GAZE CONTROL FOR VISUALLY GUIDED MANIPULATION

by

JOSÉ IGNACIO NÚÑEZ VARELA

A Thesis submitted to the
University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computing Sciences
College of Engineering and Physical Sciences
University of Birmingham
December 2012

Abstract

Human studies have shown that gaze shifts are mostly driven by the task. In manipulation tasks, for example, gaze typically leads action to the next manipulation target. One explanation is that fixations gather information about task relevant properties, where task relevance is signalled by reward. This thesis pursues primarily an engineering science goal in order to determine what mechanisms a rational decision maker could employ to select a gaze location given limited information and limited computation time. To do so we formulate three computational models of gaze shifting, where the agent imagines ahead in time the informational effects of possible gaze fixations. Our first model selects the gaze that most reduces uncertainty in the scene (*Uncertainty*). The second model maximises expected rewards by reducing uncertainty (*Rew+Unc*). Our third model maximises the expected gain in cumulative reward by reducing uncertainty (*Rew+Unc+Gain*). The secondary goal of this thesis is concerned with the way in which humans might select the next gaze location. The main contributions of this thesis are: i) the formulation of three computational models of fixation selection, implemented on a simulated humanoid robot; ii) the characterisation of the models in terms of performance using a manipulation task under variations of three environmental variables; iii) the integration of an active visual search process into the *Rew+Unc+Gain* scheme; iv) the implementation and testing of the models in a bimanual manipulation task; v) comparison of the hand-eye coordination timings of our models to previously published human data. The results from the experiments ii) demonstrate that the *Rew+Unc+Gain* strategy has the best overall performance. In addition, the experiment v) provides evidence that only the models that incorporate both uncertainty and reward (*Rew+Unc* and *Rew+Unc+Gain*) match human data.

To my family.

Acknowledgements

This thesis could not have been possible without the support of many people.

First and foremost, I would like to thank my supervisor Jeremy L. Wyatt for his excellent and invaluable guidance, support and assistance throughout my whole Ph.D. Thank you very much and I look forward to our future collaboration.

Very special thanks go to my research monitoring group, Dr. Ela Claridge and Dr. Nick Hawes, for their exceptional and helpful assistance to my research.

I am also indebted to Dr. Ravindran Balaraman for helping me to shape this project at the early stages of the process. I also look forward to future collaborations.

Special thanks are also due to the members of the Intelligent Robotics Laboratory, for their suggestions and feedback given to me every time I presented my work.

I would also like to express my love and gratitude to my family for their love and support, not only for this four years, but for all the years of my life.

Enormous gratitude goes to all my friends in Mexico for their endless friendship and support, and to all the new friends I made here during the last four years.

Finally, I sincerely thank the Mexican Council of Science and Technology (CONACyT) for the scholarship granted (Reg.179604). I am also grateful of the funding bodies that helped me throughout my Ph.D.: UKIERI (SA06-0031), CogX (FP7-ICT-215181), and the School of Computer Science Research Committee.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Most of the tasks that we perform on a daily basis, such as organising our desk, walking in the street, cooking, reading, playing sport, driving, amongst many others; require us to act under uncertain and incomplete information. This uncertainty concerns object properties relevant to the ongoing task. Examples of these properties are, the location of a book, the shape of a cup, the colour of the traffic light, the distance of the car in front of us, etc. Knowing as much information as possible, i.e. minimising the uncertainty about these object properties, allows us to accomplish the given task more efficiently. This uncertainty can be reduced by gathering new information through sensing, and one such sense is vision.

The role of vision is not to process images of the world that are received passively, but to actively explore the scene to gather new visual information that would help us act and interact with our environment. To put it differently, the main function of vision is not just seeing, but looking [9]. This perceptual exploration is carried out by moving our eyes, our head, and even our whole body, to direct the line of sight to parts of the scene (e.g. objects) that are relevant to the task being performed. In this work, the aim of this exploration is to reduce *positional* uncertainty (as illustrated in Figure 1.1). This perceptual exploration is referred to as *active sensing* in general [9], and *active vision* [10] in particular.

This thesis is interested in two main questions, i) what mechanisms a rational decision

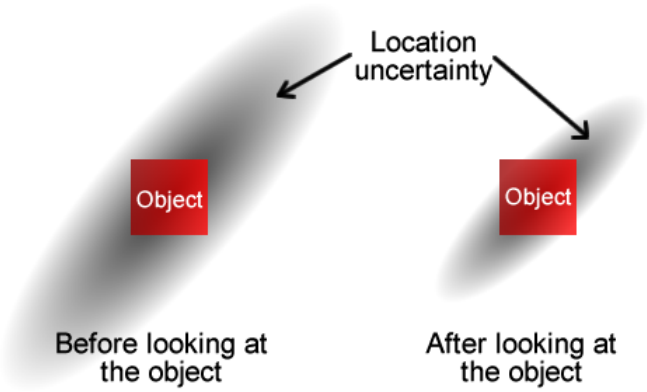

Figure 1.1: Representation of how the uncertainty about the location of an object is reduced after looking at the object.

maker could employ to select a gaze location, in order to maximise task performance, given limited information and limited computation time; and ii) how humans select the next gaze location (or fixation point). Previous work has suggested that in fact the answers to these two questions are similar: human eye movement behaviour is consistent with the use of decision making mechanisms for fixation selection, i) that are Bayes' rational [11, 12], or ii) that try to maximise reward values [13, 14]. This thesis investigates these claims further by means of two goals:

1. **Engineering science goal:** Our primary goal is to devise principled methods for gaze control by formulating three new computational models for fixation selection during the performance of a manipulation task. All three models are similar in that they calculate the benefit of a fixation by imagining its effect on the agent's information state one step ahead into the future. These models are implemented on a simulated humanoid robot which provides an excellent medium for testing and characterisation. In order to distinguish these models they are analysed by their relative performance in controlled conditions. In addition, two other models of gaze are implemented that serve as common baseline.

2. **Human behavioural goal:** Our secondary goal is to determine the goodness of fit of our models of gaze control to existing behavioural human data.

The rest of this introduction proceeds as follows. First, biological findings that motivate this thesis are discussed. Second, the processes involved during gaze control

are described. Third, a proposed decomposition for fixation selection is considered. Fourth, the benefit of using robotic simulations to study gaze control is discussed. Fifth, the three different computational models of gaze control are defined and explained. Sixth, the manipulation tasks used throughout the thesis are described. Seventh, the contributions of the thesis are listed. Finally the outline of the rest of the thesis is given.

1.1 Motivation

Even though this thesis has primarily an engineering objective, our proposed models of gaze control are in part inspired by the research of human eye movement. In particular, we are interested about the study of the role of vision in action. There are some questions falling within this topic: How are gaze shifts affected by the task? Conversely, how is the behaviour of the agent during task execution affected by the way gaze is deployed? What is the precise spatio-temporal relationship during eye-hand coordination? Possible answers to these questions have been provided by psychophysical human studies [10,15].

These psychophysical experiments, in constrained and unconstrained environments, have investigated the relationship between gaze shifts and actions during the execution of natural tasks. Examples of these tasks are, cooking, driving, walking, washing, sorting, amongst others [3, 6, 16, 17]. Empirical evidence from these studies has lead to the following findings:

1. Gaze shifts in the scene are, in general, driven by the ongoing task being performed [5, 16, 18].

2. Spatially, gaze is directed to the object(s) relevant to the task and to the sites where actions are taking place [1, 19, 20].

3. Temporally, gaze typically precedes action and guides it [1, 21, 22].

4. Vision is used to gather information about objects' properties relevant to the task [6, 23, 24], with the purpose of reducing uncertainty about those properties [25, 26].

5. Task structure and the world's dynamics can be learnt from experience, and this knowledge can be used for planning gaze shifts [15, 16].

Some questions derived from these behavioural findings are: What are the brain structures involved in the control of gaze and body actions during the performance of a task? How is the control of gaze learnt or planned? How does the brain determine the degree of relevancy of the objects involved in the task? Neurophysiological studies have provided some evidence concerning these questions and have shown that:

1. The basal ganglia are involved in various cognitive functions in the brain, such as action selection and eye movements [27].

2. Reward signals in the basal ganglia are used to modulate gaze control and the selection of actions [17, 28, 29].

3. Dopamine is regarded as the main neurotransmitter to signal expected reward [30]; whilst other neurotransmitters, such as acetylcholine, may represent expected uncertainty at the neural level [31].

These neurological findings have hinted at the possibility of employing mathematical models to formulate the decision making mechanisms involved during action selection and gaze control [32,33]. In particular, reinforcement learning [34] has been used for the implementation of gaze control models [7, 35]. At the same time, Bayesian probabilistic models have been proposed to explain how humans deal with uncertainty [11, 36]. The models of gaze control that we propose in this thesis follow a reward based approach along with a Bayesian inference process in order to select fixation locations.

1.2 Gaze Control Processes

This thesis is about the selection of fixation locations that would be performed in humans using *saccadic* eye movements, as opposed to fixations on moving targets that require smooth pursuit eye movements. The purpose of each saccade is to shift the *fovea* (the

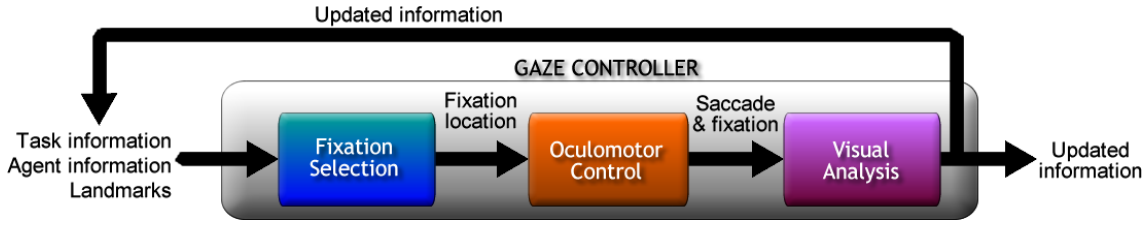

Figure 1.2: Gaze control is defined in this thesis as a sequential mechanism composed of three processes: fixation selection, oculomotor control and visual analysis.

central part of the retina which has high acuity), to a point of interest in the scene. After a saccade is completed the eyes are momentarily stationary and a *fixation* occurs [10, 37]. During a fixation, visual information is captured and analysed. Exactly what mechanisms and how gaze control is achieved by humans is still an open question, and this thesis do not attempt to give an answer to that particular question. However, based on this procedure, in this thesis we define gaze control as a sequence of three processes (depicted in Figure 1.2):

- **Fixation selection:** This is the decision making process that a rational agent (or robot) employs to select the next fixation location, and the object of this thesis. Here, the current information about the task and the information about the agent are used to make the decision. Thus, given a number of task relevant landmarks (e.g. objects on a table), the key idea is for the agent to imagine one step ahead in time the informational effects of fixating each available landmark and to select the one according to the gaze strategy being followed. Section 1.5 describes in more detail this decision making process and the different gaze strategies proposed by this thesis.

- **Oculomotor control:** Once the next fixation location is decided (i.e. a particular landmark), the aim of the oculomotor system is to direct the line of sight to fixate that location. The oculomotor control system is responsible for executing the corresponding saccade and the subsequent stabilisation of gaze that results in a fixation. How saccades are controlled at the motor level is an important issue which is further explored in Chapter 3. However, this topic is outside the scope of the original work of this thesis. Instead we rely on standard robotic control

techniques to direct the gaze to the landmark's location [38, 39]. Furthermore, saccades may involve the movement of the head and the agent's body. In this work only eye-head movements are considered, it is assumed that the body, or more specifically, the torso remains fixed.

- **Visual analysis:** After the oculomotor control system executes a saccade, a fixation occurs and this is when visual information from the current agent's view point, is captured, processed and analysed (e.g. using visual routines [40]). The aim of this visual analysis is to gather new information related to the task, the environment, and possibly about the agent as well. This new information might concern previously unseen landmarks, or be used to update the information about previously seen landmarks. It is expected that with this new information the agent could accomplish the task more efficiently by making better decisions about what action(s) to perform next. This information is also used for the selection of the next fixation location. The analysis of visual information requires the use of computer vision techniques for image processing which are also outside the scope of this thesis. Chapter 3 explores the ways in which other work on robot visual systems have approached this problem.

Because this thesis is solely concerned with the *fixation selection process* within the gaze controller, it is simpler to think about the *oculomotor control* system and the *visual analysis* as separate processes (Section 1.5.1 explains how we deal with these two processes throughout the thesis). However, this does not mean that we advocate to this particular sequential configuration. The design and implementation of these processes may change so that different parts may execute simultaneously at different levels. Furthermore, a gaze controller following a biological or neurobiological approach is more complex, as the known interactions between neurons and other brain structures is much more intricate [41, 42].

1.3 Decomposing the Fixation Selection Process

The aim of the *fixation selection process* within the gaze control mechanism (previous section) is to decide what task relevant landmark (e.g. an object on a table) to fixate next by imagining the informational effects of fixating each available landmark[1]. Therefore, the main problem faced by the gaze controller at this stage is to decide *where to look*. Nevertheless, fixation selection becomes more complex if the agent has multiple manipulation motor systems (e.g. two arms), each of which may be performing a separate task (e.g. each arm moves objects independently), or separate actions that contribute to a common task (e.g. bimanual grasping). In this case, a single fixation might not be able to assist all the manipulation actions running simultaneously. Thus, gaze should be treated as a serial resource that must be first *allocated* to a particular manipulation motor system, so that this motor system is able to select a particular fixational landmark. Thus, we identify two interrelated problems within the *fixation selection process*, namely the problems of *where to look* and *gaze allocation*. Therefore, it is a hierarchical scheme where gaze is first allocated to a manipulation motor system, then this motor system is given the ability to select its preferred landmark location.

1.3.1 Gaze Allocation

The gaze allocation problem emerges because we consider an agent with multiple motor systems (e.g. two arms) which can be running in parallel (e.g. each arm executes a separate task) or concurrently (e.g. both arms contribute to a common task)[2]. Typically, a single fixation might not be able to assist all the different manipulation motor systems running simultaneously. Therefore, gaze is treated as a serial resource that must be *shared* over time between the manipulation motor systems, and be *allocated*, or *assigned*, to only one of them at a particular time. Thus, gaze allocation can also be

[1] Throughout the thesis we might employ the general term of *gaze control* in some cases to refer to the selection of fixation points. If a particular distinction is required it will be pointed out accordingly.

[2] Both, parallelism and concurrency refer to processes that can be executed simultaneously. Parallel processes do not depend on each other, so no interaction is needed. In contrast, concurrent processes depend on each other for their correct execution, so normally a coordination mechanism is required to control the processes.

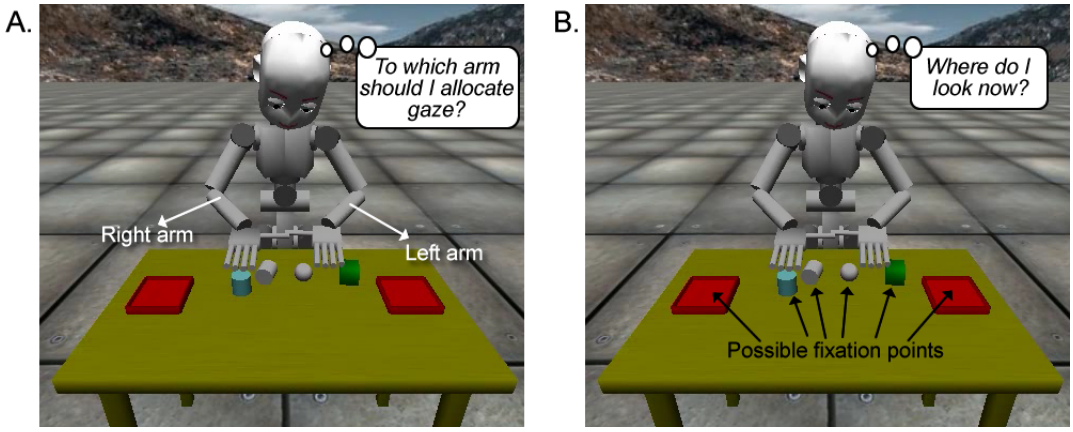

Figure 1.3: A humanoid robot performing a pick and place task. A. Gaze allocation determines which manipulation motor system should be given the ability to choose a particular landmark. B. The problem of where to look deals with the selection of a particular landmark in order to fixate it.

described as a scheduling problem [43] that assigns the choice about *where to look* to a specific manipulation motor system. As an example Figure 1.3A shows a humanoid robot performing a manipulation task that consists of picking up objects from the table top and then placing them inside one of two containers (pick & place task). The robot has two manipulation motor systems (i.e. two arms) that can simultaneously be performing separate tasks, or performing separate actions that contribute to a common task. However, if the robot fixates a landmark located on its right it might only assist the right arm, leaving the left arm with no perceptual aid[3]. In this case, the robot should rationally choose the manipulation motor system that most benefits in terms of task performance if it is given access to perception[4].

1.3.2 Where to Look

Once gaze has been *allocated*, or assigned, to a particular manipulation motor system, then it is possible to select one of the fixation locations in the scene. In this work, each fixation location is associated to a particular landmark in the scene. Thus the problem of where to look can be restated as the problem of what landmark to fixate

[3]What visual information is captured also depends on the observation model and the camera's field of view, which will be explored in Chapter 5.

[4]Gaze allocation can also emerge when an agent has multiple tasks to perform (in parallel or concurrently) but only one task can have access to perception at any one time [43].

Figure 1.4: A. The iCub humanoid robot. B. Snapshot of the iCub simulator.

next. As an example, Figure 1.3B presents once again the robot performing the pick & place manipulation task. Here, the objects and the containers serve as possible fixation points (i.e. landmarks).

