between these two behaviours would be accentuated through the active demonstrations. Nicolescu and Matarić (2003) utilise explicit signalling to signal salient experiences, however learning consists of sequencing high-level behaviours, where these behaviours represent experiences such as avoiding obstacles, approaching boxes, etc. If instead the robot were to learn some of these behaviours from its raw sensorimotor data, then the explicit signalling could be used to structure the experiences necessary for these behaviours, and active demonstrations would ensure sufficient exposure to the data.

Learning Resources

Learning at a very high level of abstraction, for example using high-level behaviours as do Kaplan et al. (2001) and Nicolescu and Matarić (2003), means that the load on the learning architecture is very light, because there are very few learning 'data' (behaviours in the examples above), and consequently very few learning steps. We also saw in Chapter 5 that when learning at a high level of abstraction, learning performance is not affected by modulation of the learning examples. That is, similar performances are achieved with few or many examples.

In contrast, we saw in Chapter 4 that at a very low level of abstraction learning benefits from as many data as possible, because data are not as well structured at this level. In fact, the results suggest that the system learns best by considering all the available sensorimotor data, therefore not requiring any designer effort in modulating, and hence abstracting the data. The simulation experiments showed that even if learning from all the data *is* desirable, the performance might be affected if there is a limit on the learning resources, for example, the number of data the learning architecture can consider.

The difficulty recognised by Demiris (1999) in finding a suitable level of granularity might be solved by learning only fine level behaviours, that is, building behaviours only for letters. Then, the learner could match demonstrations of any combinations of these letters by activating the appropriate behaviours in turn. However, this would of course increase the amount of resources needed, and slow down the recognition rate and hence the ability to imitate in real time, because observed behaviours are matched with all existing behaviours; representing words is more efficient. Fod et al. (2002) abstract sensorimotor data by projecting them onto a lower dimension using PCA, and then cluster the projected data. They mention a tradeoff between accuracy when using many principal components and clusters, and memory and processing time requirements for dealing with them. Similarly, all the examples mentioned above where the learning of trajectories relies on segmentation (Gaussier et al., 1998; Moukas and Hayes, 1996; Schaal et al., 2001; Kaiser and Dillmann, 1996; Yeasin and Chaudhuri, 1998) could learn from every data point instead of finding salient segments. Of course, the amount of learning data would increase drastically, and might not be computationally tractable.

The point about learning resources is that while it *might* be possible for a learning system to consider all the possible sensorimotor experiences for learning, and thus eliminate the need for *any* external bias regarding saliency and abstraction to be imposed on it, the performance of such a system could nevertheless be improved through such external bias for abstracting from the raw experiences and thus making better use of the available resources.

We saw in Figure 6.3 how the way that performance varies in the design space with respect to designer effort changes when there are *unlimited* resources. A similar influence might be expected with respect to social interactions, that is, if there are unlimited resources and no abstraction is therefore needed, then the benefit of explicit signalling might be insignificant. However, we would not expect social interactions to be completely insignificant because they provide task-relevance, as well as a bias for saliency detection; the robot would not be able to learn by being alone in the environment, unless of course, the learner was equipped with some other type of teaching signal (for example, reinforcement learning), and even then social interactions are useful for guiding the robot through the desirable experiences fast, as was discussed in Chapter 1.

6.4.3 Saliency Parameters

Part of the argument for having the ability to balance designer effort through social interactions is that saliency is modelled explicitly through parameters that deal with the saliency of *current* experiences. The identification of the important parameters makes it possible to transfer the responsibility of biasing learning through the setting of saliency parameter values, from the designer to the expert involved in the social interactions. While active demonstrations can influence the detection of saliency, they do not explicitly influence what constitutes saliency. The experiments in this thesis have started to address the utilisation of social interactions for explicitly influencing the *measure* of saliency.

In the physical wall-following experiment, the explicit signals serve to dishabituate the activated SOFM node, forcing attention to be given to the current perceptions. These signals therefore impose that the current experiences are salient, a role that is otherwise served by the novelty detection parameter and the full-habituation parameter, which respectively specify when experiences are novel, or when they are very familiar and should be 'forgotten-about'. In the phototaxis experiment, the explicit signals serve to specify when experiences should be learned, a role that is otherwise served by the change detection parameter of the perception of change mechanism. Thus, again, the signals explicitly influence what constitutes saliency, by taking the role of the saliency parameter.

We hypothesise that these signals can influence further what constitutes saliency, as follows. In the phototaxis experiment, the fact that the signals take the role of the saliency parameter can be used to adapt the parameter such that its notion of saliency is more similar to that of the source of the signals, that is, the expert. For example, a failure to detect a change when a signal is given — a 'miss' — might result in modifying the threshold such that it is more sensitive; 'false alarms' could be dealt with similarly.

In the physical wall-following experiment such an approach is less straight-forward, because there are two sources of saliency detection in the attention system, but there is only one type of signal. A signal from the expert could mean either that the experience is novel, or that not enough attention has been given to it, each related to a different parameter. Of course, one way of solving this is to use two different signals, but there is a difficulty here which is inherent to this kind of external parameter-tuning approach. While it is easy for the expert to know that a demonstration is novel, it is more difficult to know if enough attention has been given to particular experiences.

What might help is the robot signalling back to the expert some information on its internal (attentive) states (thus 'explicit communication'). For example, the habituation value of the active SOFM node specifies how familiar the experience that activated it is, which also specifies if it is novel (see Section 3.3). This could influence what the expert signals. But the problem above remains — how is the expert to know what is *enough* attention? This is related to the characteristics of the learning architecture. Perhaps the robot could also signal the state of its learning to suggest when it has learned enough, which could be related, for example, to weight-convergence. However, if this kind of information is subjective to the task, then we are back at the starting point.