The problems of *gaze allocation* and *where to look* have been dealt with separately. For example, [44–47] have proposed solutions for the problem of where to look, whilst [43] and [35] have tackled the gaze allocation problem. This thesis provides an integrated account where both problems are dealt with within the same gaze control mechanism.

1.4 Robotic Simulations to Study Gaze Control

The implementation of computational models in robotic platforms permits testing the models under different configurations of either model or environment that are difficult or impossible to conduct in humans or animals [48–51]. In the case of gaze control, for example, the robot's field of view can easily be altered between trials. Furthermore, employing simulated environments allows more flexibility in terms of the configuration of both the environment and the robot/agent. For instance, objects can be created or modified in real time. Robotic simulation also facilitates systematic experimentation in order to fully characterise the computational models [52].

For these reasons, we have chosen to implement our gaze control models using the iCub simulator [53] (Figure 1.4). The iCub is a humanoid robot with multiple motor systems [54], whilst the iCub simulator is an open source platform that employs the same controllers and modules used to control the real robot [55]. It also makes use of

the ODE physics engine to allow the simulation of rigid bodies [56]. In this thesis, we make use of both arms (i.e. *two manipulation motor systems*), where each has a hand with five fingers. Also, we make use of what we call the *perceptual motor system*, which includes the robot's head, neck and eyes.

1.5 Candidate Models of Gaze Control

As mentioned above, the aim of this thesis is to formulate computational models of gaze control that a Bayes' rational decision maker could employ to select a fixation location given limited information and limited computation time. This selection could be achieved by reasoning about future courses of manipulation actions over varying horizons. Thus, the gaze control models calculate the benefit of the possible fixation locations, currently known to the agent, by predicting their effect on the agent's *information state*[5] some number of steps into the future. We will, without loss of generality, consider *one-step look ahead* models in this thesis[6]. Predicting the benefit of the possible fixation locations is based mainly on two parameters: positional uncertainty of landmarks and the value of performing manipulation actions.

To understand how these two parameters affect the gaze control decision process, let us use an example using a manipulation task that consists of picking up objects from the table top and then placing them inside one of two containers. Figure 1.5A shows a humanoid robot performing such task.

- **Positional uncertainty of landmarks:** This work assumes that the location of landmarks is unknown to the robot. The robot has to look at landmarks in order to *estimate* their location. This happens when a landmark falls inside the robot's view point. As an example consider the snapshot of the pick & place task depicted in Figure 1.5A. There are five landmarks (three objects and two containers), where

[5]Central to the concept of uncertainty is the idea of the information state or the belief state. In general, this is the agent's summary of the total information acquired by the agent up to the current time about properties of interest. In this thesis, the terms information state and belief state will be used interchangeably. The first term is more intuitive; the second is commonly used in the field of Bayesian estimation (Section 2.7).

[6]*Look ahead* planning is a concept found in the artificial intelligence literature.

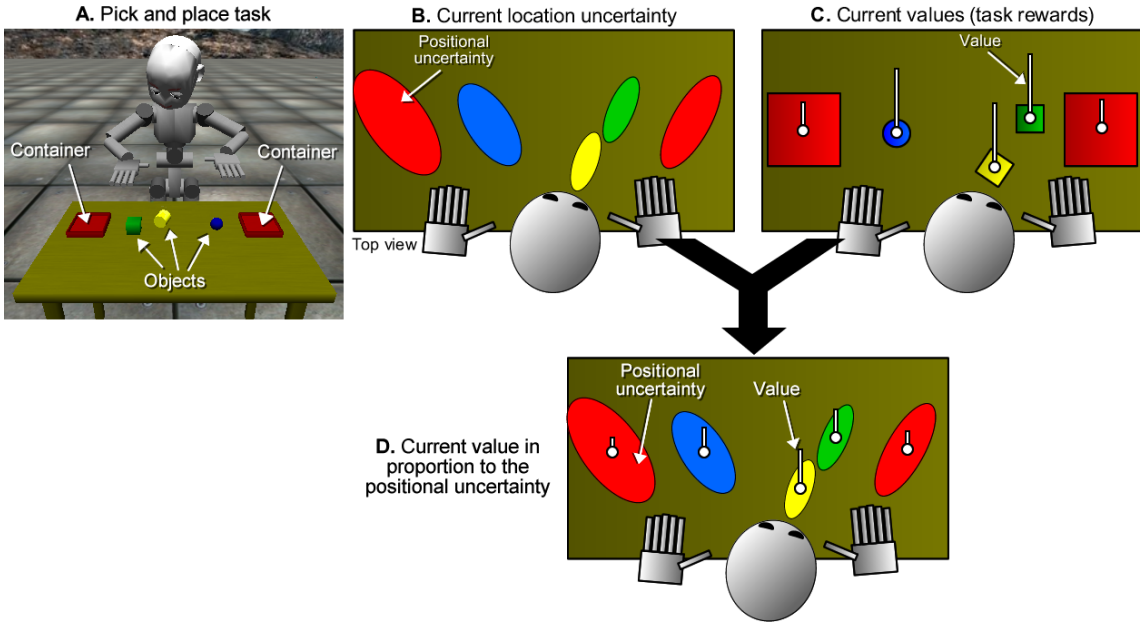

Figure 1.5: A. Pick and place task. B. Representation of the current location uncertainty of each landmark (objects and containers). C. Representation of the value of performing manipulation actions associated to each landmark. D. Representation of the current value in proportion to the location uncertainty of each landmark.

their positional uncertainty is represented as bivariate Gaussian densities (shown as ellipsoids in the top view of the same scene (Figure 1.5B)). For the current time step, notice how the left container has the highest positional uncertainty, whilst the yellow cylinder has the smallest.

- **Value of performing manipulation actions:** The manipulation actions can be quantified by the *cumulative discounted reward* (i.e. value) that the robot receives after performing such action. In this work these values are learnt via reinforcement learning [34] (Section 2.5), where the robot learns the sequence of manipulation actions that achieve some task by trial-and-error interactions with the environment. Let us assume that the robot already has learnt the values that achieve the pick & place task according to some reward function. Considering the same snapshot from Figure 1.5A, notice that each arm is ready to reach for an object (since the robot has learnt that when a hand is empty an object can be grasped). The value of reaching is then attached to each object/container. This value is represented by the vertical bars in Figure 1.5C. In fact, note that the value of reaching is the same for all three objects, because we do not make any

difference between objects.

Based on these two quantities, the agent needs to reason about how much value it is going to receive for a successful reach *traded-off* against the current positional uncertainty of each object/container. This *proportional* value is represented by the vertical bars in Figure 1.5D. For the current time step, notice how the value of reaching the blue sphere is the smallest amongst the three objects because of its high positional uncertainty.

It is based on these proportional values that the robot should decide how to control its gaze. By imagining what would happen if it looks at some fixation location in the next time step, the robot is able to answer questions such as: *How much will the uncertainty reduce if I look at landmark X? How much value do I expect to receive after looking at landmark Y? Which manipulation motor system would get more benefit if gaze is allocated to it? Which fixation location would yield the highest predicted value and the highest reduction in uncertainty?*

It is important to note that the robot can only reason about the benefit of fixating landmarks that it already *knows* exist[7]. Furthermore, from those known landmarks only a subset can actually be used by each manipulation motor system. For instance, following the same example shown in Figure 1.5A, the right arm can only manipulate those objects that are within its reach. Therefore, the space of objects is partitioned by the arms. In the example, the right container, the green cube and the yellow cylinder are associated with the right arm, whilst the left container and the blue sphere are associated with the left arm.

Depending on how the robot reasons about the positional uncertainty of landmarks and the value of performing manipulation actions, at least three gaze control models emerge. However, before such models are described we provide a list of assumptions considered throughout this thesis that would help understand our models of gaze control.

[7]The case where unknown landmarks are taken into account in the decision making process is explored in Chapter 6.

1.5.1 Assumptions

Because of the complexity of the overall robot's control system and the gaze control problem, this work makes a series of assumptions that need to be taken into account to understand the scope and the limitations of our models of gaze control, particularly when compared to human gaze control.

- **Spatial Acuity:** We make use of an *observation model* that produces a smoothly varying spatial acuity (Section 4.4.1). This models a smooth fall off in spatial acuity as we move from the centre of the field of view (0° to 10°) to the "periphery" (+10°). This is qualitatively similar but not the same as the human eye.

- **Fixations:** Because we consider non-uniform spatial acuity, centring the robot's camera(s) to a landmark is important. According to our observation model, only when landmarks lie within the central part of the current view point (0° to 10°) it is then possible to get a good estimate of its location.

- **Saccades:** As mentioned above, the control of saccadic movements at the motor level is outside the scope of our work. Instead we rely on standard robotic control techniques [38], particularly those available to the iCub humanoid robot (Section 1.4) which allow the movement of the robot's head, neck and eyes. We consider these three motor systems together as the *perceptual motor system*.

- **Visual Analysis:** Processing visual information requires the use of computer vision techniques which are also outside the scope of this work. Since we only need to *detect* landmarks in the current view point whenever a fixation occurs, then we will make use of the iCub simulator for object detection.

- **Uncertainty:** Although uncertainty may exist about many object properties e.g. shape and mass, our models are only tested with respect to the *positional* uncertainty of landmarks.

- **Learning:** The robot learns how to perform the task via reinforcement learning assuming all the landmarks' locations are known to the robot. This learning phase

is where the values of performing manipulation actions are determined according to a reward function (Section 4.3).

- **Grasping:** This is a complex problem which is also outside the scope of this paper. Once again we will take advantage of the iCub simulator by using a magnet-like function (explained in Section 4.1).

- **Manipulation Motor Systems:** These refer to the robot's arms. Because of the arm's joint limits, each arm has a set of landmarks that it can manipulate and which thus form the candidate fixation locations that could benefit that motor system. Furthermore, we assume that each manipulation motor system is able to perform a specific sub-task.

Next, we first explain our gaze control model based on uncertainty reduction; second, our model based on rewards and uncertainty; and third, our model based on rewards, uncertainty and gain.

1.5.2 Gaze Control based on Uncertainty Reduction

Our first model of gaze control (*Uncertainty* gaze strategy) aims to maximise the reduction of positional uncertainty only. It predicts one step into the future the location uncertainty that would remain if each of the known landmarks, for each manipulation motor system, is fixated. The residual positional uncertainty is calculated by the difference between the current and the predicted uncertainty. The model selects the fixation that is expected to most reduce uncertainty about the location of the corresponding landmark.

Following the same example shown in Figure 1.5, Figure 1.6A presents once again the same instance of the pick & place task, whilst Figure 1.6B depicts the current proportional value (as it was shown in Figure 1.5D). Thus, the predicted residual uncertainty that would result if each landmark is fixated the next time step is depicted as the small solid ellipsoids in Figure 1.6C (the big ellipsoids are those from Figure 1.6B representing the current uncertainty). In this example, looking at the right container will benefit the

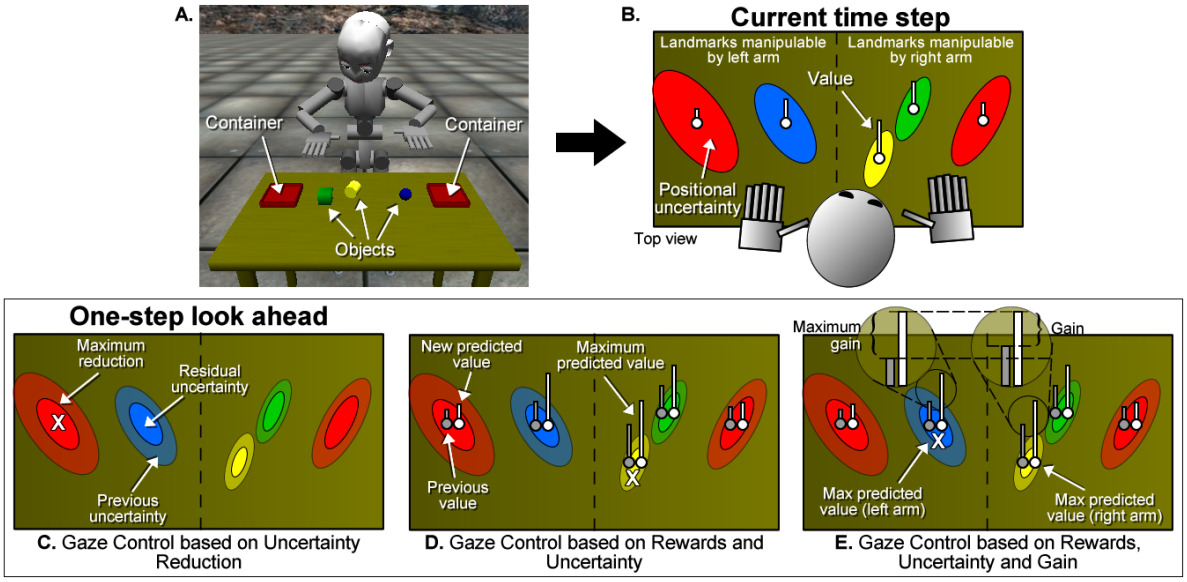

Figure 1.6: A. Pick and place task. B. Representation of the positional uncertainty (ellipsoids) and the value of reaching for each object/container (bars) in the current time step. C. Gaze control based on uncertainty reduction (*Uncertainty* gaze scheme). D. Gaze control based on rewards and uncertainty (*Rew+Unc* gaze scheme). E. Gaze control based on rewards, uncertainty and gain (*Rew+Unc+Gain* gaze scheme). The X specifies the next chosen fixation point for each scheme (See text for a detailed explanation).

right arm, whilst looking at the left container will benefit the left arm. Both cases have a high reduction in uncertainty, however the highest reduction comes from looking at the left container. Thus, gaze is *allocated* to the left arm and then, from those landmarks it can manipulate, the left container is chosen to be fixated next (marked with an X).

By disregarding task rewards completely (i.e. the values of performing manipulation actions), the robot focuses on gathering new information about the environment. However this information might not be relevant to the current needs of the task. In the example, the robot should reach for an object but has chosen to look at the left container instead. The experiments analysed in Chapter 5 demonstrate that, in general, this model of gaze control is not effective.

1.5.3 Gaze Control based on Rewards and Uncertainty

Our second gaze control model (*Rew+Unc* gaze strategy) extends the previous model by incorporating task rewards (i.e. the values of performing manipulation actions) into the decision process. By reducing the positional uncertainty of a task relevant landmark, the

robot should accomplish the manipulation actions more efficiently and so their expected value will increase. The model predicts the value of performing manipulation actions that would be obtained *if* each landmark, for each manipulation motor system, is fixated the next time step. Figure 1.6D shows this *predicted* value for each landmark (white vertical bars) contrasted with the grey bars that represent the *current* value (taken from Figure 1.6B). In the example, looking at the blue sphere produces a high value that benefits the right arm, whilst the left arm will benefit if the yellow cylinder is fixated. From these two landmarks, the maximum value comes from the yellow cylinder. Thus gaze is *allocated* (or assigned) to the right arm and the yellow cylinder is chosen to be fixated next (marked with an X).

The main drawback of this strategy is the tendency to look at landmarks with low positional uncertainty and high predicted value, which will not typically provide much new task information. Our results in Chapter 5 confirm that in most cases this is a sub-optimal gaze control strategy.

1.5.4 Gaze Control based on Rewards, Uncertainty and Gain

Our third gaze control model (*Rew+Unc+Gain* gaze strategy) is similar to the previous model, but it eliminates the tendency of fixating objects with high value that might not offer new task information. The model first predicts the expected value for each landmark, exactly in the same way as the previous model. But instead of fixating the landmark with the maximum expected value, each manipulation motor system selects the fixation that maximises the difference (or *gain*) between the prior and the posterior expected value. This is shown in Figure 1.6E, where fixating the blue sphere and the yellow cylinder give the maximum gain in return for the left and right arm respectively. This scheme allocates gaze to the manipulation motor system that most *gains* if it is given access to perception (similar to [43]). In the example, gaze is *allocated* to the left arm and the blue sphere is selected to be fixated (marked with an X).

Note that even in this simple example each gaze control model selected a different landmark to fixate next. Therefore, the behaviour of each gaze scheme is expected to

influence the performance of the task being achieved by the robot. Chapter 5 shows how the *Rew+Unc+Gain* gaze strategy is, in general, the most effective in terms of task performance and robustness. Then, Chapter 8 demonstrates that both the *Rew+Unc* and *Rew+Unc+Gain* gaze schemes fit existing human behavioural data.

1.6 Manipulation Tasks

One of the aims of our work is to study gaze control during the performance of tasks. In particular, this thesis focuses on manipulation tasks that require the use of, i) one arm, ii) two arms working separately, and iii) two arms working interactively. These tasks are used to experiment and characterise the gaze control models described above.

1.6.1 Pick & Place Task

This task has already been introduced above as an example in order to explain the models of gaze control in Section 1.5. The task consists of picking up objects from the table top and then placing them inside one of two containers. Figure 1.7A shows the iCub simulated robot performing the task. In this particular task the arms of the robot do not interact with each other (e.g. to transfer objects from one hand to the other). A new object appears at a random position on the table: i) every 60 seconds, ii) every time an object is put inside a container, and iii) whenever an object falls from the table. The robot has to look at objects and containers in order to get an estimate of their location. The goal is to put as many objects as possible inside the containers in trials of five minutes each. The aim of this task is to characterise the candidate models of gaze control, and to identify their advantages and their weaknesses (Chapter 5).

1.6.2 Bimanual Pick & Place Task

An extension of the previous task is to allow the arms of the robot to interact with each other by performing a bimanual task. The task is exactly the same as before, however the left container is removed. The robot needs to transfer the objects appearing on

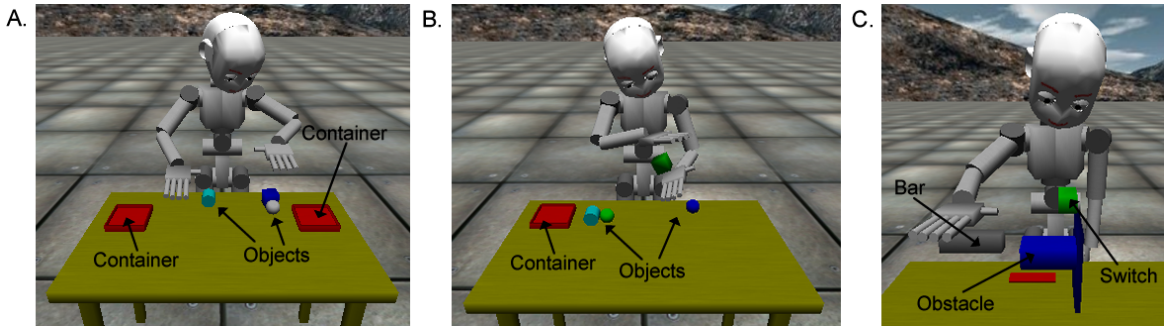

Figure 1.7: A. The iCub performing the pick & place task. B. Extension of the pick & place task that allows bimanual actions. C. Simulation of Johansson et al. [1] task. See text for details on all three tasks.

the left hand side of the table to the right side, in order to be placed inside the right container (Figure 1.7B). This transfer can only be achieved by passing the object from one hand to the other. The aim of this task is to further test the behaviour of the gaze control models, in a situation where there is at least one shared possible fixation location between the manipulation motor systems, i.e. the object to be transferred. This task is explored in Chapter 7.

1.6.3 Johansson's Task

This is a reproduction of the manipulation task devised by Johansson et al. [1] for a psychophysical study. The subject (or the robot, in our case) has to reach for and grasp a bar located on the table, then move the bar in a vertical plane in order to touch a target switch whilst avoiding an obstacle located between the table and the target switch. Once the switch is touched, the bar has to be put back onto the table (Figure 1.7.C). For Johansson et al. the aim of the task was to investigate spatio-temporal eye-hand coordination. For us, the aim of this task is to show how well our proposed models of gaze control match human behavioural data qualitatively and quantitatively (Chapter 8).

1.7 Thesis Contributions

The following is the list of contributions of this thesis:

1. It provides integrated models of gaze control for the problems of:
   (a) Where to look
   (b) Gaze allocation
2. It models systems:
   (a) With an oculomotor[8] system (or *perceptual* motor system)
   (b) Composed of multiple manipulation motor systems able to alter the physical world state
   (c) Capable of executing actions with variable time duration
3. It defines three new computational models of task-based gaze control that follow a one-step look ahead approach:
   (a) Gaze control based on uncertainty reduction (*Uncertainty*)
   (b) Gaze control based on rewards and uncertainty (*Rew+Unc*)
   (c) Gaze control based on rewards, uncertainty and gain (*Rew+Unc+Gain*) [57, 58]
4. It tests the models of gaze control using the iCub simulator performing a manipulation task. Each model is characterised in terms of task performance under variations of three environmental variables [58]:
   (a) Sensitivity in the manipulation actions
   (b) Observation noise
   (c) Field of view
5. It provides an integrated account of an *active visual search* process into the *Rew+Unc+Gain* gaze control.

[8] Abusing a little of the terminology, by oculomotor system we refer to a robotic pan-tilt camera system which may be monocular or binocular.

6. It demonstrates how the gaze control model based on task rewards, i.e. *Rew+Unc+Gain* and *Rew+Unc*, reproduce human behavioural data previously published by Johansson et al. [1].

1.8 Thesis Outline

The rest of the thesis is organised as follows:

Chapter 2: This chapter provides an overview of the mathematical foundations of the thesis. In particular, it describes decision making probabilistic processes, reinforcement learning, action selection mechanisms, and belief update mechanisms.

Chapter 3: The first part of this chapter presents an account of experimental findings from psychophysical and neurophysiological studies, which provided much of the inspiration of this work. The second part discusses and critiques different computational gaze control models.

Chapter 4: This chapter describes the robot's control architecture and provides a detailed formulation of our three gaze control models.

Chapter 5: This chapter presents a characterisation of our three gaze control models whilst performing the pick & place task. It describes the experiments, results and analysis of the three gaze control models in terms of three environmental variables. Also, two other models of gaze control are defined that serve as common baseline for comparison.

Chapter 6: This chapter describes the integration of an active visual search process into the *Rew+Unc+Gain* gaze control model. Experiments, results and analysis of this integration are also given using the same pick & place task.

Chapter 7: This chapter employs the bimanual pick & place task where both arms interact with each other, in order to show how the gaze control models are able to deal with these concurrent actions.

Chapter 8: This chapter presents how the gaze control models fit existing human behavioural data for the eye-hand coordination problem during a manipulation task

(Johansson's task) [1].

Chapter 9: The final chapter provides a summary of the thesis, the main conclusions and proposes lines of future work.

Chapter 2

Mathematical Foundations

The aim of this thesis is to formulate computational models of gaze control for an agent engaged in a manipulation task. Thus, the agent is required to make decisions about what to do in order to interact with the environment and fulfil the goal of the task. This chapter provides the mathematical foundations of decision making in stochastic processes[1]. First, a description of the general decision making problem faced by an agent is described. Second, Markov decision processes (MDPs), a common approach to modelling decision problems, are defined. Third, two extensions to MDPs are presented, partially observable MDPs and semi-MDPs (POMDPs and SMDPs respectively). Fourth, an account of reinforcement learning is given, which allows an agent to learn how to perform a task based on feedback in the form of rewards. Fifth, the selection of actions for the different Markov models is described. Finally, belief update techniques, which maintain the information state of the agent, are explained.

2.1 The Decision Making Problem

Before formal definitions are given, it is worth starting with a simple explanation of the decision problem faced by an agent attempting to accomplish some task. Figure 2.1 illustrates in a simplistic way the interaction between an agent and the world. In

[1]This chapter provides basic information about decision making processes. In particular, Sections 2.1 and 2.2 present general background information. Those with knowledge on these topics can skip this chapter without any loss in continuity.

Figure 2.1: Basic interaction between an agent and the world, where the agent acts according to what it perceives and the feedback provided.

general, the agent is expected to have sensors in order to obtain information about the world (e.g. the robot's cameras). This information is captured, processed and analysed in order to update the information state of the agent. This state describes properties of the world (e.g. the colour or the location of an object on a table). It may also be assumed that the agent is able to perceive its own state, e.g. the position of its hand. This ability is known as *proprioception*. Based on this information state, the agent should be able to perform some action using its actuators (e.g. grasping an object). Actions are likely to change both the state of the world and that of the agent. So essentially, the agent decides what action to perform based on the information acquired by its sensors. It is expected, of course, that the actions taken by the agent will accomplish some task. Also, it is preferable if the task is completed optimally with respect to some cost function (e.g. minimising the time to perform some task). One way in which the agent can improve its performance is by receiving feedback on the quality of its performance. This feedback can be generated by an observer acting like a teacher (e.g. a human), or the same environment.

Modelling decision making problems and finding optimal solutions is not easy, as the following sections will show. Several, sometimes unrealistic assumptions need to be made in order to model decision making problems. In some cases, optimal solutions cannot be expected to be found in practice, so near-optimal approximations need to be

found instead. First, one of the simplest decision making formalisations is presented, known as the Markov decision process.

2.2 Markov Decision Processes

A common approach to model decision making problems is to use *Markov decision processes* (MDPs) [59, 60]. The model is formalised as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function represented as $\mathcal{T}(s, a, s')$ being the probability of transition from state s to state s' after action a is executed, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function written as $\mathcal{R}(s, a)$ being the expected reward of taking action a in state s.

MDPs assume that the agent interacts with the world in discrete time steps. At time step t the agent observes the current state of the world s_t. According to that state the agent chooses to execute an action a_t. This action will produce a transition to a new state s_{t+1}, and the world will provide the agent with some immediate reward r_{t+1}. This reward serves as an indicator of the immediate goodness of the taken action a_t in state s_t. MDPs allow states and actions to have discrete or continuous values. Solving MDPs with continuous states is more complex, but solutions exist [61, 62]. Since the aim of this chapter is to provide basic information, only discrete spaces are considered.

The goal of the agent is to choose actions that maximise the expected discounted reward according to the state it is in. To do so we can define a *policy* π for action selection as $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which specifies a mapping from states to actions. The agent can calculate the value of a state s under the policy π by the expected discounted infinite-horizon sum of rewards. This is denoted as the *state-value function*[2] $\mathcal{V}^{\pi}(s)$:

$$\mathcal{V}^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(s, a) \left[\mathcal{R}(s, a) + \gamma \sum_{s \in \mathcal{S}} \mathcal{T}(s, a, s') \mathcal{V}^{\pi}(s') \right] \tag{2.1}$$

Where $0 \leq \gamma < 1$ is a discounted factor to indicate the trade-off between short term and long term rewards, $\mathcal{V}^{\pi}(s')$ is the state-value function of the next state s', and $\pi(s, a)$

[2] In this thesis we do not calculate the state-value function, however this information is provided as general background.

is a function that returns the value of performing action a in state s following policy π. Ideally, the aim is to find the optimal policy π^*. For this case the optimal state-value function is given by the maximum state-value under all possible policies:

$$\mathcal{V}^*(s) = \max_{\pi} \mathcal{V}^{\pi}(s)$$

$$\mathcal{V}^*(s) = \max_{a \in \mathcal{A}} \left[\mathcal{R}(s,a) + \gamma \sum_{s \in \mathcal{S}} \mathcal{T}(s,a,s') \mathcal{V}^{\pi}(s') \right] \tag{2.2}$$

Eq. 2.1 and 2.2 are known as *Bellman equations*. There are several algorithms to solve MDPs by means of these recursive equations. For instance, value iteration, policy iteration, linear programming and dynamic programming [59, 60].

It is also possible to define Bellman equations for state-action pairs. The *state-action value function* $\mathcal{Q}^{\pi}(s,a)$ is given by:

$$\mathcal{Q}^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') \sum_{a' \in \mathcal{A}} \pi(s',a') \mathcal{Q}^{\pi}(s',a') \tag{2.3}$$

Where $\mathcal{Q}^{\pi}(s',a')$ is the state-action value for the next state s' and next action a'. Similarly, the optimal state-value function is defined as:

$$\mathcal{Q}^*(s,a) = \max_{\pi} \mathcal{Q}^{\pi}(s,a)$$

$$\mathcal{Q}^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') \max_{a' \in \mathcal{A}} \mathcal{Q}^*(s',a') \tag{2.4}$$

Even though MDPs are able to model a large number of tasks, they are simplistic and make unrealistic assumptions. For example, if we are interested in modelling real-world tasks, MDPs make at least two unrealistic assumptions:

- Fully observable states: MDPs assume that the state of the world is completely known to the agent at all times. In a real setting this is practically impossible, e.g. a camera produces an image showing just part of the world, and even the image

may be noisy. In order to avoid this assumption, MDPs have been extended to what are known as partially observable Markov decision processes (Section 2.3).

- Actions with same duration: MDPs assume that actions take one time step to perform, however in a real setting this is not the case. For example, the time to reach for an object is not the same as the time to grasp an object. To overcome this limitation, semi-Markov decision processes have been defined (Section 2.4).

2.3 Partially Observable Markov Decision Processes

As mentioned above, MDPs assume that the state of the world is completely observable to the agent every time step. This means that the agent receives all the information needed to know exactly what the current state s is. Of course this assumption is unrealistic for a large class of tasks, in particular for real-world tasks. Typically, the agent's sensors are limited, which means that the agent only perceives part of the world. Moreover, the sensors can be noisy as well, which means that the perceived information is just a projection of the real state of the world.

Partially observable Markov decision processes (POMDPs) extend the MDP framework by considering *observations*, which specify incomplete measurements of the true state [60, 63, 64]. POMDPs are formalised as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{Z}, \Omega, \mathcal{R} \rangle$, where $\mathcal{S}, \mathcal{A}, \mathcal{T}$ and $\mathcal{R}$ are defined in the same way as for an MDP (Section 2.2). $\mathcal{Z}$ is the set of observations, and $\Omega : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$ is the observation function written as $\Omega(s, a, z)$ being the probability of making the observation z after being in state s and taking action a.

POMDPs allow the modelling of state uncertainty which covers more realistic tasks, however this comes with a price. Since the state s is not completely observable anymore, the agent has to estimate, based on the observations z, a posterior distribution over all possible states. This posterior distribution is known as the *belief state*, or *information state*, normally denoted as $bel(s)$. The problem is that the belief state is continuous, which makes it difficult to develop exact algorithms for solving POMDPs. Typically, the most common algorithms just approximate an optimal solution [60, 65, 66]. Section

2.7 provides a description of some of the techniques used to maintain the belief state, which in turn are used to solve the POMDP.

2.4 Semi-Markov Decision Processes

MDPs make the assumption that all actions take the same time to complete. Nevertheless, this assumption is impractical for real-world tasks. The semi-Markov decision process (SMDP) [59,67] model actions that take variable amounts of time to complete. SMDPs are formalised as the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{S}, \mathcal{A}, \mathcal{R}$ are defined as in the MDP model (Section 2.2). The main difference is that the transition function is defined as $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathbb{R} \to [0, 1]$, and can also be written as $\mathcal{T}(s, a, s', n)$. This function still denotes the probability of transition from state s to state s' when action a is executed, but here n specifies the completion time of action a.

2.4.1 Options

Options are a generalised and hierarchical approach to SMDPs [67,68], similar to macro-actions [69]. In her dissertation, Precup [70] developed a framework for control and learning in multiple temporal scales. In this framework, options are defined as temporally extended actions that can be composed of one or multiple single-step actions. In particular, our thesis is focused on *subgoal options*.

A subgoal option o is modelled as $o = \langle \mathcal{M}_o, \mathcal{I}, \beta \rangle$, where $\mathcal{M}_o = \langle \mathcal{S}_o, \mathcal{A}_o, \mathcal{T}_o, \mathcal{R}_o \rangle$ is a Markov decision process, $\mathcal{I} \subseteq \mathcal{S}_o$ is an initiation set that specifies the states where the option can start, and $\beta : \mathcal{S}_o \to [0, 1]$ defines a termination condition, which determines whether a particular state s_o is a terminal state or not.

Apart from modelling temporally extended actions, options allow the hierarchical decomposition of tasks. This is because an option as a whole can be treated as an action in a high level sub-task, whilst the set of actions defined within an option form another level. This is, in fact, the approach that this thesis follows as will be described in Chapter 4. Moreover, since options are derived from SMDPs, it is possible to employ

similar techniques to those used to solve SMDPs for options as well.

2.5 Reinforcement Learning

Reinforcement learning (RL) is a class of problems where the agent learns by trial-and-error interactions with the environment [34]. Typically, MDPs are employed to model such environment. Thus, the agent tries to approximate the optimal value function. As described above, MDPs, POMDPs and SMDPs require the definition of a transition function $\mathcal{T}$, which characterises the dynamics of the environment. However, RL is appealing as it does not require the definition of such model. The kind of RL approach that we follow in this thesis is an instance of *temporal difference* (TD) learning [71], which employ the Bellman equations defined in Section 2.2.

A fundamental problem with RL is the *exploration-exploitation* trade-off [72]. During learning, the agent is expected to expand its knowledge about its environment by executing as many actions as possible, even if those actions turn out to be sub-optimal (i.e. exploration). At the same time, the agent should make use of what has already learnt (i.e. exploitation), otherwise the agent may never learn how to behave optimally. There is no optimal procedure for the TD case, however it has been proved that by systematically controlling the exploration, the agent eventually learns the optimal policy [73, 74].

2.5.1 Q-Learning

One of the most popular model-free RL algorithms for MDPs is *Q-learning* [75]. The aim of this algorithm is to converge on the optimal action-value function $\mathcal{Q}^*(s,a)$ by means of the following learning rule:

$$\mathcal{Q}^*(s,a) \leftarrow \mathcal{Q}^*(s,a) + \alpha \left[r + \gamma \max_{a' \in \mathcal{A}} \mathcal{Q}^*(s',a') - \mathcal{Q}^*(s,a) \right] \tag{2.5}$$

Where $\mathcal{Q}^*(s',a')$ is the state-action value for the next state s' and next action a', $0 < \alpha \leq 1$ is the learning rate, r is the immediate reward received after executing action

a in state s, and $0 \leq \gamma < 1$ is a discounted factor to indicate the trade-off between short term and long term reward. The agent learns a greedy policy π with respect to the Q function. It is *greedy* because the policy selects in each state s the action a whose *Q-value* is largest, where the Q-value contains the estimated expected return for each state-action pair $Q(s, a)$.

2.5.2 SMDP Q-Learning

In the case of SMDPs, it is possible to learn policies by means of the *SMDP Q-learning* algorithm [76]. Here the learning rule defined in Eq. 2.5 above is extended in order to take into account the duration of each action:

$$\mathcal{Q}^*(s, a) \leftarrow \mathcal{Q}^*(s, a) + \alpha \left[r + \gamma^k \max_{a' \in \mathcal{A}} \mathcal{Q}^*(s', a') - \mathcal{Q}^*(s, a) \right] \tag{2.6}$$

All variables are the same as in Eq. 2.5, except for variable k that specifies the duration of action a. Similarly as for Q-learning, the agent learns a greedy policy with respect to the learnt Q-values.