This is a difficult and interesting problem, which might be easier if the expert has some knowledge of the robot's learning abilities. We therefore suggest that it is a problem worth investigating in future work.

6.5 Overall Conclusions from the Thesis

This thesis addressed the issue of influencing the learning by a robot, through design, and through social interactions. It argued that such external influences are necessary when training a robot to perform a task that is defined externally, because a useful interpretation of all the robot's sensorimotor experiences is highly subjective to the actual task. 'Useful' means firstly that the robot learns from experiences which are relevant to the particular task, and avoids distracting experiences; this issue was addressed by providing the robot with an expert, and equipping the robot with mechanisms for social interactions through which the expert can guide the robot to have particular experiences. Secondly, 'useful' means that the robot processes and compares sensorimotor data at an appropriate level of granularity; this is important because if the level is too coarse the robot could miss out important complexities in the task, and if too fine the job of finding structure is more difficult and requires heavier processing. The thesis motivated the need to address these issues by providing an empirical investigation focusing purely on the robot's experiences as it is exposed to different tasks in different environments, in Chapter 3.

Social interactions are used widely in the robotics field for various purposes, as discussed in Chapter 2, and they are also used widely for the purposes of this thesis — to influence the learning by a robot. It is recognised in the literature that social interactions are useful for reducing a robot's experiences to those relevant for a particular task, and thus speeding up learning. It is also recognised that even with social interactions there is a difficulty arising from an imprecise exposure to experience due to the robot's imprecise copying of the expert's actions. Indeed, the empirical investigation of the robot's experiences in Chapter 3, and the experiments testing the robot's ability to learn from these experiences in Chapters 4 and 5 showed that it is desirable for the robot to rely on the expert for exposure to sensorimotor data, and tested the effect of imprecise exposure. The utilisation of social interactions in this thesis is novel in that it proposes that they can be specifically tailored towards biasing the robot in terms of detecting saliency, and hence learning from its sensorimotor data at an appropriate level of granularity, especially in the face of imprecise exposure to data. Social interactions are recognised in this thesis as mechanisms with which an expert can actively and explicitly influence the robot's notion of saliency.

By recognising this role for social interactions, the thesis proposes that the necessary influences on the robot's learning could be obtained through a balance of *a priori* design, and situated on-line social interactions. In order to facilitate this balance, a design space is identified, consisting of two dimensions that correspond to the amount of effort required either by the designer or the expert, with regards to saliency. Each point in this space represents the amount of 'assistance' given to the robot by each of these two external sources to detect saliency usefully for learning. An increasing effort from the expert corresponds to more active and explicit interactions; with active demonstrations the expert firstly increases the precision of the exposure to experiences, and secondly accentuates the important differences between the different experiences; with explicit interactions the expert signals the occurrence of saliency to the robot, and directly influences the robot's internal notion of saliency through parameter-tuning. An increasing effort from the designer corresponds to more abstraction of the sensorimotor data, that is, forcing the level of granularity at which the robot detects saliency and learns. Because in related work social interactions have not been tailored towards biasing a robot's detection of saliency, the balancing suggested in this thesis has also not been attempted, and this identifies a gap in the research, clearly identifiable in the design space (see Figure 6.4).

The thesis argues that a crucial ingredient to having the ability to balance design effort with social interactions is that saliency is modelled through explicit parameters that reflect the robot's current experiences. The reason for this is that the expert is situated in the environment with the robot and influences its learning, while the robot is learning. For the expert's biases to be incorporated by the robot, the robot must be able to modify its measures of saliency on-line. In this thesis, the explicit treatment of saliency parameters was realised through the concept of attention. Attention as an abstraction mechanism was shown to serve two useful purposes: modulating the amount of learning through the familiarity of experiences, and structuring low-level experiences for high-level learning. Using attention as a parameterised modelling of saliency, this thesis addresses some of the gaps identified in the research, namely balancing designer effort in abstracting perceptual data through the setting of saliency parameters, with active demonstrations and explicit signalling.

In order to demonstrate the value of the design space identified in this thesis, examples of related work in the literature were organised into this space in Section 2.3, and characterised based on the two dimensions. The applicability of the various components that make up the two dimensions to related work illustrated their usefulness beyond the characterisation of the work reported in this thesis. The value of the design space was demonstrated further by showing how learning performance varies in this space, and how this is affected by the available resources. A performance surface was characterised from the results, showing how designer effort can be balanced through social interactions while maintaining a level of performance, how performance can be improved to a certain point by increasing any one of the two types of effort, and how the amount of one type of effort affects the significance of modifying the other type in terms of affecting the performance. Such a characterisation of performance is useful for suggesting how other systems could be improved. Thus the value of the design space was demonstrated even further by considering the related work again, suggesting the implications of moving them to different locations in the space for improving their performance.

The conclusions from the results in this thesis were summarised in Section 6.3. The general claims derived from these conclusions were supported further in Section 6.4 which identified how similar conclusions would be observed in related work if the framework for balancing designer effort with social interactions was applied. The first claim is that designer effort can always be balanced through social interactions, as long as the abstraction of the robot's experiences by the designer is flexible enough to be influenced by social interactions that are specifically tailored to deal with the abstraction of the robot's experiences and its notion of saliency. The second claim is that performance can generally be improved by increasing either type of effort, where the amount of improvement and its levelling out depend on learning resources (how well the learning can deal with a low level of abstraction which corresponds to fine granularity or even ‘raw’ data), and other design issues that influence how well the system is set up to learn the particular task (such as the robot’s morphology and learning architecture). The third claim is that generally there is a *desirable* balance between designer effort and social interactions, and this is also dependent on the design issues mentioned for the second claim; particularly, while it is possible that a good performance will be obtained from maximising or minimising the two types of effort, it will often be the case, for example, that *some* designer effort is desirable to guarantee a certain degree of reliability, but not too much so that sufficient influence can be obtained from the social interactions to guarantee some generality without over-exerting the expert.

The work reported in this thesis, together with the consideration of related work, shows that the design space proposed in this thesis provides a useful framework for addressing the trade-off between influencing a robot’s learning through design thus providing more reliability, and through social interactions thus providing more generality. The challenge is to find a desirable balance, for a reliable learning setup which is also general, adaptive, and faithful to the robot’s experiences, and which achieves a desirable level of performance considering the available resources.

Bibliography

Acklam, P. J. (2003). An algorithm for computing the inverse normal cumulative distribution function. Published on the World Wide Web at `http://home.online.no/~pjacklam/notes/invnorm/`. Document contains links to various implementation in different programming languages.

Afifi, A. A. and Clark, V. A. (1996). *Computer-Aided Multivariate Analysis*. Chapman & Hall, London.

Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2000). Learning how to do things with imitation. In *AAAI Fall Symposium on Learning How to Do Things*.

Aloimonos, J., Weiss, I., and Bandyopadhyay, A. (1987). Active vision. *International Journal of Computer Vision*, 1(4):333–356.

Andry, P., Gaussier, P., and Nadel, J. (2002). From visuo-motor development to low-level imitation. In Prince, C. G., Demiris, Y., Marom, Y., Kozima, H., and Balkenius, C., editors, *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Edinburgh, UK.

Arbib, M. A. (1981). Perceptual structures and distributed motor control. In Brooks, V. B., editor, *Handbook of Physiology: The Nervous System II. Motor Control*, pages 1449–1480. American Physiological Society, Bethesda, USA.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):996–1005.

Balkenius, C. (2000). Attention, habituation and conditioning: Toward a computational model. *Cognitive Science Quarterly*, 1(2):171–204.

Balkenius, C. and Hulth, N. (1999). Attention as selection-for-action: A scheme for active perception. In *Proceedings of Eurobot '99*.

Balkenius, C. and Morén, J. (1999). Dynamics of a classical conditioning model. *Autonomous Robots*, 7:41–56.

Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., and Breazeal, C., editors (2001). *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies.

Behrmann, M. and Haimson, C. (1999). The cognitive neuroscience of visual attention. *Current Opinion in Neurobiology*, 9:158–163.

Billard, A., Dautenhahn, K., and Hayes, G. (1998). Experiments on human-robot communication with Robota, an imitative learning and communicating doll robot. In *Proceedings of Socially Situated Intelligence Workshop, SAB'98 conference*.

Billard, A. and Hayes, G. (1999). DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behaviour Journal*, 7(1):35–64.

Bishop, C. M. (1994). Novelty detection and neural network validation. *IEEE Proceedings on Vision, Image and Signal Processing*, 141(4):217–222.

Bogacz, R., Brown, M., and Giraud-Carrier, C. (1999). High capacity neural networks for familiarity discrimination. In *Proceedings of ICANN'99*, Edinburgh.

Breazeal, C. and Scassellati, B. (1999a). A context-dependent attention system for a social robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*, pages 1146–1151, Stockholm, Sweden.

Breazeal, C. and Scassellati, B. (1999b). Infant-like social interactions between a robot and a human caretaker. *Adaptive Behavior*, 7. Special Issue on Simulation Models of Social Agents, Guest Editor Kerstin Dautenhahn.

Breazeal, C. and Velásquez, J. (1998). Toward teaching a robot 'infant' using emotive communication acts. In *Proceedings of 1998 Simulation of Adaptive Behavior (SAB98) Workshop on Socially Situated Intelligence*, Zurich, Switzerland.

Broadbent, D. E. (1958). *Perception and Communication*. London: Pergamon Press.

Brooks, R. A. (1991). Intelligence without reason. A.I. Memo No. 1293, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Brooks, R. A., Breazeal, C., Irie, R., Kemp, C. C., Marjanovic, M., Scassellati, B., and Williamson, M. (1998). Alternate essences of intelligence. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 961–968.

Brown, M. and Xiang, J. (1998). Recognition memory: Neuronal substrates of the judgement of prior occurrence. *Progress in Neurobiology*, 55:149–189.

Byrne, R. W. and Russon, A. E. (1998). Learning by imitation: a hierarchical approach. *Behavioral and Brain Sciences*, 21:667–721.

Byrne, R. W. and Tomasello, M. (1995). Do rats ape? *Animal Behaviour*, 50:1417–1420.

Carpenter, G. A. and Grossberg, S. (1988). The ART of adaptive pattern recognition by a self–organising neural network. *IEEE Computer*, 21:77–88.

Clouse, J. (1995). Learning from an automated training agent. In *Working Notes of the ICML '95 Workshop on Agents that Learn from Other Agents*.

Coull, J., Frith, C., Buchel, C., and Nobre, A. (2000). Orienting attention in time: behavioural and neuroanatomical distinction between exogenous and endogenous shifts. *Neuropsychologia*, 38:808–819.

Coull, J. and Nobre, A. (1998). Where and when to pay attention: the neural systems for directing attention to spatial locations and to time intervals as revealed by both PET and fMRI. *Journal of Neuroscience*, 18:7426–35.

Cowan, N. (1988). Evolving conceptions of memory storage, selective attention, and their mutual constraints within the human information processing system. *Psychological Bulletin*, 104(2):163–191.

Crook, P. (2000). Spotting novelty: A neural network model for familiarity discrimination. Master's thesis, Division of Informatics, University of Edinburgh.

Damper, R. I., French, R. L. B., and Scutt, T. W. (2000). ARBIB: An autonomous robot based on inspiration from biology. *Robotics and Autonomous Systems*, 31(4):247–274.

Dautenhahn, K. (1995). Getting to know each other — artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16(3–4):333–356.