2.6 Action Selection

The final goal of modelling decision making problems and the techniques to solve them, is to allow the agent to make decisions about what actions to perform in order to accomplish some goal or task. This section explains how action selection is achieved assuming the agent has previously learnt a policy π (e.g. using the RL algorithms).

2.6.1 Action Selection in MDPs and SMDPs

Assuming that the agent has access to the action-value function Q for an optimal policy, it is then straightforward for the agent to select optimal actions according to the current state. Since the state is fully observable, the agent always knows in which state s it is. The state s is used to index the Q function in order to determine which action produces the maximum value for that state:

$$a_t = \arg\max_{a \in \mathcal{A}} \mathcal{Q}(s, a) \tag{2.7}$$

Where a_t is the action chosen to be executed at time t. After this action is completed the state changes, and this state is once again available to the agent in order to index the policy. The same procedure for action selection is used for SMDPs as well.

2.6.2 Action Selection in POMDPs

As mentioned in Section 2.3, a POMDP decision model only receives an imperfect observation of the true state of the world. These observations are used to maintain a posterior distribution over the states called the belief state $bel(s)$ (Section 2.7 below presents ways in which the belief state can be maintained). Given a policy π, an action a is chosen according to the current belief state: $a = \pi(bel)$. However, in practice, POMDPs are computational intractable and approximate solutions need to be found instead [60, 65].

Littman [60] provides several approaches for representing policies that may be used to achieve the optimal value function. Cassandra [65] presents a RL framework for POMDPs based on the *neuro-dynamic programming* (NDP) approach developed by Bertsekas and Tsitsiklis [77]. The RL/NDP framework assumes that a POMDP can be restated as a continuous state space MDP. Still, it is not trivial to learn under this framework and other techniques have been employed.

A less mathematically motivated way to solve a POMDP is to use greedy heuristics that assume that the most likely state from the current belief state is actually the true state, thus ignoring partial observability [65, 66]. To do so, the underlying MDP is solved, which results in the policy π^{MDP}. The agent needs to keep track of the belief state, and on every time-step employ the current belief state to determine the most likely state the agent believes it is in. Cassandra [65] defines a heuristic policy for the most likely state (MLS) as:

$$\pi^{MLS} = \pi^{MDP}(\arg\max_{s} bel(s)) \tag{2.8}$$

The retrieved most likely state is then used to select the best corresponding action. However, there could be cases where a different action could provide more benefit from less likely states. In this thesis we make use of the *Q-MDP* algorithm for action selection [65, 78]. Instead of selecting just a single most likely state this algorithm uses the likelihood of every state to determine the best action to be performed:

$$a_t = \arg\max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} bel(s) Q(s, a) \tag{2.9}$$

Where a_t is the action to be chosen for execution at time t. The belief state $bel(s)$ is a probability distribution that determines the likelihood of being in some state s. The probability of being in state s is used as *weight* to determine the expected value of taking action a, according to the action-value function $Q(s, a)$ of the underlying MDP. The action a_t with the maximum value in the current belief state is selected. Note that this algorithm assumes that all partial observability disappears after the current action is chosen. This may be problematic if the current action does not provide much information about the state. However, as we will explain in Chapter 4, our gaze control model is the one dealing with the partial observability and it does not make such assumptions. Therefore, in our case, the interaction between the action selection module and gaze control seems to avoid the drawback of the pure Q-MDP heuristic strategy.

Cassandra [65] and Roy [66] present more heuristic algorithms following the same idea of ignoring the partial observability.

2.7 Maintaining the Belief State

As described in Section 2.3, in a POMDP the underlying state is not completely observable. POMDPs are required to estimate and maintain a probability distribution over all possible states based on observations and observation probabilities about the state of

the environment. This probability distribution has been already defined as belief state or information state. This section presents some techniques, known as *state estimators*, used to maintain the belief state. Basically, these estimators specify how to re-estimate the new belief state as new observations arrive [64].

Bayes rule is found at the core of state estimation algorithms. The rule is stated as:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \tag{2.10}$$

The rule provides a way to infer a hypothesis x from the observation y. The probability $p(x \mid y)$ is the *posterior* probability distribution, $p(y \mid x)$ is the *likelihood*, or the "inverse" conditional probability, $p(x)$ is the prior probability, and $p(y)$ is the normaliser and it is required that $p(y) > 0$. Belief distributions are in fact posterior probability distributions.

In this thesis attention is given to a group of state estimators called *Bayes filters* [79]. Following [64], Bayes filters can be divided into, i) parametric and, ii) nonparametric filters. One of the best known parametric filters is the *Kalman filter*, whilst in the case of nonparametric filters it is the *particle filter*. Both are explained in more detail next. For a recent survey of Bayes filters please refer to [79].

2.7.1 The Kalman Filter

The Kalman filter is a well-known algorithm, part of a family of estimators known as Gaussian filters [80]. The Kalman filter represents the posterior distribution as a Gaussian distribution, which is parameterised by its mean μ and its covariance Σ. One key aspect of Kalman filters is the assumption that the transition probability and the observation probability distributions are linear in its arguments.

A basic overview of the algorithm is as follows[3]. The algorithm requires as input the Gaussian belief distribution at time $t-1$, i.e. the mean μ_{t-1} and covariance Σ_{t-1}, along with the action (also called *control*) a_t, and the observation z_t. Roughly speaking, the

[3]Thrun et al. [64] provide an excellent detailed explanation of the algorithm and its mathematical derivation.

Kalman filter consists of two steps. The first step, or the *prediction step*, calculates a temporary belief state at time t by incorporating the action a_t. Then in the second step the observation z_t is incorporated to the predicted belief by some proportion specified by the *Kalman gain*. The output of the algorithm is the belief distribution at time t.

One disadvantage of Kalman filters is that they are restricted to systems with linear dynamics. However, there exists several variants that to a greater or lesser extent overcome this issue, e.g. the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [64, 79, 81]. Another disadvantage is that the posterior is represented as a Gaussian distribution, which means that they only approximate unimodal densities with some reasonable degree of fidelity. Nevertheless, the Kalman filter and its variants have been successfully applied in robotic vision and gaze control [7, 82].

2.7.2 The Particle Filter

As explained above, parametric filters represent the posterior distribution using a family of densities parameterised by, for example, the mean μ and the covariance Σ. An alternative to this approach is the use of nonparametric filters. They approximate the posterior by means of a finite number of values, where each value represents an instance of the state space. Furthermore, nonparametric filters are not restricted to unimodal belief densities, thus they can represent multiple hypothesis using multimodal distributions. One of the most common nonparametric filters is the particle filter [83,84].

A Particle filter $\mathcal{G}$ represent the belief state by a set of particles $\mathcal{G} = g^1, g^2, ..., g^J$, where J denotes the number of particles, and each particle g^j is an instantiation of the state. In other words, each particle corresponds to a hypothesis of what the true state may be. Similarly to the Kalman filter, the particles are updated after an action (or control) a is taken and a new observation z of the world is obtained. This update results in a new particle set, which represents the new posterior distribution. The basic algorithm can be divided in three steps[4]:

[4]For a more detailed explanation and a mathematical derivation of the algorithm please consult [64, 85]

1. **Action update:** This step creates a new temporary belief state $bel'(s_t)$ by sampling particles from the transition distribution. Given the current set of particles $\mathcal{G}_{t-1}$, each particle g_{t-1}^j and the action just taken a_t specify the probability of a new state g_t^j. This new state, i.e. the sampled particle, is added to the temporary belief state $bel'(s_t)$.

2. **Importance factor:** This step incorporates the new observation z_t by calculating the weight w_t^j for each of the new particles (from $bel'(s_t)$). This weight represents the probability of the new observation z_t given the particle g_t^j, i.e. $p(z_t \mid g_t^j)$. This step creates a set of weighted particles $\langle g_t^j, w_t^j \rangle$.

3. **Resampling:** In this step the new particle set $\mathcal{G}_t$ is obtained by drawing particles with replacement from the set of weighted particles $\langle g_t^j, w_t^j \rangle$. The weight w_t^j determines the probability of drawing the associated particle g_t^j into the new set $\mathcal{G}_t$. The size of the previous and the new particle set remains the same.

This thesis makes use of particle filters to keep track of the positional uncertainty of objects during manipulation tasks (described in Section 1.6). Therefore, the particle filter is further explained in Chapter 4.

Even though particle filters just approximate the posterior, as opposed to Kalman filters for example, they have become increasingly popular in robotics. This popularity is due to their functionality, flexibility and ease of implementation. They have successfully been applied to problems such as, localisation and mapping [86,87], fault-detection [88], and tracking [89, 90], amongst others. One of the main drawbacks is that particle filters cannot extend very well to high dimensional states, for this some variants have been proposed, e.g. Rao-Blackwellised particle filters [84, 91] and decentralised particle filters [92].

Chapter 3

Background

As mentioned in Chapter 1, this thesis pursues primarily an engineering science goal with the formulation of computational models of gaze control. Nevertheless, our proposed models are in part motivated by human behavioural and neurobiological findings. Thus, the aim of this chapter is to provide background information about gaze control from the biological and engineering perspectives. The first part of this chapter presents an account of active vision from the perspective of psychophysical and neurophysiological studies, along with a small overview of visual attention in humans. The second part of this chapter discusses models of gaze control and is further divided into three parts according to how fixation locations are selected: i) models based on image features alone (bottom-up models), ii) models that combine image features and task information, and iii) models that mainly employ task information or that are considered goal-oriented (top-down models). As our gaze control models are considered to follow a top-down approach, particular emphasis is placed on that last sub-section and on those models based on reward and for uncertainty. For completeness, the last part of this chapter explores other topics of gaze control that may not be directly related to the goals of this thesis but that are important to discuss, as they also face the problem of fixation selection.

Figure 3.1: A. Diagram of the human eye showing where the fovea is located (image taken without permission from [2]). B. The graph illustrates how the highest acuity is centred at the fovea, where the dotted lines indicate the edges of the fovea (2°) (image taken without permission from [3]).

3.1 Active Vision

Active perception, or active sensing, studies and models the intelligent control of information gathering mechanisms that acquire information according to the task being performed [9]. In particular, active vision is concerned with the study of gaze shifts and their effects in human cognition and perception [10]. Some of the questions that active vision attempts to answer are: Why and how do gaze shifts occur? How is the decision made about where to look next? What is the kind of information that is captured by the eyes? How is this information integrated between gaze shifts? Amongst several other questions [10,15,93]. This section presents evidence that attempts to answer these questions about active vision from two fronts: psychophysical and neurophysiological studies. Before this evidence is discussed, a taxonomy and explanation of human eye movements in general is provided, since these concepts are key to understand the active vision approach.

3.1.1 Eye Movements

One of the most important properties of the human eye is the inhomogeneity of the retina (Figure 3.1A), where only its central part, known as the *fovea*, has high acuity (or high resolution). Spatial acuity starts to reduce moving away from the fovea. More importantly, the fovea is just a very small region of the retina, with an angular diameter between 0.3°and 2°(Figure 3.1B) [37]. It is due to this property that the active nature

of the eyes emerges because the eyes must move in order to obtain a detailed view of the whole scene [10, 93]. The main reason for this foveated system is economical. If the whole field of view had the same acuity as the fovea, then the brain's processing power required to process such information would be immense [94].

Even though foveation requires constant movement of the eyes, these movements are rapid, efficient and economical. A common physiological taxonomy of eye movements is the following [10, 15, 95]:

Movements for gaze stabilisation: These are automatic and involuntary reflexes that are used to control gaze such that the fovea remains stable. There are two basic stabilisation movements, the vestibulo-ocular reflex (VOR) and the optokinetic reflex (OKR). These movements are further explored in Section 3.3.3 when the oculomotor control problem is discussed.

Fixational eye movements: When the eyes remain momentarily stationary a fixation occurs. Nevertheless, it has been found that the eyes do not remain completely stationary and three types of miniature movements may occur during a fixation: drift, tremor and microsaccade. Although it is still an open question, it has been suggested that one purpose of these movements is to stimulate the photoreceptors in order to avoid the retinal image from fading [96, 97].

Gaze shifting movements: These movements involve in some way the selection of a target, and three types of movements are defined:

- **Smooth pursuit:** Keeps track of a moving target by maintaining it in the fovea. The selection of the target is voluntary but it appears that smooth pursuit is not [15, 17].
- **Vergence:** Moves the eyes in opposite directions to converge into a target that moves in depth, or when shifting gaze to another target with different depth [10].
- **Saccade:** Shift gaze from one fixation to the next in rapid jump-like movements [10, 15, 37]. Saccades are central to the study of active vision and are

the type of eye movement employed in this thesis. Next, a more detailed description of saccadic eye movements is given.

3.1.1.1 Saccadic Eye Movements

Every time our eyes shift from one target location to the next a saccade is performed. Saccadic eye movements are rapid, taking around 30 to 40 ms, and have the properties of being *stereotyped* and *ballistic* [10, 37]. Stereotyped refers to the eyes following the same pattern of movement all times, i.e. they start to accelerate, reach a maximum velocity, and decelerate rapidly until the eyes are in the new target location. Ballistic means that the eyes cannot change their trajectory once the movement has initiated (but see [98] for a recent study of multisaccade planning). Saccades are voluntary movements driven by our goals, or involuntary, normally driven by particular visual features in the field of view (e.g. salient features such as motion or contrast). When a saccade is complete a fixation occurs. As mentioned above, even then the eyes do not remain completely stationary and miniature movements occur. Nevertheless, for practical purposes it is common to disregard these fixational eye movements. It is during a fixation that visual information is captured and processed, since vision is inhibited during a saccade (saccadic suppression) [37].

Even though the aim of saccades is to shift the fovea to obtain high resolution samples, *peripheral vision* has a significant role in detecting motion and targets that can be foveated next (e.g. in visual search tasks) [99–101].

Next, a series of findings from psychophysical and neurophysiological studies are presented. These findings provide evidence about the function of saccadic eye movements and how they are deployed. Particularly when humans are engaged in physical tasks (e.g. those that involve the manipulation of objects).

3.1.2 Psychophysical Findings

Psychophysical studies of eye movements in humans have demonstrated that fixation duration and gaze patterns vary according to the current cognitive goals or to the task

Figure 3.2: Gaze recordings of subjects whilst scanning a picture with different questions asked. A. *The unexpected visitor* painting. B. Free viewing. C. Remember the clothes of the people. D. Remember the position of the objects and people. Image adapted from [4] and taken without permission from [5].

being performed by the subject. One of the earliest studies to suggest this hypothesis was made by Yarbus [4, 5]. He studied subjects during a picture scanning task. The subjects were given different instructions whilst viewing the picture, e.g. to remember the clothes of the people in the picture, or remember the position of objects and people (Figure 3.2). Depending on the instructions the subjects' gaze pattern changed. This finding suggests that even with the same visual input, subjects selectively chose fixation locations according to the given task. In turn, this also suggests that saccadic gaze patterns may be linked to cognitive processes in the brain.

Experiments such as the one carried out by Yarbus helped to understand the importance of measuring gaze shifts and their role during the performance of tasks [16, 93]. Nevertheless, for several decades eye movement studies were conducted in constrained environments, with restricted movement of the subject's head and body. However, it is also in natural conditions and while performing daily visual tasks that eye movement should be studied [93]. It is thanks to recent advances in eye tracking technologies that experiments in natural tasks are now feasible (Figure 3.3A) [102, 103]. Examples of these unconstrained tasks include: picture viewing [104], reading [105],

Figure 3.3: A. Eye tracker used to record eye movements. B. Sandwich making task with superimposed gaze pattern, where the lines represent saccades, the circles fixations, and the circle size fixation duration. (Images taken without permission from [6])

sorting [21, 23, 106, 107], stacking [108], tea making [109], walking in virtual [24, 110], and real environments [111, 112], sandwich making [113] (Figure 3.3B), playing sport such as cricket [114], table tennis [115] and squash [116], hand-washing [117], and driving [22, 118], amongst other tasks.

The study of these natural tasks has provided a wealth of empirical evidence about the role of the task in the control of gaze, the spatio-temporal properties of eye movements, the relation between gaze and memory, the part that learning and planning play during gaze control, and the coordination between eyes and body. These findings are discussed in more detail next.

3.1.2.1 Eye Movements and Task

Perhaps the most important finding is that the task dominates how gaze is to be controlled. From picture viewing to driving, the goals of the task influence how gaze is deployed. As mentioned above, Yarbus found that the scanpath, or gaze pattern, in a picture viewing task changed according to the questions that subjects were asked (Figure 3.2) [5]. The same finding has been recently confirmed in another viewing task of colour photographs [104].

In a sandwich making task, task-irrelevant objects were placed amongst those needed to complete the task [113]. Before the task started the probability of looking at relevant and irrelevant objects was about the same. However, once the task started practically all the fixations landed only on task-relevant objects. Furthermore, the task not only

directs the movement of the eyes, but also the movement of the limbs and body.

3.1.2.2 Where to Look

In a foveated system, such as ours, it should be of great importance the place where fixations are made in the scene. As mentioned above, empirical evidence suggests that where we look is largely determined by the task. The fovea is practically always directed to objects, or parts of an object, that are relevant to the task [3,6]. Ballard et al. [21], in a block-copying task, investigated the ties between gaze and actions. It was found that fixations are always close to the site where actions are taking place. This is known as a *"do it where I'm looking"* strategy. Further experiments such as the tea and sandwich making tasks have confirmed such findings [109, 113].