Dautenhahn, K. and Nehaniv, C. L. (2002). The agent-based perspective on imitation. In Dautenhahn, K. and Nehaniv, C. L., editors, *Imitation in Animals and Artifacts*. MIT Press, Cambridge, MA.

Decety, J., Perani, D., Jeannerod, M., Bettinardi, V., Tadary, B., Woods, R., Mazziotta, J. C., and Fazio, F. (1994). Mapping motor representations with positron emission tomography. *Nature*, 371(6498):600–602.

Demiris, J. (1999). *Movement Imitation Mechanisms in Robots and Humans*. PhD thesis, Division of Informatics, University of Edinburgh.

Demiris, J. and Hayes, G. (1996). Imitative learning mechanisms in robots and humans. In *Proceedings of the 5th European Workshop on Learning Robots*, pages 9–16.

Demiris, J., Rougeaux, S., Hayes, G. M., Berthouze, L., and Kuniyoshi, Y. (1997). Deferred imitation of human head movements by an active stereo vision head. In *Proceedings of the 6th IEEE International Workshop on Robot Human Communication*, pages 88–93.

Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–197.

Deutsch, J. A. and Deutsch, D. (1963). Attention: Some theoretical considerations. *Psychological Review*, 70:80–90.

Fadiga, L., Fogassi, L., Pavesi, G., and Rizzolatti, G. (1995). Motor facilitation during action observation: A magnetic stimulation study. *Journal of Neurophysiology*, 73(6):2608–2611.

Fod, A., Matarić, M. J., and Jenkins, O. C. (2002). Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems, Special issue on Socially Interactive Robots*, 42(3–4):143–166.

Fritzke, B. (1995). A growing neural gas network learns topologies. In Tasauro, G., Tourestzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems*, volume 7, pages 625–632. MIT Press, Cambridge MA.

Gadanho, S. C. and Hallam, J. (1998). Emotion-triggered learning for autonomous robots. In *SAB'98 Workshop on Grounding Emotions in Adaptive Systems*.

Galef, Jr., B. G. (1988). Imitation in animals: history, definition, and interpretation of data from the psychological laboratory. In Zentall, T. R. and Galef, Jr., B. G., editors, *Social Learning: Psychological and Biological Perspectives*, pages 3–28. Lawrence Erlbaum Associates, Hillsdale, N.J.

Galef, Jr., B. G. and Beck, M. (1985). Aversive and attractive marking of toxic and safe foods by Norway rats. *Behavioral and Neural Biology*, 43(3):298–310.

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119:593–609. Part 2.

Gaussier, P., Moga, S., Banquet, J., and Quoy, M. (1998). From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 7–8(12):701–727.

Green, D. M. and Swets, J. A. (1966). *Signal Detection Theory and Psychophysics*. Wiley, New York.

Hallam, B. (2001). *Simulating Animal Conditioning: Investigating Halperin's Neuro-Connector Model*. PhD thesis, University of Edinburgh.

Hayes, G. M. and Demiris, J. (1994). A robot controller using learning by imitation. In Borkowski, A. and Crowley, J. L., editors, *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, pages 198–204. LIFIA-IMAG, Grenoble.

Hayes-Roth, B. (1995). An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72:329–365.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction To The Theory of Neural Computation*. Addison-Wesley, Redwood City, CA.

Heyes, C. (1993). Imitation, culture and cognition. *Animal Behavior*, 46:999–1010.

Hogan, D. E. (1988). Learned imitation by pigeons. In Zentall, T. R. and Galef, Jr., B. G., editors, *Social Learning: Psychological and Biological Perspectives*, pages 225–238. Lawrence Erlbaum Associates, Hillsdale, N.J.

Hugues, L. and Drogoul, A. (2001). Shaping of robot behaviors by demonstration. In *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 23–29. Lund University Cognitive Studies.

Husain, M. and Kennard, C. (1996). The role of attention in human oculomotor control. In Zangemeister, W. H., Stiehl, H. S., and Freksa, C., editors, *Visual Attention and Cognition*, pages 165–175. Elsevier Science Publishers, Amsterdam.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Kahneman, D. (1973). *Attention and Effort*. Prentice Hall, Englewood-Cliffs, NJ.

Kaiser, M. and Dillmann, R. (1996). Building elementary robot skills from human demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2700–2705.

Kaiser, M., Friedrich, H., and Dillmann, R. (1995). Obtaining good performance from a bad teacher. In *ICML Workshop on Programming by Demonstration*, Tahoe City, California.

Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., and Miklosi, A. (2001). Taming robots with clicker training : a solution for teaching complex behaviors. In Quoy, M., Gaussier, P., and Wyatt, J. L., editors, *Proceedings of the 9th European workshop on learning robots*, LNAI. Springer.

Klingspor, V., Demiris, J., and Kaiser, M. (1997). Human-robot-communication and machine learning. *Applied Artificial Intelligence Journal*, 11:719–746.

Knill, D. C. and Richards, W., editors (1996). *Perception As Bayesian Inference*. Cambridge University Press, Cambrdige.

Koch, C. and Ullman, S. (1985). Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kohonen, T. (1984). *Self-Organisation and Associative Memory*. Springer-Verlag, NY.

Kohonen, T. and Oja, E. (1976). Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. *Biological Cybernetics*, 25:85–95.

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6).

Kuniyoshi, Y. and Inoue, H. (1993). Qualitative recognition of ongoing human action sequences. In *Proceedings of IJCAI-93*.

Langley, P. (1997). Learning to sense selectively in physical domains. In *Proceedings of the First International Conference on Autonomous Agents*, pages 217–226. ACM Press.

Macmillan, N. A. and Creelman, C. D. (1991). *Detection Theory : A User's Guide*. Cambridge University Press, Cambridge.

Maistros, G. and Hayes, G. M. (2001). An imitation mechanism for goal-directed actions. In Nehmzow, U. and Melhuish, C., editors, *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Marom, Y., Maistros, G., and Hayes, G. (2001). Experiments with a social learning model. *Adaptive Behavior*, 9(3–4):209–240.