It has also been found that the place where fixations are located is determined by the goal or the specific action being performed. Johansson et al. [1] measured precisely the landing sites of fixations during a simple manipulation task. Subjects had to move a bar in a vertical plane in order to touch a target switch with the point of the bar, whilst avoiding an obstacle located in the middle of the path. To avoid the obstacle fixations were rarely made to the centre of the obstacle, they landed in a space where both the tip of the bar and the obstacle, could be checked. But when grasping the bar, fixations landed on the bar itself. More recently, Rothkopf et al. [20] obtained similar results in a virtual navigation task, where subjects had to walk along a pathway whilst avoiding and colliding with tall rectangular objects. When objects had to be avoided, fixations were made to the lateral sides of the objects, whereas when the subject had to collide with an object fixations were made at the centre of the objects.

3.1.2.3 When to Look

Temporal aspects of gaze control have also been found from the study of natural tasks. In the block-copying task, Ballard et al. [21, 119] determined that fixations follow a *"just-in-time"* strategy. This means that, in general, gaze is deployed according to the needs of the task. More importantly, the measurement of gaze and body actions has

revealed that fixations precede actions (in some cases by less than a second [109]), and gaze moves to the next target once the current action is complete or almost finished. This suggests that gaze guides the actions by acquiring the information needed at the right moment [1, 3, 16].

Both the correct timing and the effective placement of gaze enable us to achieve tasks more efficiently, e.g. in terms of task performance or execution time. This has been better demonstrated when actions need to be more precise and accurate, for instance whilst playing sport. Land and Lee [22] showed how batsmen playing cricket are able to predict where the ball is going to bounce on the pitch. Subjects first fixated the bowler's release, with that information they predicted the trajectory of the ball. The second fixation was in the area where the ball was expected to bounce. Similar results have been found for table tennis [115] and squash [116].

3.1.2.4 Visual Analysis

Having discussed the spatio-temporal properties of gaze control, the study of natural tasks has also provided evidence about what to look at during a fixation. The aim of the visual system is to acquire information from the environment, where this information is typically task-relevant, as has been discussed so far. One question is, what kind of information is extracted during a single fixation. In turn, this question is directly related to the nature of memory. During the block-copying task, that consisted of copying a coloured block pattern from a template, Ballard et al. [119] found that whilst moving a block subjects performed a fixation to the template to check the colour, and then another one to check the location of the block. This suggests two things, i) that subjects prefer to look again rather than remembering these simple features, and ii) that each fixation is performed to extract specific information. The first hypothesis indicates that it is more costly to use memory than to perform an eye movement. For instance, Rensink [120] proposes to view the environment as an external memory that is accessed through eye movements. The second hypothesis is further investigated by studying the phenomenon known as *change blindness* [121]. This refers to subjects not noticing changes about

objects or features in the scene, whilst a saccade is performed, or because of blinking, or due to occlusions. Triesch et al. [23] performed a sorting experiment in a virtual environment where it was possible to change the size of the object being manipulated. They found that when the task did not involve the size of the object, even if the object was being fixated, subjects some times were unaware of the change in the object's size.

Ullman [40] has proposed a theory of *visual routines* that follows the idea that specific information is acquired during each fixation. The proposed visual routines are composed of a sequence of basic operators, where each routine is specialised in the extraction or processing of specific visual information. Roelfsema [42] describes a possible set of basic operators and their biologically plausible implementation in neurons. In a recent experiment, Rothkopf and Ballard [24] extracted image statistics at point of gaze during a navigational task in a virtual environment. It was found that gaze was directed at three distinctive features during the task. This provided quantitative evidence that specific features might be acquired on each fixation.

The analysis of fixation duration provides further evidence about the extraction of specialised information [6]. Depending on the task and the specific actions being realised, fixation duration may vary from tens of milliseconds to several seconds [109, 113]. This variation may be dependent on the information being extracted from the scene [122].

3.1.2.5 Uncertainty Reduction

As mentioned above, the eye-tracking experiments suggest that the goal of eye movements is to gather information about the world in order to act efficiently, e.g. to maximise task performance or minimise execution time. One way of information gathering is that the gaze is deployed in order to minimise or reduce uncertainty about properties of the environment or specific objects. Najemnik and Geisler [11] found quantitative evidence that humans follow an optimal fixation selection strategy in a visual search task, compared to an ideal Bayesian observer[1]. The aim of the ideal observer is to maximise the information gain about target location on each saccade. In a more recent

[1]For an explanation of Bayes' rule and Bayesian inference please refer to Section 2.7.

study, Najemnik and Geisler [123] confirmed the result that humans try to maximise the information gain about target location across fixations, in contrast to fixating at locations with high target probability only. Similar findings were obtained by Renninger et al. [36] in a shape-matching task. They used a shape-matching task to investigate whether humans move their eyes to locations that maximise *total* or *local* information about a shape to be matched. They found that subjects fixate the most informative locations or regions of these shapes, i.e. humans seem to reduce *local* uncertainty about the shape, and not the shape as a whole. This is similar to our models, since the robot reasons about the reduction of positional uncertainty of single landmarks, rather than trying to reduce positional uncertainty of several landmarks in one fixation. However, in a multi-target search task Verghese [124] observed that subjects tended to fixate at the most likely target location, instead of following an ideal information gain strategy. Thus, it is still not clear exactly what kind of strategy humans perform and further research must be done in this respect.

Note that the tasks just discussed did not involve body actions. During a driving task in a virtual environment, Sullivan et al. [26] manipulated the uncertainty about the speed of the car, which was shown in the car's speedometer. More fixations were made to the speedometer and fixation duration also increased when uncertainty about the speed of the car was introduced, in contrast with those cases with no added uncertainty. Another interesting finding was reported by Steinman et al. [25] during a precise tapping task. The subjects' visual acuity was degraded such that nearby targets were seen clearly and as the distance increased targets became blurred. The rather surprising finding was that fixation accuracy was better for the blurred targets, whilst large gaze errors were found on the clearer targets. The authors suggested that this shows that targets with no uncertainty do not need to be fixated directly, as opposed to those targets that have high location uncertainty.

3.1.2.6 Learning and Planning

The same psychophysical studies analysed so far have revealed that learning should take place at different levels in the brain [16]. At a high level, subjects must have learnt how to perform the given task. For example, in the tea and sandwich making tasks, subjects knew the steps required to prepare a cup of tea and a sandwich [109, 113]. At the same time, subjects must have learnt the complex dynamics of the environment and deploy gaze accordingly. For instance, skilled batsmen in cricket are able to fixate on the area where the ball is about to bounce [115]. Similarly, for navigational tasks such as driving and walking along a pathway, subjects learn the dynamics of cars, pedestrians, traffic lights, etc. [22, 110, 118]. It appears that subjects also learn how to control gaze efficiently, since there is a big similarity between subjects' scanpaths in practically all the natural tasks listed above [16]. This also suggests that models of the limitations of gaze and body should be learnt as well [15].

At the low level, however, planning seems to take place for gaze deployment and the execution of body actions. Even though subjects have learnt how to prepare tea or sandwiches, the location and, sometimes the appearance of objects, is likely to be different across instances of the same task. This suggests that low level motor commands are quite specific for the particular instance of a task. The same happens with the control of gaze. Fixations are made to places where relevant information can be acquired for the ongoing needs of the task, but these places are not always the same. Behavioural analysis suggests that the internal models about the world, the task and our own motor systems, that we have acquired with experience, are used for saccadic planning [15]. In a recent work, Ramakrishnan et al. [98] studied the case of multisaccade planning. It appears that, in some instances, subjects are able to plan two saccades in parallel. If the first saccade is heading towards an incorrect target, it can be cancelled and a second saccade guides gaze to another target.

3.1.3 Neurophysiological Findings

The previous section presented behavioural evidence that indicates how eye movements in general, and saccades in particular, are linked to cognitive goals in the context of a task. However, eye-tracking studies can hardly provide specific details about how eye movements are controlled by the brain. Which is why we now turn our attention to recent neurolobiological and neurophysiological experiments that provide clues about how the brain controls gaze during the performance of a task.

Perhaps the most important finding is that gaze and action control in the brain seem to follow a reward-based approach [16, 17, 28]. This finding mainly comes from studies of the basal ganglia, which consist of a group of strongly connected brain nuclei that function as a unit. It has been established that the basal ganglia are associated with various cognitive functions, such as motor control, learning and eye movements [27,125]. Schultz [30] has suggested that dopamine [126], one of the brain's neurotransmitters, is able to signal the reward prediction error of actions, which allows the basal ganglia act as a reward based system [127] (in contrast, Redgrave and Gurney [128] have suggested that dopamine allows for the discovery of novel actions rather than signalling the expected reward of known actions). Moreover, Hikosaka et al. [29, 129] have shown that saccadic eye movements are modulated by the basal ganglia to places where reward is available. They found that caudate neurons reflect expected reward before the target is fixated [130]. These findings suggest that behavioural control and learning, and saccadic control in particular, are modulated by the basal ganglia by means of reward signals.

An important consequence of these studies is the possibility of using mathematical models of decision making and reinforcement learning (Chapter 2) [34, 59] for modelling the internal workings of the basal ganglia and the eye movement circuitry [28, 131–134]. These mathematical models are key for the definition of gaze control models that will be discussed in Section 3.2.

Although dopamine is regarded as the main contributor to signal expected reward, other studies have also revealed the function of other neurotransmitters in the brain and their possible role in behaviour control and learning. Doya [32] presents a computational

theory of reinforcement learning based on four neurotransmitters: Dopamine, serotonin, noradrenaline and acetylcholine. In his model, dopamine represents the reward signal for reinforcement of actions; serotonin represents the trade-off between short and long term rewards; noradrenaline controls the trade-off between exploration and exploitation; and acetylcholine represents the learning rate (see Section 2.5). In a recent study, Seymur et al. [135] have found further evidence that serotonin encodes reward during decision-making. Another role for acetylcholine, which is of relevance to this thesis, is that of representing uncertainty [33]. Yu and Dayan [31] investigated uncertainty in a Bayesian statistical framework, where they use acetylcholine to represent expected uncertainty that comes from known unreliability in the environment. On the other hand, they also studied the role of the neurotransmitter norepinephrine, which may represent unexpected uncertainty that comes from completely unexpected changes in the environment. The interaction of these two neurotransmitters enabled inference and learning. Doya et al. [136] have further explored how a probabilistic Bayesian framework may be implemented in the brain.

The role of reward and uncertainty at the neural level motivates the creation of computational models of gaze and body control based on reward and uncertainty. Neurophysiological research, particularly in the last few years, have provided a wealth of information that is currently being used in neurocomputational models of decision making. Nevertheless, there is still much research to be done to determine the precise functionality of the brain's structures, in particular the basal ganglia, and their role in learning and planning eye movements and body actions.

3.1.4 Saccades and Visual Attention

Visual attention is, in a broad sense, a selective process which is intimately related to saccadic eye movements. So far it has been discussed that saccades are influenced by our current cognitive goals, and that visual information is acquired during a fixation. Thus it may be possible for us to say that our attention, whilst performing a task, is directed towards the current fixation location. However this is just half of the attentive process.

Overt visual attention is concerned with eye movements and the fixation location being attended, whilst *covert visual attention* is the ability to attend at a location in the periphery without moving the eyes [37].

Since our models of gaze control only employ saccadic eye movement, covert visual attention is not covered in full here[2]. Nevertheless, the interaction between overt and covert attention is important, since covert attention typically provides possible fixation locations to which the fovea can be directed in the next saccade(s) [138,139][3]. In Section 3.2 computational models of gaze control will be discussed. A few visual attentional models will also be analysed there since they can be used for the selection of gaze shifts.

An important question arising from visual attention is, "to what exactly is attention directed?" In other words, "what are the units of attention?" Yantis [140] provides a good review of the three major approaches found in the literature: i) location-based, ii) feature-based, and iii) object-based attention. If only overt attention is considered, Section 3.1.2 already presented behavioural experiments supporting all three approaches. For instance, in the sandwich or tea making task it is reported that subjects fixated the spatial location where an object was about to be placed [109, 113]. In the case of the block-copying task, it was discussed how subjects appear to focus on specific properties or features of a block whilst completing the task [119]. Finally, in all the tasks discussed so far, fixations (and attention) are mostly directed toward objects. It is safe to say that overt attention includes all three approaches. In this respect, this thesis follows mostly an object-based approach [141].

3.2 Models of Gaze Control

The previous section presented psychophysical evidence suggesting that the efficient control of gaze shifts is critical to accomplish visually guided tasks effectively. In addition, neurophysiological experiments were described that suggest that reward might play a major role in learning and planning the control of gaze and body actions, by following a

[2]For a survey of recent advances in covert attention please refer to [137].

[3]See Chapter 3 of Findlay and Gilchrist [10] for a discussion about the possible ways in which overt and covert attention interact.

Bayesian probabilistic approach. Therefore one question is: *What mechanisms should a rational agent*[4] *follow in order to select the next fixation location?* This section presents models of gaze control that provide answers to that question.

In very broad terms, the selection of fixation locations depends on whether or not the agent is engaged in some task. Thus far we have stressed the importance that the task has on the overall gaze control problem. Nevertheless, a large number of non-task based models of gaze control has been developed so it is important to discuss their properties. This discussion can also serve to highlight the importance of task based models. Typically, non-task based models are also known as *bottom-up* (data-driven or stimuli-driven) approaches, whilst task based models are also known as *top-down* (goal-oriented) approaches.

This section is divided into three sub-sections. First, bottom-up approaches are presented. Second, models that combine bottom-up and top-down processes are discussed. These models are a special case since the top-down information is mostly used to bias the bottom-up decision process. Finally, top-down models are described. We will see that these models may contain a bottom-up mechanism, however top-down information dominates the decision process. In this last sub-section particular emphasis is placed on those models that follow a reward based approach and those implemented in an embodied agent (e.g. a robot).

3.2.1 Bottom-Up Models

As mentioned above, bottom-up refers to those models that are not based on any particular task, i.e. the agent would be free to explore the environment. In this case, its gaze will be attracted to *salient* or *conspicuous* features of the visual scene [142]. For instance, features or attributes of high contrast in terms of colour, motion, orientation, size, depth and luminance [143]. These salient features constitute possible fixation locations to which the agent can look at any time step. Therefore, the agent's decisions are mainly influenced by the environment. It is worth noting that most of the bottom-up

[4]In this context, an agent could refer to a virtual or physical, human or robotic subject.

models described here have been studied from the visual attention perspective [144], and not to explain saccadic eye movements. However, as mentioned above in Section 3.1.4, models of visual attention could also be employed to select fixation locations if only overt attention is considered.

Koch and Ullman [142] proposed a way in which the salient features could be globally represented by means of a *saliency map*. Later on, Itti and Koch [145] defined a computational model to produce such saliency map. Basically, the most salient region in the map would have a higher weight, which means that the agent's attention (or its gaze in our case) would be drawn towards this region by following a *winner-take-all* (WTA) strategy. For this strategy to work, the saliency map must avoid to fixate or attend again at the recently visited regions by means of a special property called *inhibition of return* [146], which decreases the weight of a region for some period of time after it has been attended. Psychophysical evidence, during picture viewing tasks, supports the idea of a saliency map in humans [147]. Recently, saliency maps have been extended in order to calculate saliency in time varying input such as video, rather than just static picture viewing scenarios [148–150].

Robotic platforms have also been equipped with saliency detection models. For example, Butko et al. [151] implemented a saliency model in robotic pan/tilt camera in order that is able to maintain the most salient region of the current scene in the centre of the camera. Schlesinger et al. [152] incorporated a bottom-up model in the iCub simulator in order to study the perceptual development of human infants, using the psychological experiment of perceptual completion. Vijayakumar et al. [153] employ a stimuli-based model in a humanoid robot as an attention mechanism. The robot's oculomotor system moves and centres its cameras on the salient regions of the visual scene.

Finally, it is worth mentioning that some recent saliency models attempt to follow the neural characteristics of the brain mechanisms known to be part of the selective visual attention process [154–156].

3.2.2 Bottom-Up + Top-Down Models

The main problem with bottom-up models is that, if the agent is engaged in a particular task, they will not necessarily be able to detect regions of interest that are task-relevant [20, 122]. Therefore, a logical extension to purely bottom-up models is to incorporate some kind of top-down mechanism in order to use task information whilst calculating the saliency map. This sub-section presents models that are essentially bottom-up models but that are biased by top-down information. These models mostly come from the visual attention literature [157, 158], as those described in the previous sub-section.

The implementation of these models typically follows the same structure, where the saliency map is created based on image features and at the same time, or in a later step, contextual task information is added to the map. These top-down cues will bias the saliency map depending on the task being performed. Perhaps the most commonly used contextual cue is the target object information [159–161], where the saliency map is biased towards specific target features. For example, higher weight may be incorporated to red blobs in a scene if we are interested in finding stop signs in roads. Another important cue comes from the context given by the scene, which provides likely locations of the target(s) [159,162]. For instance, when looking for pedestrians it is more likely to find them in sidewalks or roads, than on top of buildings. Recently, some models have combined saliency with the probability of expected rewards [13, 14][5].

For robotic platforms, information about the target object(s) is also commonly employed. Orabona et al. [163] implemented a visual attention model in a humanoid robot, combining saliency and object-based cues. They also simulate the inhomogeneity of the human retina by transforming camera images to log-polar images. Models of the target objects are learnt off-line using different viewpoints of each object. These models bias the saliency map in order to find a particular target. A similar mechanism is presented by Hawes and Wyatt [164], where a robot can be asked questions about the location and properties of certain objects. The robot also learns a model of each object, but in this case learning can occur on-line. Bohg et al. [165] employs target information to

[5]For more information about this kind of models please refer to Borji and Itti's recent survey [158].

detect objects for grasping. In this case, the model of the object is encoded by a neural network. Once the object is detected and recognised the robot performs the grasping action. Finally, Frintrop and Jensfelt [166] dealt with the problem of self-localisation and mapping (SLAM) for a mobile robot. Regions detected by a saliency detector are used as landmarks, then as the robot moves, it has to keep track of these landmarks by means of feature vectors, in order to localise itself.