Marsland, S. (2001). *On-line Novelty Detection Through Self-Organisation, With Application to Inspection Robotics*. PhD thesis, University of Manchester.

Marsland, S. (2003). Novelty detection in learning systems. *Neural Computing Surveys*, 3:157–195.

Marsland, S., Nehmzow, U., and Shapiro, J. (2001). Novelty detection in large environments. In Nehmzow, U. and Melhuish, C., editors, *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Marsland, S., Shapiro, J., and Nehmzow, U. (2002). A self-organising network that grows when required,. *Neural Networks*, 15(8–9):1041–1058.

Matarić, M. J. (1997). Studying the role of embodiment in cognition. *Cybernetics and Systems*, 28(6):457–470.

Matarić, M. J. (2000). Getting humanoids to move and imitate. *IEEE Intelligent Systems*, 14(4):18–24.

McMillan, S., Orin, D. E., and McGhee, R. (1995). DynaMechs: An object oriented software package for efficient dynamic simulation of underwater robotic vechicles. In Yuh, J., editor, *Underwater Vechicles: Design and Control*, chapter 3, pages 73–98. TSI Press.

Meltzoff, A. N. and Moore, M. K. (1983). Newborn infants imitate adult facial gestures. *Child Development*, 54:702–709.

Meltzoff, A. N. and Moore, M. K. (1989). Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms. *Developmental Psychology*, 25(6):954–962.

Michel, O. (1996). Khepera simulator version 2.0. Robot simulator written at the Univeristy of Nice Sophia-Antipolis. Downloadable from the World Wide Web at `http://diwww.epfl.ch/lami/team/michel/˜khep-sim`.

Miniussi, C., Wilding, E., Coull, J., and Nobre, A. (1999). Orienting attention in the time domain: modulation of brain potentials. *Brain*, 122:1507–18.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill. ISBN 0070428077.

Moukas, A. and Hayes, G. (1996). Synthetic robotic language acquisition by observation. In *Proceedings of the Fourth Conference on Simulation of Adaptive Behavior (SAB'96)*, Cape Cod.

Nehaniv, C. and Dautenhahn, K. (2000). Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In Demiris, J. and Birk, A., editors, *Learning Robots: An Interdisciplinary Approach*. World Scientific. Robotics and Intelligent Systems Series, Vol. 24.

Nehmzow, U. and McGonigle, B. (1994). Achieving rapid adaptations in robots by means of external tuition. *From Animals to Animats*, 3.

Nicolescu, M. and Matarić, M. J. (2003). Natural methods for learning and generalization in human-robot domains. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Palameta, B. and Lefebvre, L. (1985). The social transmission of food-finding technique in pigeons: what is learned? *Animal Behaviour*, 33(Aug):892–896.

Pashler, H. E. (1998). *The Psychology of Attention*. MIT Press.

Pfeifer, R., Lungarella, M., Kuniyoshi, Y., Sporns, O., Metta, G., Sandini, G., and Nunez, R., editors (2001a). *Emergence and Development of Embodied Cognition (EDEC-2001)*, Beijing, Japan.

Pfeifer, R., Westermann, G., Breazeal, C., Demiris, Y., Lungarella, M., Nunez, R., and Smith, L., editors (2001b). *Developmental Embodied Cognition (DECO-2001)*, Edinburgh, UK.

Pomplun, M. and Matarić, M. J. (2000). Evaluation metrics and results of human arm movement imitation. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000)*.

Posner, M. and Peterson, S. (1990). The attention system of the human brain. *Annual Review of Neuroscience*, 13:25–42.

Posner, M., Snyder, C., and Davidson, B. (1980). Attention and the detection of signals. *Journal of Experimental Psychology General*, 109:160–174.

Prince, C. G., Demiris, Y., Marom, Y., Kozima, H., and Balkenius, C., editors (2002). *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies.

Rensink, R., O'Regan, J., and Clark, J. (1997). To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368–373.

Rimey, R. D. and Brown, C. M. (1990). Selective attention as sequential behavior: Modelling eye movements with an augmented hidden markov model. Technical Report 327, University of Rochester.

Rimey, R. D. and Brown, C. M. (1994). Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision, Special Issue on Active Vision*.

Rizzolatti, G., Carmada, R., Fogassi, L., Gentilucci, M., Luppino, G., and Matelli, M. (1988). Functional organization of inferior area 6 in macaque monkey .2. Area F5 and the control of distal movements. *Experimental Brain Research*, 71(3):491–507.

Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141.

Rizzolatti, G., Fogassi, L., and Gallese, V. (2000). Cortical mechanisms subserving object grasping and action recognition: A new view on the cortical motor functions. In Gazzaniga, M., editor, *The New Cognitive Neurosci.*, pages 539–552. MIT Press.

Schaal, S. (1997). Learning from demonstration. *Advances in Neural Information Processing Systems*, 9:1040–1046.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242.

Schaal, S., Vijayakumar, S., D'Souza, A., Ijspeert, A., and Nakanishi, J. (2001). Real-time statistical learning for robotics and human augmentation. In *International Symposium of Robotics Research*.

Sebastiani, P., Ramoni, M., and Cohen, P. R. (2000). Sequence learning via bayesian clustering by dynamics. In Sun, R. and Giles, C. L., editors, *Sequence Learning*, pages 11–34. Springer-Verlag.

Sokolov, E. (1963). A neuronal model of the stimulus and the orienting reflex. In Sokolov, E., editor, *Perception and the Conditioned Reflex*, pages 282–294. Pergamon Press, Oxford.

Terkel, J. (1996). Cultural transmission of feeding behaviour in the black rat (*rattus rattus*). In Heyes, C. M. and Galef, Jr., B. G., editors, *Social Learning in Animals: The Roots of Culture*, pages 17–47. Academic Press, San Diego, CA.

Tsotsos, J., Culhane, S., Wai, W., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling visual attention via selective tuning. *Artifical Intelligence*, 78(1–2):507–547.

van Reekum, C. and Scherer, K. (1997). Levels of processing in emotion-antecedent appraisal. In Matthews, G., editor, *Cognitive Science Perspectives on Personality and Emotion*, pages 259–300. Elsevier, Amsterdam.

Venables, W. and Ripley, B. (1994). *Modern Applied Statistics with S-Plus*. Springer-Verlag, New York.

Vijayakumar, S., Conradt, J., Shibata, T., and Schaal, S. (2001). Overt visual attention for a humanoid robot. In *IEEE International Conference on Intelligent Robots and Systems (IROS 2001)*.

Wang, D. (1995). Habituation. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, pages 441–444. MIT Press, Cambridge, Massachusetts.

Wang, D. and Arbib, M. (1992). Modeling the dishabituation hierarchy in toads: The role of the primordial hippocampus. *Biological Cybernetics*, 67:535–544.

Wessler, M. (1995). A modular visual tracking system. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Whitehead, S. and Ballard, D. (1990). Active perception and reinforcement learning. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 179–188, Austin, TX.

Whiten, A. (2000). Primate culture and social learning. *Cognitive Science special issue on Primate Cognition*, 24(3):477–508.

Wood, D., Bruner, J. S., and Ross, G. (1976). The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry*, 17:89–100.

Yeasin, M. and Chaudhuri, S. (1998). Automatic generation of robot program code: Learning from perceptual data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 889–894.

Zentall, T. R. and Galef, B. G., editors (1988). *Social learning: Psychological and Biological Perspectives*, Hillsdale, N.J. Lawrence Erlbaum Associates.

Ziemke, T., editor (2002). *Cognitive Systems Research, Special Issue on Situated and Embodied Cognition*, volume 3. Elsevier Science B.V.

Appendix A

Principal Components for the Robotic Tasks

This appendix shows the principal components (PC's) calculated in Section 3.2.2 by Principal Components Analysis (PCA) for all the robotic tasks mentioned there. Each table shown in this appendix shows three PC's as its columns, and each row is labelled with the sensor number that corresponds to that bit of the PC. The captions on the figures mention where in the thesis the relevant PC's are plotted, and also where a diagram of the location of the robot's sensors can be found. A diagram of the sensors of the simulated humanoid in the object-interaction experiments was not provided in the thesis, and is therefore shown below in Figure A.1.

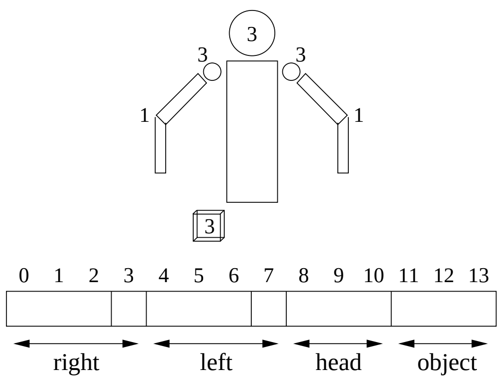


Figure A.1: A schematic diagram of the humanoid used in the object-interactions experiments. Indicated on the diagram of the robot are the degrees of freedom involved, consisting of joint-angles of the arms and head, and coordinates in 3D of the object. These values make up the first 14 bits of the input to the attention system, as shown below the diagram of the robot; the next 14 bits correspond to the instantaneous velocities of these values (not shown here).

	PC 1	PC 2	PC 3
0:	0.700584	-0.702847	0.122952
1:	0.114031	-0.051821	-0.923956
2:	0.000148	0.000931	-0.027873
3:	0.000004	0.000620	-0.010723
4:	-0.112107	-0.059763	0.351001
5:	-0.695421	-0.706928	-0.084231

(a)

	PC 1	PC 2	PC 3
0:	0.649039	-0.703036	0.289605
1:	0.186055	-0.223707	-0.954679
2:	0.000102	0.000674	-0.023214
3:	0.000069	0.000215	-0.010726
4:	-0.209675	-0.199377	0.063607
5:	-0.707225	-0.644937	-0.004240

(b)

	PC 1	PC 2	PC 3
0:	0.801318	-0.497884	0.257604
1:	0.263110	-0.201761	-0.671076
2:	0.016439	-0.022530	-0.524199
3:	0.006393	-0.015442	-0.428865
4:	-0.141607	-0.244012	-0.138476
5:	-0.517976	-0.806918	0.073567

(c)

	PC 1	PC 2	PC 3
0:	0.889933	-0.322009	0.177633
1:	0.291439	-0.116887	-0.314862
2:	0.048984	-0.040570	-0.613490
3:	0.030310	-0.048697	-0.613280
4:	-0.084790	-0.268128	-0.315143
5:	-0.335522	-0.898186	0.132330

(d)

Figure A.2: Three of the principal components calculated for the (a) hand-crafted, (b) following with social facilitation, (c) following, and (b) random scenarios (see Figure 3.11); the locations of the sensors are shown in Figure 3.9.

	PC 1	PC 2	PC 3
0:	0.216668	-0.104569	0.364163
1:	0.267264	-0.047667	0.365129
2:	0.187532	-0.062306	0.413763
3:	0.312391	0.055552	0.150093
4:	0.282071	0.115169	0.236727
5:	0.295709	0.115349	0.101258
6:	0.227503	0.186680	0.104964
7:	0.042995	0.191510	0.015853
8:	0.094615	0.297011	0.055138
9:	0.013572	0.365798	0.047461
10:	-0.040018	0.462550	-0.056961
11:	-0.156488	0.389401	0.140867
12:	-0.218054	0.333623	0.187132
13:	-0.262294	0.242282	0.177308
14:	-0.287698	0.133080	0.131286
15:	-0.122836	-0.019003	0.248942
16:	-0.315050	-0.025254	0.170128
17:	-0.294543	-0.105266	0.205598
18:	-0.237687	-0.201916	0.289880
19:	-0.190570	-0.231423	0.365357

Figure A.3: Three of the principal components calculated for the ‘hand-crafted’ scenario of the physical wall-following experiment (see Figure 3.14(a)); the locations of the sensors are shown in Figure 3.3.