3.2.3 Top-Down Models

As mentioned above, there is evidence that purely bottom-up models are not able by themselves to predict gaze patterns if the agent is engaged in some task [20, 122]. The main reason is that the object(s) or spatial location(s) that are considered task-relevant might not be conspicuous or salient enough for the agent to fixate on them. This is why top-down biases have been incorporated into saliency models [157, 158], as explained in the sub-section above. The models presented in this sub-section select fixation locations mainly based on the properties and features of the task at hand, rather than being driven by the stimuli captured in the image(s) [15]. Thus, the majority of the gaze control models presented next are formulated using decision-theoretic approaches (some of these approaches were described in Chapter 2).

Even though our proposed gaze control models follow an engineering approach, we first start by describing biologically grounded, or biomimetic, models of gaze control. These models are restricted to follow some of the neurobiological structures of the brain and their interaction. It is important to talk about these models since a big part of the motivation of this thesis comes from biological findings. So the models described here build on the findings described above. The second part describes computational models that may be biologically motivated, but that follow engineering approaches, such as ours. Finally, we talk in more detail about the model of gaze control developed by Sprague and Ballard. As explained in Section 1.5, our models build directly from theirs.

3.2.3.1 Top-Down Neurocomputational Models

As presented in Section 3.1.3, saccadic eye movements and action selection are modulated by the basal ganglia [30, 129], and it has been suggested that the basal ganglia can be represented as a reward based system [32, 127]. Following these findings some researchers have developed computational models of action selection and gaze control mimicking the functionality of brain modules, in particular the basal ganglia, the cortex and cerebellum. The brain and its modules work at different levels, thus computational models typically work at specific levels [167].

An agent may engage in a task at two different phases, the *learning* and the *proficient* phases[6] [168,169]. The learning phase is where the agent acquires the mapping between stimuli[7] and actions, whereas the proficient phase is where the agent selects or decides what action to perform based on the learnt mapping.

The learning phase is typically carried out by means of external reward, and can be formulated with reinforcement learning algorithms [34]. As explained above, Doya [32] presents a neurocomputational reinforcement learning algorithm based on the basal ganglia and the neurotransmitters: Dopamine, serotonin, noradrenaline and acetylcholine. In his model, dopamine represents the reward signal for reinforcement of actions; serotonin represents the trade-off between short and long term rewards; noradrenaline controls the trade-off between explorations and exploitation; and acetylcholine represents the learning rate (see Section 2.5). On the other hand, Bogacz and Gurney [168, 170] present a model for action selection assuming the agent already has learnt the task. They model a statistically optimal action selection mechanism at the system level of the basal ganglia and the cortex. Actions are selected following a similar approach to a Bayesian decision-making mechanism (Section 2.7), where this selection is based on noisy inputs. Recently, Bogacz and Larsen [169] have provided an integrated account of reinforcement learning and action selection based on noisy sensory input. In this respect, their model is actually similar to partially observable Markov decision making processes (POMDPs)

[6] In this thesis the proficient phase is called *execution phase*, as it will be seen in Chapter 4.
[7] This is also known as the information state as explained in Chapter 2.

(Section 2.3), which are used in robot control [65]. In this same line, Rao [171] explicitly formulates a neural model of action selection based on POMDPs, where decisions are made in the face of the uncertainty caused by unreliable observations (or evidence). He shows that this model matches some results from behavioural and neurophysiological studies of information gathering behaviour. Both, Bogacz-Larsen's and Rao's model, are similar to ours in that they re-value rewards in the face of uncertainty.

Regarding decision making models, an important line of research has been the modulation of the trade-off between decision speed and accuracy [170]. Ideally, decisions should be made accurately, but due to noisy sensory input the agent would need to accumulate evidence over time before making an accurate decision, thus affecting the decision speed. In some circumstances, this is not possible and decisions need to be made fast even under uncertain information. For instance, an animal that has just detected a possible treat could try to make a more accurate decision about whether the treat is real or not by trying to look for further evidence (thus risking its life), or it could just move away from the possible treat (although it might not be a real treat). Trimmer et al. [172] formulate two sub-systems, one fast but that might be inaccurate, and another one slow but more accurate (i.e. a sub-system that integrates evidence over time). They provide probabilistic mechanisms in which these two sub-systems can be combined by making different assumptions about their interaction. A more neurobiologically grounded implementation has been recently presented by Marshall et al. [173].

The basal ganglia have also been shown to modulate saccadic eye movements [129]. Building on Bogacz and Gurney's model of decision making in the basal ganglia [174], and in the neural saccadic generator mechanism developed by Gancarz and Grossberg [175], Chambers et al. [41] present an oculomotor control system that performs reactive saccades. Positive feedback is employed within the oculomotor system in order to saccade to a salient target that suddenly appears whilst the system fixates another target. The aim of this system is to compare the abnormalities present in people suffering Parkinson's disease. Another model of saccade generation is proposed by Krishnan et al. [176], which follows a reinforcement learning approach. The model performs sac-

cades to those parts that are more rewarding based on a saliency map. Finally, Ognibene et al. [177] present a neural architecture for a camera-arm robot following a reinforcement learning approach for a reaching task. Their model does not explicitly follow the structure of brain modules, however, the arm control is driven by the stimuli present in the camera's saliency map.

Even though in this thesis we do not follow a neurocomputational approach, the models just described are interesting in that they provide a way to see how the brain might make decisions about action selection and gaze control at some level of abstraction.

3.2.3.2 Top-Down Engineering Models

In this sub-section we will review models that follow a top-down approach from an engineering perspective. First, we start by describing models that are implemented using only a virtual or a robotic camera. These models provide interesting formulations but they are mainly used in visual search tasks for object detection/recognition. Second, we review gaze control models for embodied agents (virtual or robots). These models are similar to ours as they are employed to perform tasks that involve the direct manipulation of or interaction with the environment.

Aragon-Camarasa et al. [178] present a binocular vision system that is able to search for and recognise target objects in cluttered scenes by means of SIFT features (Scale Invariant Feature Transform) [179]. Rather than following a decision-theoretic approach, the next fixation point is determined by the SIFT features of objects, thus avoiding calibration and geometric knowledge of the camera. Similarly, Forssen [180] learns how to saccade and center specific image features in a binocular system. This is done by learning, via a forward model, how image features move when a gaze shift occurs, where the image features are also SIFT features.

Renninger et al. [181] follow an entropy based approach by selecting the next fixation location as the one that minimises uncertainty. They consider single images for a shape-discrimination task, and assume peripheral-foveal vision (i.e. only the fovea has high acuity). Another entropy based model is that of Arbel and Ferrie [182], which selects

view points with the aim of maximising the amount of information about a 3D object in order to recognise it. They make use of a camera, rather than just a 2D image, that is able to move anywhere around the 3D object. The next view point is determined based on an entropy map associated with that particular object, by specifying the places where more observations are required. Gould et al. [183] simulate a peripheral-foveal vision system with one low-resolution fixed camera and a pan-tilt-zoom camera for high-resolution images. Their task is to recognise and keep track of objects already recognised. Gaze is allocated to one of these two tasks by calculating the entropy over the objects being tracked and those that have not been identified. For recognition, the next fixation location is determined by identifying pixels that may contain a new recognisable object, this is done by a previously-learnt probabilistic model of each object. The identification of these pixels is made by the low-resolution camera. For tracking, the system can decide to look back at one of the objects being tracked in order to update their position and velocity.

Building on the work of Najemnik and Geisler [11] (discussed in Section 3.1.2.5), Butko and Movellan [184] propose a model called I-POMDP, an information-gathering (infomax) approach to visual search using a POMDP (partially observable Markov decision process) (see Section 2.3). They assume peripheral-foveal vision and employ small visual arrays. The key idea is to select the next fixation location that maximises the likelihood of finding the target, determined by the current belief state about the target's location. This is an interesting Bayesian approach that is able to integrate information across saccades, which is similar to the way we followed. However, they learnt the POMDP and in turn, where to look. This is possible for visual search tasks confined to small search areas, and because there is no external reward for physical actions as in our case. Recently, Butko and Movellan [185] implemented the I-POMDP model as part of a "digital eye" that is tested in visual search tasks with larger images for face recognition. They also provide a more general approach to the visual search problem and how the next fixation is determined by several sources of uncertainty, e.g. uncertainty about the target, the eye movement, the sensing of objects, and motion produced

by the eye movement. In [186] they implement a learning mechanism in two real robots that allows them to learn how to look at a target taking into account the uncertainty of the motors and sensory information.

Vogel and Murphy [46] modelled eye movements using POMDPs for visual search tasks. This model is simulated using pre-acquired images of different scenes. Nevertheless, the model involves pan and zoom actions, and the POMDP models noisy observations about the location of the target. Contrary to Butko and Movellan's model, the next fixation location is planned according to the likelihood of finding the object in the current view point. Also, planning occurs at different horizons (lookahead planning). This model was later extended [187], in order to incorporate bottom-up saliency and scene context information.

Minut and Mahadevan [188] learnt gaze patterns by means of reinforcement learning (RL) (Section 2.5), using a pan-tilt camera, and for a visual search task set in an office environment. The next fixation location was decided based on the most salient point from the current view point. Positive reward was given if the target object was located closer to the centre of the camera. The main drawback of this model is that it has to re-learn the gaze pattern if the target object is moved to a different location. Also following an RL approach, Stober et al. [189] modelled a foveated retina that learns a policy to center the fovea on points of interest. The idea is to directly map motor commands to sensor features. The model is tested in simulation and in a pan-tilt camera using synthetic and natural images.

Sridharan et al. [44, 190] present a framework for planning visual operations on single images based on hierarchical POMDPs. Even though they do not explicitly decide fixation locations, some of the visual operators determine the object that is being processed, and in consequence, that is being fixated. The hierarchy employed allows them to make POMDP planning tractable.

None of these models of gaze control employ an embodied agent or robot. They can model the dynamics of objects [185], but the agent is not able to manipulate or interact directly with the environment. Next, we will review gaze control models for agents with

one or more motor systems that allow them to perform physical actions, similar to the models we formulate in this thesis.

An evolutionary approach for active vision is presented by Floreano et al. [191, 192]. A mobile robot with an attached pan-tilt camera is able to navigate in an indoor and outdoor environment by means of an evolved neural network. The robot learns where to look and navigate simultaneously. The pan-tilt motor values are directly learnt according to the sensory input and the current trajectory of the robot.

Recently, Borji et al. [193] developed top-down attentional mechanisms modelled as Dynamic Bayesian Networks. These models were compared to human data from subjects playing video games (e.g. a driving task) by predicting their eye fixations. Physical actions performed by the subjects, such as the wheel position, gear change, pedals, etc. are considered by the model in the form of object properties. In fact, the models follow an object-based approach, as opposed to spatial models (Section 3.1.4).

Seekircher et al. [194] formulate an entropy-based model of gaze control for a humanoid robot for self-localisation and ball tracking in a soccer field. In a similar way to one of our proposed gaze strategies (the *Uncertainty* gaze scheme (Section 1.5)), the next gaze location is determined by maximising the expected information gain in the next time step. Something that we do not consider in our models but which they do is the cost of performing a saccade, by measuring the difference between the expected and current camera angles. The robot maintains its belief state using particle filters, and the state is based on image features of the environment. The main difference with our models is that we control gaze by considering the physical actions that are being performed, rather than just features from the environment. Another entropy-based model for the soccer domain is presented by Kohlbrecher et al. [195]. In addition to self-localisation and ball tracking, they also account for obstacle detection. A vision-based occupancy grid map is generated in real-time that fuses different sources of information (static and dynamic knowledge about obstacles). Based on this map an entropy-based map is created and employed to determine the best next view point, i.e. the view point with the maximum entropy.

Karaoguz et al. [35], present a saliency and reward based approach in a multi-tasking scenario where a simulated humanoid robot has to reach for an object whilst attending to an interaction partner. Their visual architecture consists of several modules that keep track of colours, motion, objects that have not been visited for some time and the interaction partner. The robot learns weights, via a reward mechanism, to determine which module should get access to perception according to an activation map created by each module. The activation maps specify the desired motor commands and they are represented as 2D Gaussian distributions. Whilst this model can be employed for multi-tasking, gaze control is learnt rather than planned, and only one motor system is considered.

An interesting manipulation scenario is presented by Schenck et al. [196]. They investigate the precision of a robotic arm in grasping foveated and non-foveated objects. As one would expect, the robot arm has a high grasping success rate for those objects that are fixated precisely. For non-foveated objects a prediction model is learnt using neural networks. They show the success rate of grasping non-foveated objects by using a prediction model. They show how the success rate of grasping non-foveated objects by using the prediction model is much higher than if the model is not used. Of course, foveating the object is better than using the prediction model.

Erez et al. [45] proposed an interesting model for hand-eye coordination in a simulated reaching task using a POMDP. In the task, two "hands" must reach to a common point whilst passing a pair of obstacles. The joint behaviour of eye and hands is modelled in the POMDP, where states and actions can have continuous values. Having such a high-dimensional space, approximation techniques are needed in order to plan the corresponding hand-eye movements. The model is compared to human data obtained by subjects performing the same reaching task by means of thumbsticks (from a game controller) and eye tracking. This model proposes an interesting way to coordinate gaze and actions for two arms (there are other works that map directly eye movements with a single arm using sensorimotor mapping of coordinates [197]). In this thesis we avoid having such high-dimensional space, and direct mapping between eye and hands, by

splitting the decision problem into physical and perceptual action selection.

3.2.3.3 Ballard's Model of Gaze Control

This thesis is largely motivated and builds on the work of Ballard, and in particular on Sprague and Ballard's model of gaze control. In an early paper Ballard introduced a theory of gaze control following an "animate vision" approach [19], where the agent has the ability to actively fixate towards task relevant visual stimuli. This theory argues that gaze shifts allow *early* visual computations (i.e. the construction of retinotopically-indexed maps of, for example depth, colour or velocity) to be less expensive, compared to a passive vision system [198]. This is mainly because in animate vision the frame of reference is fixation point centred, rather than camera-centred, which allows the coordinate system to have zero velocity and zero disparity at the fixation point. Animate vision also proposes that agents should have a set of visual behaviours, these refer to task-specific or special-purpose algorithms. For example, a simple visual behaviour could consist of a colour detector algorithm, or an edge detector algorithm. The use of behaviours simplify the computation by executing those behaviours currently needed by the task. Ballard also suggests that these visual behaviours should be learnt, for example using reinforcement learning.

Whitehead and Ballard [199] implemented animate vision using a reinforcement learning approach in a block stacking task. The system makes use of *deictic* representations and actions that significantly reduce the the information that is stored in memory. Rather than keeping track of all possible objects and their features from a given scene, it is better to just keep track of those objects relevant to the task. Furthermore, an object to be manipulated could be referred to as a deictic representation, i.e. "manipulate the object I am fixating". This is the *"do it where I'm looking"* strategy observed by Ballard et al. in the block-copying task in humans [21]. Furthermore, Whitehead and Ballard's model consider the idea of visuo-motor behaviours, where each behaviour is expected to perform a sub-task or a specific sub-goal. Their model is based on two markers known as: *action-frame* and *attention-frame*. The action-frame performs perception and phys-

ical actions, whereas the attention-frame only performs perception (in order to gather new information). Thus, the set of available actions is determined to what the agent is currently fixating or attending. For instance, in the block stacking task there could be some number of blocks, so rather than having a specific action for each block, the actions are in the form of "grasp-object-at-action-frame". Their system is able to learn policies to solve specific stacking tasks (i.e. specific visuo-motor behaviours), however one disadvantage is that it does not generalise well.

Following the idea of visuo-motor behaviours, Salgian and Ballard [200] developed visual routines [40][8] for a driving task and provided a scheduling mechanism for their execution. Each driving behaviour could be associated to one or more visual routines. For example, the *stop-sign-behaviour* has associated the *stop-sign-detection* and *intersection-detection* routines. Behaviours are implemented as finite state machines. The idea is that several behaviours may be running simultaneously, this implies that several visual routines may be competing for execution time. In other words, as gaze is a serial resource, visual routines can only be executed one at a time. They test the performance of several scheduling rates along with variations on the visual behaviours (another scheduling mechanism for gaze allocation is implemented by Hernandez et al. [201]).

This work leads to the gaze control model formulated by Sprague and Ballard [7,43], which is important to our work since this thesis builds on their model. In particular, our model of gaze control based on rewards, uncertainty and gain (the *Rew+Unc+Gain* gaze scheme explained in Section 1.5). Therefore it is important to explain in more detail Sprague's model, and delineate the similarities and differences with our models.

The model proposed by Sprague and Ballard is a reward based approach for the problem of gaze allocation in a multi-tasking domain (similar to Salgian and Ballard's above). They consider a simulated human agent with the task of following a sidewalk whilst avoiding obstacles and collecting litter. The agent is expected to perform these three sub-tasks *concurrently*[9] by controlling its steering direction; the virtual agent keeps

[8]Recall that Ullman's theory of visual routines follows the idea that specific visual information can be extracted during each fixation.

[9]Concurrency here is defined at the task-level, where a single action affects all sub-tasks being executed. In contrast, in our work (see Chapter 7) we define concurrency at the level of the manipulation

on walking during the task. The agent, therefore, is restricted to have only one motor system. The sub-tasks can be run simultaneously because they share the same action space (three steering directions). The agent first learns each sub-task via reinforcement learning (Section 2.5), under an assumption of complete observability. This means that the simulator provides the exact location of the sidewalk, the obstacles and the litter to the agent during the learning phase. For example, for the obstacle avoidance sub-task, depending on the distance between the agent and the closest obstacle, the agent may steer to the right, left or keep walking straight. The distances to the sidewalk, obstacles and litter constitutes the information state of the agent.

After the learning phase, the agent executes the three sub-tasks simultaneously. But now the agent must maintain the information state of each sub-task by extracting visual information from the scene using a visual routine. Based on the information state of each sub-task the agent performs an action that affects all sub-tasks, since the action space is shared amongst the sub-tasks. The serial nature of gaze is modelled by allowing only one visual routine to be executed every time step. Their key idea is to allocate (or assign) gaze to the sub-task that is likely to lose more reward, based on the current state uncertainty, if it is not given access to perception. Only the sub-task that has access to perception is able to extract information from the image relevant to it using its corresponding visual routine. The sub-tasks that do not have access to perception have increased uncertainty about their objects of interest.

Figure 3.4 shows an overshot of the simulated agent performing seven time-steps of the three sub-tasks. In the figure, obstacles are the blue rectangular objects whilst litter is represented by the purple cylinders. Figure 3.4.A presents how gaze is allocated, notice that for the first three time-steps gaze is allocated to the obstacle avoidance sub-task, then the agent attends the sidewalk, and then the litter object. Figure 3.4.B illustrates the location uncertainty of the obstacle (OA), the sidewalk (SF), and the litter object (LC). The beige region represents the location uncertainty before perception, whilst the coloured regions is the location uncertainty after perception. They compare

motor systems, where a concurrent action is accomplished by the coordination of both arms.

Figure 3.4: A. Overview of the simulated agent performing the navigation task. Obstacles are the blue rectangular objects, whilst litter is represented by the purple cylinder. The lines represent the fixations made by the agent during seven time steps of the task. B. The estimated location uncertainty for each sub-task during the seven time steps, where OA = obstacle avoidance, SF = sidewalk following, and LC = litter collection. The beige regions represent the uncertainty before perception, whilst the coloured regions represent the uncertainty after gaze is allocated to that sub-task. (Images taken without permission from [7])

the performance of their reward based model against Random and Round Robin gaze schemes, and show that the reward based approach obtains in average more reward.

There are several important differences with respect to our models of gaze control:

1. We deal not only with the *gaze allocation* problem but also with the problem of *where to look* (see Section 1.3). In Sprague's model the visual routines are hand-coded so that the closest obstacle or litter is the one being attended; in the case of the sidewalk the agent always attends the leftmost part. Thus, Sprague's model does not deal directly with the problem of where to look.

2. By dealing with the problem of *where to look*, we must consider a separate oculomotor system in order to direct gaze to specific fixation locations. Sprague's

agent does not have the ability to move its eyes independently from its body (as the iCub does (Section 1.4)), depending on the steering direction the agent will change its view point.

3. We model multiple manipulation motor systems, as opposed to one, as in Sprague's system. However, our manipulation motor systems execute a single sub-task each at any one time, whilst Sprague's only motor system is able to execute multiple sub-tasks concurrently.

4. We model manipulation actions with variable duration, as opposed to actions with a single time step.

5. A key difference is that our gaze control models reason about the remaining uncertainty that would result after a fixation location is selected, as opposed to Sprague's model that assumes that all uncertainty disappears after a fixation is made.

6. We consider manipulation tasks, as opposed to a navigation task.

7. We present the formulation of three models of gaze control based on reward and uncertainty, whilst Sprague presents only one gaze control model.

8. We incorporate an active visual search process into the gaze control mechanism in order to find new objects in the scene. As mentioned above, Sprague's agent does not have the ability to move its eyes independently from its body, so the agent's physical movement determines the current view point.

The similarities between our gaze control models and Sprague and Ballard's model are:

1. The separation between gaze control and physical (manipulation) action selection, which allows us to learn under the assumption of complete observability. This means that we also consider a learning phase, where all information is assumed to be known to the agent, and a subsequent execution phase, where the information is maintained by extracting visual information.

2. The comparison of our models with *Random* and *Round Robin* gaze strategies.

3. We follow the general idea of allocating gaze to the sub-task that loses more reward for our gaze model based on rewards, uncertainty and gain (*Rew+Unc+Gain* strategy) (Sections 1.5 and 4.5.3).

3.3 Other Related Gaze Control Topics

The previous section presented a number of gaze control models with the aim of selecting fixation locations. Still, gaze control has been also studied and implemented in other fields. This section presents a summary of three of these other fields. First, *visual servoing* is discussed in order to point out the differences and similarities with the aims pursued by this thesis. Second, we explore gaze control in the field of *human-computer and human-robot interaction*. Finally, we provide some information regarding *oculomotor control* since it is essential to the overall gaze control problem.

3.3.1 Visual Servoing

In robotics, visual servoing has been a very important topic that deals with the control of the robot using visual feedback in a closed-loop manner [202, 203]. Therefore, it is important to delineate the similarities and differences with the kind of control we deal with in this thesis.

Visual servoing has been typically applied to manipulation tasks, although more recently it has also been applied for walking in humanoid robots [204]. The idea is that the environment is being tracked and image features are extracted in order to guide the manipulator to a specific position. The typical configuration of a visual servoing system involves a manipulator and one or more cameras. The camera(s) could be placed: i) on the manipulator itself (known as *eye-in-hand*), ii) placed in a fixed location in such way that the field of view covers the robot's working space, or iii) a combination of the previous two [203]. In this respect, active gaze control as it has been defined so far, is different to visual servoing as the former deals with the selection of fixation points,

whilst the latter is typically fixed or unable to select different viewpoints dynamically, as it is possible to see enough, or all of the workspace to solve the problem.

On the other hand, this thesis is also concerned with the guidance of manipulation actions by extracting visual information and transforming it into 3D pose information (see Section 4.4.1). In this respect, although our gaze control model is different, our visual analysis and state update mechanism are similar to that employed by *position-based visual servo systems* [202].

3.3.2 Human-Computer and Human-Robot Interaction

Sociological studies have demonstrated the importance that gaze has during social interactions [205]. Eye contact and where we look has proven to be crucial for a successful interaction between two or more persons. This evidence has been recently employed in the field of human-robot interaction (HRI), with the aim that humans may engage more easily with robotic agents. To do so, robots keep track of the gaze direction of the human partner and then their gaze is subsequently directed towards the same location. This is important, for example, when the human acts as a teacher, or just in a casual conversation [206–208]. During the opposite case, when the robot leads or initiates the conversation, its gaze should be used to guide the human towards points of interest [209]. This last situation has also been explored for virtual agents [210]. Finally, the way in which humans visually engage with user interfaces has also been studied in order to predict human behaviour during human-computer interaction (HCI), which helps in better user interface design [211].

3.3.3 Oculomotor Control

In Section 1.2 it was explained how gaze control could be decomposed into three different processes: i) fixation selection, ii) oculomotor control, and iii) visual analysis. This thesis is solely about the *fixation selection process*. This sub-section presents some information about the oculomotor control process, since both problems are interrelated.

From a human perspective, oculomotor control is responsible for the movement and

stabilisation of the eyes. These movements are those described in Section 3.1.1 above: saccades, smooth pursuit, vergence, vestibulo-ocular reflex (VOR), and optokinetic reflex (OKR) [10, 37]. A robotic camera system, which can be part of a humanoid robot for example, should be able to perform practically the same movements as the human eye [38]. The camera(s) should move to specific locations in order to acquire images. These movements could be slow or fast, although fast movements should be desirable so that a new image (i.e. new information) can be captured quickly. If the movements are fast then stabilisation of the camera becomes a critical issue. Recent work has focused on biomimetic oculomotor control for the development of robotic oculomotor systems [38,212–214]. These oculmotor systems have dealt with the integration and combination of different movements found in the human eyes. Normally this also requires an embodied agent, because the stabilisation reflexes must take into account body movements [38, 212, 213]. Other works consider the robotic camera system alone [214–216].

Oculmotor control is essential for an active vision system, and gaze control depends on good oculomotor control mechanisms. This is why it is important to provide a quick overview of such system. This thesis, however, relies on an existing robotic oculmotor control mechanism developed specially for the iCub humanoid robot [39].

3.4 Summary

This chapter provided background information about the problem of gaze control presented in three main parts. The first part reviewed research in eye movement literature. As has been stated before, even though this thesis has as primary goal an engineering approach to the problem of gaze control, our work has been motivated by biological findings from the fields of psychology and neuroscience. The second part of this chapter reviewed specific mathematical models of gaze control. Here special focus was placed to the work of Ballard, as our models build in part on his research. Finally, the third part reviews other research areas where gaze control is also studied but because of their aims may not apply directly to our work.

Chapter 4

Modelling Task-Driven Gaze Control

This chapter describes in detail the formulation of the three gaze control models introduced in Section 1.5. These models of gaze control are driven by the task being performed by the agent (or in the case of this thesis, by a simulated humanoid robot). Embodied agents, such as humans and robots, achieve tasks by interacting with the environment (e.g. using manipulation actions). These actions can change the state of the robot and that of the environment. As discussed in Chapters 1 and 3, one of the aims of deploying gaze is to acquire relevant task information and to guide the manipulation actions. Therefore, the gaze control problem should not be treated in isolation, but rather as part of the robot's control architecture (Figure 4.1). This architecture encodes assumptions which apply across all the models of gaze control that will be formulated. The operation of this architecture is divided, temporally, into the *learning phase* and the subsequent *execution phase*. To facilitate explanation of the system, this chapter starts by modelling the pick & place task (introduced in Section 1.6). Then the common architecture is described, and finally the different gaze control models are formulated.

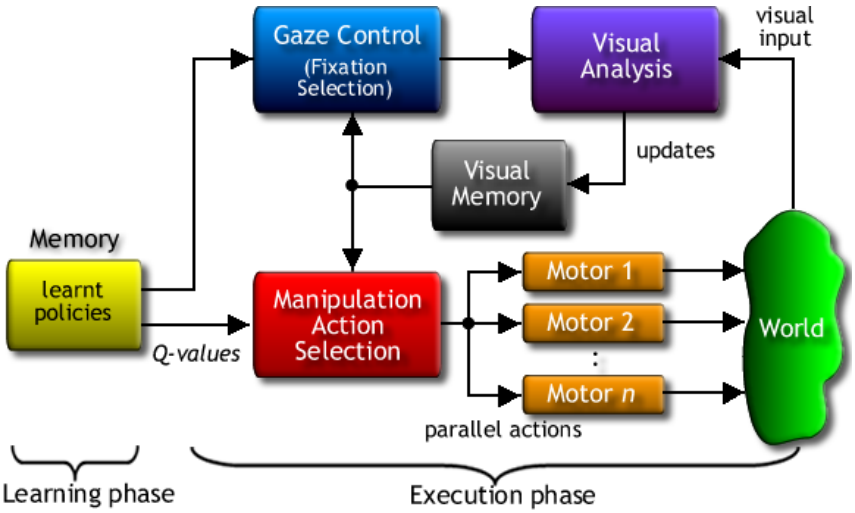

Figure 4.1: Schematic view of the robot's control architecture. The system is divided temporally into the learning phase and the execution phase. See text for details on each control module.

4.1 Modelling the Pick & Place Task

This section describes how tasks are modelled in our system using as an example the pick & place task. As described in Section 1.6, the pick & place task consists of picking up objects from the table top and then placing them inside one of two containers (Figure 4.2 shows the robot performing the task)[1]. This task considers two manipulation motor systems: the right and left arm. We assume that the task is divided into two sub-tasks, each assigned to one arm (since for this task the arms do not interact with each other). Each sub-task (or manipulation motor system) is modelled as a semi-Markov decision process (SMDP)[2] (Section 2.4) using a discrete state representation.

Therefore, each manipulation motor system $ms \in MS$ (where MS is the set of motor systems), is modelled as a tuple $\langle \mathcal{S}_{ms}, \mathcal{O}_{ms}, \mathcal{T}_{ms}, \mathcal{R} \rangle$, where $\mathcal{S}_{ms}$ is the set of discrete states, $\mathcal{O}_{ms}$ is the set of manipulation actions, $\mathcal{T}_{ms} : \mathcal{S}_{ms} \times \mathcal{O}_{ms} \times \mathcal{S}_{ms} \times \mathbb{R} \rightarrow [0, 1]$ is the transition probability density, where $\mathbb{R}$ is the set of real numbers representing the duration time of each manipulation action, and $\mathcal{R} : \mathcal{S}_{ms} \times \mathcal{O}_{ms} \rightarrow \mathbb{R}$ is the reward

[1] In this task the arms of the robot do not interact with each other (e.g. to transfer objects from one hand to the other). A new object appears at a random position on the table: i) every 60 seconds, ii) every time an object is put inside a container, and iii) whenever an object falls from the table. The goal is to put as many objects as possible inside the containers.

[2] As explained in Section 2.4, SMDPs are an extension of the Markov decision processes (MDPs) [59] that allow actions to have variable duration.

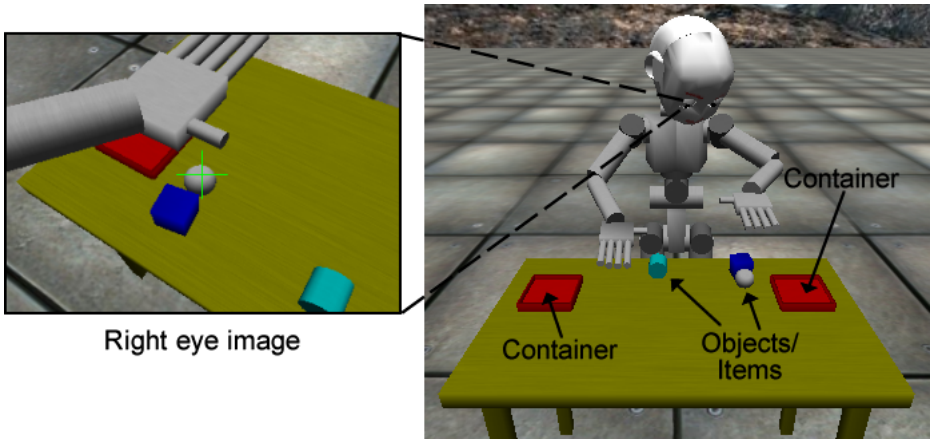

Figure 4.2: The iCub simulator performing the pick & place task where objects should be picked up and subsequently placed inside the containers. The image on the right shows a snapshot captured by the right eye of the robot.

function, that is assumed to be the same for all manipulation motor systems.

For the pick & place task the set of manipulation motor systems is $MS = \{right\_arm, left\_arm\}$. Table 4.1 depicts the factorised discrete state space for both arms ($\mathcal{S}_{right\_arm}$ and $\mathcal{S}_{left\_arm}$), where each arm has three state variables. Notice that these state variables do not consider the location of objects/containers. This information is in fact continuous, and is handled by the *visual memory* explained below in Section 4.2. Thus, there is a mapping (explained in Section 4.4.2) between the continuous location information, used by the manipulation actions, and the discrete state space.

Table 4.1: Factorised state space for the right and left arms.

State Variable	*State Value*
armPosition	{onObject, onTable, onContainer, outsideTable}
handStatus	{grasping, empty}
tableStatus	{objectsOnTable, empty}

Because we consider that each manipulation action can take variable duration, the set of manipulation actions is modelled following the concept of *options* [67] (see Section 2.4.1), where an option can be used to model temporally extended actions. Each option should be learnt or planned, however we do *not* do this as we use the commands available in the iCub libraries for the low-level control of each manipulation action[3]. Nevertheless,

[3]These commands are implemented using PID-like Cartesian motor controllers [55]. The arm controller accepts as input 3D coordinates that specify the desired final position of the centre of the hand. Furthermore, the iCub arm controller has been designed to produce smooth human-like trajectories.

we still use the concept of options to indicate that our manipulation actions have variable duration and the idea that these actions can be learnt or planned.

For the pick & place task the set of manipulation actions (i.e. *options*[4]) for both arms ($\mathcal{O}_{right\_arm}$ and $\mathcal{O}_{left\_arm}$) is defined in Table 4.2, where each set consists of six options. The completion time of each option is also shown in the same table, indicated as Gaussian distributions with their mean and standard deviation (μ_o, σ_o) specified in seconds. The completion time distributions were obtained by executing all manipulation actions in sequence for 60 minutes. The completion time for each manipulation action varies because of noise in the motor controllers and because of positional changes (i.e. actions do not start exactly from the same position at all times). The time it takes for an option to complete at the current time step is denoted by k_o.

Table 4.2: Manipulation actions for the right and left arms.

Options	*Avg. completion time (secs)*	*Stand. deviation (secs)*
moveToObject	2.65	0.56
moveToTable	2.95	0.45
moveToContainer	3.25	0.33
graspObject	2.87	1.08
releaseObject	1.0	0.2
noAction	0.01	-

The manipulation of objects is not a straightforward problem, particularly grasping. In this work we have simplified the problem of grasping by using a magnet-like command available in the iCub simulator. When activated, this command creates a virtual link between the centre of the hand and the centre of the specified object. In that way the object will move as if it were part of the robot's arm until it is released. Even though grasping is simplified, it is still possible to control the sensitivity of reaching and grasping by checking the offset between the centre of the hand and the centre of the manipulated item. So grasping and reaching options are said to "fail" (except for *releaseObject*) if that offset is greater than some threshold (e.g. 1.0 cm). How manipulation actions are determined to "fail" or "succeed" is explained in Section 4.4.2. The iCub motor controllers have a limited positional accuracy, therefore the minimum effective value of

[4]Throughout this chapter we will use the term option and manipulation action interchangeably.

that threshold is set to 0.5 cm.