	PC 1	PC 2	PC 3
0:	0.208229	-0.028935	0.423714
1:	0.273677	-0.054283	0.322589
2:	0.193524	-0.029265	0.191986
3:	0.259587	-0.276860	0.217854
4:	0.265332	-0.254272	0.238327
5:	0.309683	-0.163796	0.044508
6:	0.253676	-0.225382	-0.055515
7:	0.045064	-0.122772	-0.259605
8:	0.136087	-0.343111	-0.271578
9:	-0.024488	-0.439736	-0.084227
10:	-0.011353	-0.246566	0.055918
11:	-0.122165	-0.366991	-0.191725
12:	-0.207256	-0.368737	-0.066770
13:	-0.272445	-0.241740	0.033350
14:	-0.292678	-0.153285	0.065047
15:	-0.133895	-0.120110	0.295410
16:	-0.301553	-0.133777	0.168033
17:	-0.302400	-0.062561	0.182709
18:	-0.258211	-0.024660	0.288565
19:	-0.200556	0.006242	0.374347

Figure A.4: Three of the principal components calculated for the ‘following + social facilitation’ scenario of the physical wall-following experiment (see Figure 3.14(b)); the locations of the sensors are shown in Figure 3.3.

	PC 1	PC 2	PC 3
0:	0.212647	-0.017367	0.417719
1:	0.276501	-0.052047	0.316755
2:	0.196826	-0.031274	0.196405
3:	0.261338	-0.266746	0.231174
4:	0.264271	-0.236774	0.249901
5:	0.309933	-0.163662	0.041018
6:	0.252220	-0.235180	-0.061316
7:	0.045224	-0.149712	-0.284173
8:	0.131682	-0.346061	-0.258724
9:	-0.024008	-0.428864	-0.042997
10:	-0.012622	-0.244718	0.065754
11:	-0.120059	-0.378825	-0.173558
12:	-0.203823	-0.375547	-0.057831
13:	-0.270003	-0.249148	0.033649
14:	-0.291873	-0.156469	0.060152
15:	-0.130287	-0.093481	0.298798
16:	-0.301839	-0.129559	0.174801
17:	-0.302134	-0.059818	0.188387
18:	-0.258831	-0.016687	0.290882
19:	-0.203096	0.015382	0.365031

Figure A.5: Three of the principal components calculated for the ‘following’ scenario of the physical wall-following experiment (see Figure 3.14(c)); the locations of the sensors are shown in Figure 3.3.

	PC 1	PC 2	PC 3
0:	0.243243	-0.197905	0.205058
1:	0.291101	-0.142456	0.230995
2:	0.287992	-0.101579	0.297432
3:	0.253055	0.047360	0.514634
4:	0.096592	0.102602	0.187146
5:	0.269818	0.166879	0.190253
6:	0.178377	0.263927	0.178297
7:	0.059688	0.181875	-0.146061
8:	0.038576	0.402197	0.051099
9:	-0.041680	0.182023	0.214792
10:	-0.102580	0.350852	0.193841
11:	-0.170729	0.336672	0.148361
12:	-0.245897	0.268888	0.173799
13:	-0.325113	0.159900	0.129299
14:	-0.338167	0.076299	0.101489
15:	-0.052472	0.006851	0.058232
16:	-0.306351	-0.105706	0.232434
17:	-0.294382	-0.233263	0.247868
18:	-0.240517	-0.294889	0.253353
19:	-0.164540	-0.315032	0.279887

Figure A.6: Three of the principal components calculated for the ‘random’ scenario of the physical wall-following experiment (see Figure 3.14(d)); the locations of the sensors are shown in Figure 3.3.

	PC 1	PC 2	PC 3
0:	0.006424	-0.148012	0.100396
1:	-0.065452	-0.899375	0.384874
2:	-0.717815	-0.160578	-0.567415
3:	-0.692032	0.234338	0.526860
4:	-0.038513	0.292744	0.483291
5:	0.004969	0.053097	0.093180

(a)

	PC 1	PC 2	PC 3
0:	0.221928	-0.625946	0.746153
1:	0.011290	-0.762310	-0.637812
2:	-0.731187	-0.146015	0.056217
3:	-0.639881	-0.067279	0.178539
4:	-0.080540	0.030078	0.037446
5:	0.007180	0.017955	-0.003615

(b)

	PC 1	PC 2	PC 3
0:	0.072981	-0.354960	0.704509
1:	-0.303056	-0.364013	0.480946
2:	-0.655055	-0.042854	-0.090064
3:	-0.649910	0.015372	-0.117517
4:	-0.220762	0.605659	0.281686
5:	0.051136	0.610412	0.413629

(c)

Figure A.7: Three of the principal components calculated for the (a) hand-crafted, and (b) following, and (c) random scenarios in the phototaxis experiments (see Figure 3.17); the locations of the sensors are shown in Figure 3.9.

	PC 1	PC 2	PC 3
0:	0.000334	-0.000460	0.000372
1:	0.005792	-0.007432	0.015465
2:	0.000070	-0.000930	-0.000180
3:	0.000097	-0.000482	0.000180
4:	-0.198485	-0.020979	-0.010387
5:	0.010693	-0.025806	-0.082067
6:	0.237261	-0.091046	-0.068678
7:	0.180465	0.023169	0.076620
8:	0.000060	-0.000268	-0.000208
9:	0.000277	-0.002440	0.003163
10:	-0.000249	0.000362	-0.000032
11:	-0.000085	-0.000045	-0.000353
12:	-0.000023	-0.000467	0.000533
13:	0.000090	-0.000174	0.000414
14:	-0.000392	-0.001827	0.002704
15:	-0.000692	0.032590	-0.020936
16:	0.000027	0.000831	-0.001059
17:	-0.000146	0.001737	0.000913
18:	0.056452	0.276618	-0.374677
19:	0.013252	0.063521	0.117319
20:	0.009536	0.109243	0.412385
21:	-0.007169	0.887582	-0.205137
22:	-0.000050	-0.000070	-0.000140
23:	-0.001575	0.006908	-0.011551
24:	-0.000105	0.000052	-0.000717
25:	0.000133	-0.000559	-0.000186
26:	-0.000158	-0.000180	-0.000122
27:	-0.000022	0.000528	-0.000426

Figure A.8: Three of the principal components calculated for action 1 of the object-interaction experiment (see Figure 3.22); the locations of the sensors are shown in Figure A.1.

	PC 1	PC 2	PC 3
0:	0.000274	-0.000719	0.000721
1:	-0.001454	0.002708	-0.001630
2:	0.000161	-0.000154	-0.000191
3:	0.000021	0.000254	0.001375
4:	0.225905	0.075213	-0.048426
5:	-0.107786	0.116296	-0.053636
6:	-0.221521	0.072005	0.146340
7:	-0.128797	0.013644	-0.065111
8:	-0.020429	0.014913	0.004490
9:	0.012751	-0.008814	-0.005012
10:	0.001775	-0.002617	-0.002226
11:	-0.000024	0.000284	0.000233
12:	-0.000109	-0.000378	0.000489
13:	0.000201	0.000138	0.000240
14:	-0.000293	0.001240	-0.000352
15:	0.002205	-0.005119	0.003080
16:	0.000065	-0.001065	0.000241
17:	-0.000088	-0.000408	-0.000122
18:	-0.017089	0.145824	0.371812
19:	-0.031319	-0.421650	-0.140698
20:	-0.013574	-0.182614	-0.388225
21:	0.028951	-0.261225	-0.478389
22:	-0.003026	-0.031040	-0.115381
23:	0.003815	0.014378	0.067778
24:	0.000789	0.000375	0.006244
25:	0.000074	-0.000424	0.000462
26:	0.000045	0.000148	0.000239
27:	0.000059	0.000106	-0.000336

Figure A.9: Three of the principal components calculated for action 2 of the object-interaction experiment (see Figure 3.22); the locations of the sensors are shown in Figure A.1.

	PC 1	PC 2	PC 3
0:	0.208695	-0.411267	0.025290
1:	-0.072145	0.145055	-0.009015
2:	0.290218	-0.113197	0.318047
3:	-0.075585	0.285490	0.125341
4:	0.104117	-0.109785	0.172612
5:	0.022910	0.026535	0.094566
6:	0.016742	-0.248426	-0.119971
7:	0.038593	-0.183047	-0.052536
8:	-0.044491	-0.020157	-0.102748
9:	0.001850	-0.000296	-0.003368
10:	0.002029	0.007284	0.015265
11:	0.000239	0.000483	-0.000803
12:	-0.000038	0.000918	-0.000233
13:	0.000071	0.000581	-0.000667
14:	-0.534094	0.154431	-0.150985
15:	0.074858	-0.121446	0.005291
16:	-0.049757	-0.639764	-0.109081
17:	0.407632	0.215655	-0.213983
18:	-0.421122	-0.219007	-0.369098
19:	-0.063824	0.050566	-0.089560
20:	0.068426	0.115141	-0.548403
21:	-0.437783	-0.143240	0.476541
22:	-0.094050	0.167609	0.250970
23:	-0.005646	0.003606	0.010341
24:	0.007048	-0.015929	-0.004958
25:	0.000454	-0.000301	-0.001206
26:	0.000003	0.000879	0.000302
27:	-0.000366	0.000604	0.000454

Figure A.10: Three of the principal components calculated for action 3 of the object-interaction experiment (see Figure 3.22); the locations of the sensors are shown in Figure A.1.

	PC 1	PC 2	PC 3
0:	0.202302	-0.339058	0.219962
1:	-0.050842	0.076395	-0.079750
2:	0.124927	-0.239280	0.187155
3:	-0.326290	0.456433	-0.421608
4:	0.090781	0.048504	0.101461
5:	0.008009	0.044821	0.002431
6:	0.096803	-0.012890	0.364725
7:	0.217652	0.268021	0.380643
8:	0.000617	0.000295	0.000399
9:	0.001517	-0.000427	0.000947
10:	-0.000103	0.000313	-0.000377
11:	0.000107	-0.000182	-0.000372
12:	-0.000052	-0.000264	0.000207
13:	-0.000260	0.000136	-0.000073
14:	0.078728	0.110640	0.120090
15:	-0.046201	-0.054133	0.027687
16:	0.211866	0.179517	0.002593
17:	-0.631568	-0.545627	-0.014898
18:	-0.119622	0.204463	0.047666
19:	-0.079384	0.042923	-0.017265
20:	0.170730	0.096348	-0.304460
21:	-0.517201	0.373110	0.577558
22:	-0.000236	0.000290	0.000533
23:	-0.003191	0.002336	-0.001829
24:	0.000078	-0.000483	-0.000679
25:	-0.000698	-0.000089	0.000820
26:	0.000012	0.000083	0.000412
27:	-0.000122	-0.000061	0.000082

Figure A.11: Three of the principal components calculated for action 4 of the object-interaction experiment (see Figure 3.22); the locations of the sensors are shown in Figure A.1.