4.2 Visual Memory

The central component in the system is the *visual memory* ($\mathcal{VM}$) (see Figure 4.1), defined as the set of ordered pairs $\mathcal{VM} = \langle (e_1, pos(e_1)), (e_2, pos(e_2)), \ldots, (e_n, pos(e_n)) \rangle$, where e_i is the i^{th} landmarks's id of interest, where a landmark can be an object or a container in the pick & place task. On the other hand, $pos(e_i)$ is a probability density function for the location of landmark e_i. These pairs capture the continuous state information needed for low-level control of each option, i.e. for the execution of manipulation actions. Subsequently, this information is used to set the discrete values of the state variables defined above (i.e. $\mathcal{S}_{right\_arm}$ and $\mathcal{S}_{left\_arm}$). $pos(e_i)$ can also be thought of as the representation of the location uncertainty of the landmark e_i. The location of a landmark refers to its centre projected in the X-Y plane in robot coordinates (objects are 4 cm in diameter and length, whereas containers are 10x10x3 cm in width, length and height).

In this work, each landmark's location ($pos(e_i)$) is approximated by a particle filter [64][5] (see Section 2.7.2). A particle filter contains a set of particles $\mathcal{G}_i = g^1, g^2, ..., g^J$, where J denotes the number of particles, and each particle g^j represents a possible location *(x,y)* for landmark e_i. Figure 4.3 shows a graphical representation of the visual memory and its contents. In this example, the robot has information about four landmarks (two objects and both containers). The figure shows each landmark e_i along with its corresponding particle set $pos(e_i)$. The spread in the particles represents the location uncertainty of that landmark. For instance, the location uncertainty of the first container ($pos(e_1)$) is higher than the rest of the landmarks.

There are four basic operations that are employed to maintain the visual memory:

- **Addition of landmarks:** During the execution of the task, when a new land-

[5]In this thesis we make use of particle filters, rather than the Kalman filter for example, mainly because the uncertainty about the location of landmarks is not restricted to be Gaussian. Also, the probability for the location of a landmark may be multimodal. Even though particle filters approximate the posterior, they allow for more flexibility.

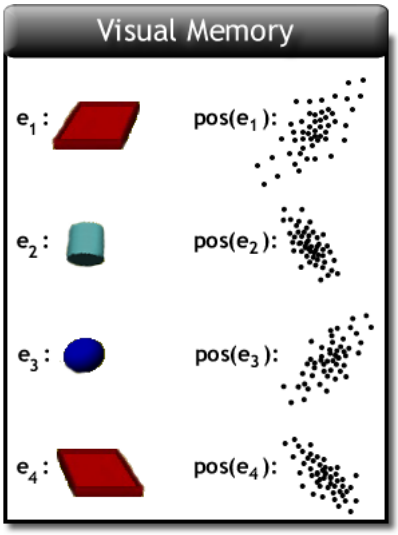

Figure 4.3: Representation of an instance of the visual memory which contains four landmarks (two objects and both containers). Each landmark has associated a particle set that represents the location uncertainty of that landmark.

mark is inside the robot's view point (i.e. when the robot "sees" or detects a new landmark on the table), that landmark is added to the visual memory and its corresponding particle filter is created. The particle filter is initialised by distributing the particles uniformly across the table.

- **Landmark updating:** Every time that a landmark that is already stored in visual memory appears again in the robot's view point (i.e. the landmark is "seen" or detected again), its particle set is updated using the new detected location.

- **Information retrieval:** The robot decides what manipulation action(s) to perform and how to control gaze based on the information stored in the visual memory. The robot can retrieve information about a particular landmark using its id.

- **Removal of landmarks:** When an object is put inside a container or when it falls from the table its pair $(e_i, pos(e_i))$ is removed from the visual memory.

Section 4.4.1 will explain in more detail how the particle filters are maintained, in order to add and update landmarks in visual memory.

4.3 Learning Phase

As described above, a SMDP models a particular sub-task, and learning to behave in each sub-task is achieved via reinforcement learning using *SMDP Q-learning* [76] (Section 2.5). Each manipulation motor system ms learns a policy $\pi_{ms} : \mathcal{S}_{ms} \rightarrow \mathcal{O}_{ms}$, that defines a mapping from states to options (i.e. manipulation actions), and contains the estimated expected return for each state-action pair (known as *Q-value*). These values quantify each manipulation action (introduced in Section 1.5).

In this phase the robot learns how to perform the task under an assumption of *complete observability*, i.e. the visual memory contains the complete list of landmarks (e_i) and their true location[6]. Therefore, the density function $pos(e_i)$ becomes a Dirac delta function with value 1 at the true location (i.e. an infinitely thin spike at the centre of the landmark).

For learning, a minimal time to goal strategy is followed. For any action taken the robot receives -1 units of reward, and 0 units of reward when all objects are put in the containers. During learning the number of objects appearing on the table is limited to 10. Objects appear in random locations across the table, however during this phase this is not a problem since their locations are given to the robot. Note that the same policy can be used for both arms in the pick & place task. The learning rule is formulated as:

$$\mathcal{Q}(s,o) \leftarrow \mathcal{Q}(s,o) + \alpha \left[r + \gamma^{k_o} \max_{o' \in \mathcal{O}_{ms}} \mathcal{Q}(s',o') - \mathcal{Q}(s,o) \right] \tag{4.1}$$

Where $0 < \alpha \leq 1$ is the learning rate, r is the immediate reward received for executing option o in the discrete state s (where $s \in \mathcal{S}_{ms}$), $0 \leq \gamma < 1$ is a discount factor to indicate the trade-off between short-term and long-term rewards, $\mathcal{Q}(s',o')$ is the state-action value for the next state s' and next option o', and k_o is the duration of option o. For the pick & place task, $\alpha = 0.2$, $\gamma = 0.9$, and k_o takes the time values associated to each option defined in Table 4.2.

[6]The assumption of complete observability allow us to separate the processes of manipulation action selection and fixation selection. Otherwise, we would have to learn how to control gaze whilst learning to perform the task. In this case the state space would have to be continuous and it could grow too big for learning.

4.3.1 Learning Simulation Programme

One of the disadvantages of the iCub simulator is that it only works in real time. Thus, it would take a long time to learn any policy even for simple tasks. Another issue is that during learning the robot is expected to perform exploratory manipulation actions which might crash the simulator, although this could be solved to some extent by implementing safety mechanisms. Due to these issues, we decided to implement a separate learning programme that is able to accelerate the simulation of manipulation actions.

This programme implements the SMDP Q-learning rule (Eq. 4.1) and simulates the changes in the state and the manipulation actions (i.e. options). As explained in Section 4.1 above, the time distributions for each manipulation action were determined using the iCub simulator by executing the actions in sequence for 60 minutes. These time distributions are specified by the mean and standard deviation (μ_o, σ_o), shown in Table 4.2). Therefore, in order to determine the time it takes for a simulated manipulation action to finish, we sampled its corresponding time distribution. Thus k_o takes the sampled time value. After a manipulation action was executed we simulated the change in the state of the world by changing the value(s) in the state variables defined in Table 4.1).

The output of the learning programme is the *Q-table* that contains the values for each state-action pair. Besides the advantage of accelerating the learning process, this learning simulator also eases the evaluation and analysis of the learnt policies, as they can be tested before they are employed in the iCub simulator.

4.4 Execution Phase

Once the policies for each sub-task have been learnt and stored in the robot's memory, they can be used to solve the task, provided there is state certainty. However, the robot is now in charge of maintaining the visual memory, i.e. the robot has to look at the landmarks in order to estimate their location. Hence, the robot has to decide what to do under uncertain information. To estimate the location of a particular landmark the

robot needs to *fixate* that landmark. By fixation we mean that the robot moves its "eyes" (i.e. cameras) so that the landmark appears in the centre of its field of view. For example, Figure 4.2 shows the robot fixating the grey sphere. As introduced in Section 1.4, we make use of what we call the *perceptual motor system*, which includes the robot's head, neck and eyes. As in the case of the manipulation actions, we employ a motor controller implemented in the iCub libraries that is able to move the eyes, neck and head of the robot to achieve the desired fixation location [39]. The desired fixation point is specified in 3D coordinates.

First, this section starts by explaining how the visual memory is maintained using a kind of Bayes' filter by making use of an observation model learnt off-line. Maintaining the visual memory is part of the *visual analysis* module shown in Figure 4.1. Second, we describe how the robot performs the *manipulation action selection* for each motor system under uncertainty. Finally, we explain how the *fixation selection process* is done for each gaze control model in turn.

4.4.1 Visual Analysis

At the beginning of the execution phase the visual memory is empty. When the robot performs a fixation, visual input is captured and the landmarks inside the field of view (FoV) are detected and added to the visual memory, whilst their 3D locations are estimated. The estimation of 3D locations can also be referred to as *triangulation*. Using stereo vision in the simulator provided precise 3D locations, however this is hardly the case in a real setting. Since our models assume positional uncertainty of landmarks, we are interested in having a noisy triangulation process. Instead of using artificial noise, we found that triangulation is in fact noisy in the simulator using a *command* that employs monocular vision (right camera). This command uses the distance from the eye to the landmark's true location, but because the true location is not known this distance needs to be *estimated* using the current knowledge about the landmark's location. Another source of noise comes from the positional error in the robot's head, neck and eyes in the simulated odometry.

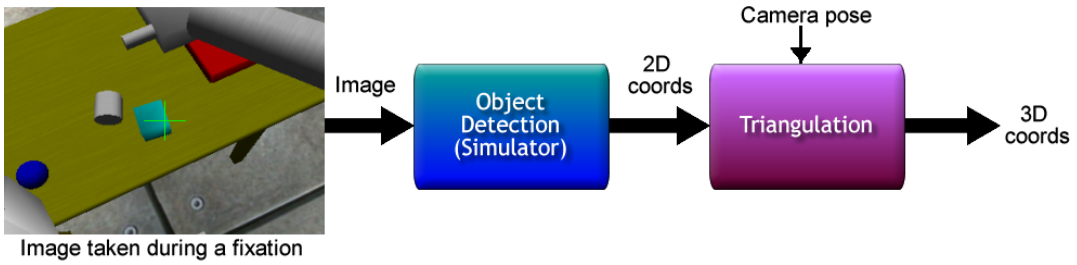

Figure 4.4: Schematic representation of the visual analysis employed to estimate the landmarks' 3D coordinates.

Figure 4.4 illustrates the visual analysis process. For the detection of landmarks we ask the simulator the 2D coordinates of those landmarks that are inside the FoV. By using the simulator we effectively have a noiseless object detection mechanism. This allow us to focus on just one source of uncertainty, namely the estimated 3D locations[7]. In order to represent the uncertainty that results from the triangulation process an observation model was learnt off-line.

4.4.1.1 Observation Model

The aim of the observation model is to characterise the uncertainty that results from the triangulation process. Several versions of the observation model were developed in order to determine the parameters that affect the location estimate. In one version of the observation model, an object was systematically moved across the table, whilst the robot's gaze remained fixed on the centre of the table. This observation model was not useful because the location of the fixation point also affects the location estimate of a landmark. In a second version of the observation model, both an object and the fixation point were systematically moved across the table. This observation model is good for our pick & place task, however if the position of the table changes the observation model could no longer be used. In order to generalise the observation model, three parameters need to be considered: i) the position of the camera, ii) the fixation point, and iii) the object's location.

The final version of the observation model was created by systematically moving

[7]In future work, it is possible to add more sources of uncertainty and analyse the effects that each source has in the system (Chapter 9).

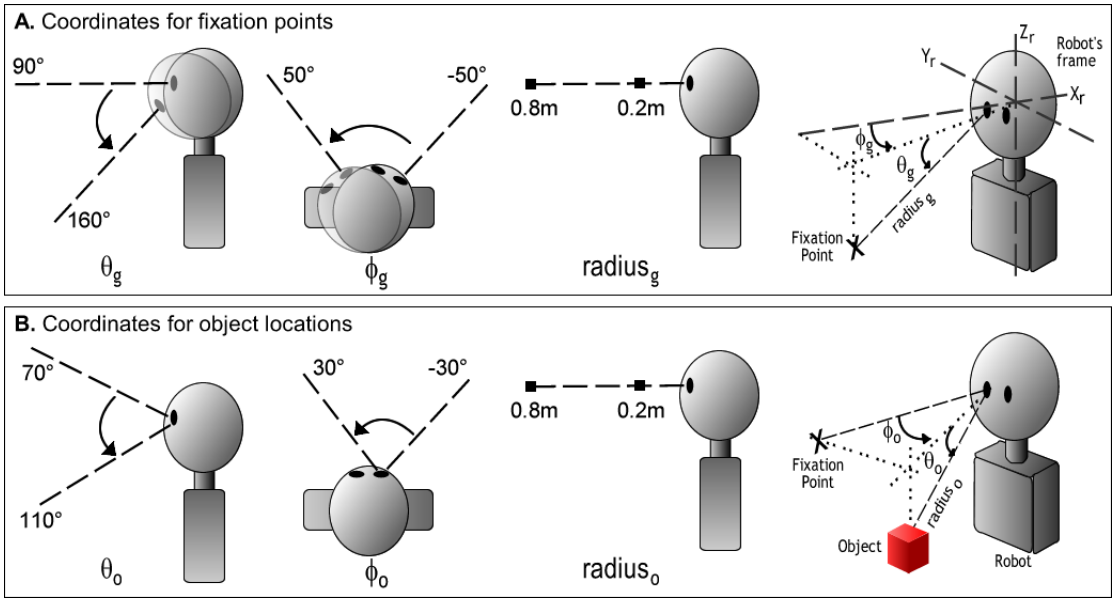

Figure 4.5: A. The coordinates and values used to specify fixation points $(\theta_g, \phi_g, radius_g)$. B. The coordinates and values used to specify object locations $(\theta_o, \phi_o, radius_o)$. These coordinates are used to learn the observation model (see text for details).

the robot's fixation point and an object in a discretised space. Each fixation point and object location is represented using spherical coordinates: $(\theta_g, \phi_g, radius_g)$ and $(\theta_o, \phi_o, radius_o)$ for the gaze and object respectively. The values used to create the gaze pattern are shown in Figure 4.5A. During each fixation point the object was moved in such way that it always appeared inside the camera's FoV. The values used to move the object are shown in Figure 4.5B. Whilst varying these parameters we recorded the offset between the estimated 3D location (using the simulator's triangulation *command* mentioned above) and the true object's location. Instead of using the distance to the table's centre for triangulation, the observation model is generalised by triangulating using the distance at which the fixation point is located $(radius_g)$. Thus, each data point is represented by the rest of the coordinates except for $radius_g$. The resulting data points were binned according to a quantisation of $(\theta_g, \phi_g, \theta_o, \phi_o, radius_o)$ and then fitted by bivariate Gaussian densities. This resulted in 405 densities that cover a big part of the robot's working space.

As an example, Figure 4.6 shows nine of these densities for the case when the robot is looking at the centre of the table $(\theta_g = 140°, \phi_g = 0°, \theta_o = 70°)$. Notice how the noise accrues as the object horizontal eccentricity in the FoV increases, as ϕ_o and $radius_o$

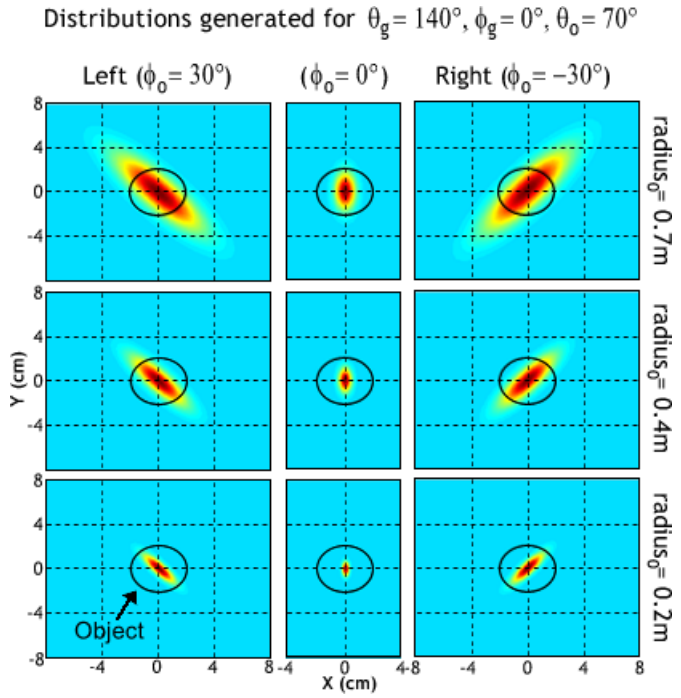

Figure 4.6: Examples of nine distributions generated by the observation model representing the uncertainty about the triangulation of an object (represented by a circle in the graphs). The robot is looking at the centre of the table ($\theta_g = 140°$). The uncertainty in the object's location increases as the object moves away from the robot (columns), and as it moves to the right or left of the fixation point (rows).

varies (indicated by each column in the figure). The most important thing to notice is that the estimated location is more accurate when the robot "looks directly" at the object (central column, $\phi_o = 0°$). In other words, when the robot *fixates* the object, and the object appears in the centre of its FoV (as explained at the beginning of this section).

4.4.1.2 Maintaining the Visual Memory

The observation model can now be used to estimate the location of landmarks once they are detected. These estimates are used in turn to maintain the visual memory (Section 4.2). The visual memory is updated depending on how the landmarks are classified whilst a fixation occurs:

- **New landmark:** When the robot detects a new landmark (i.e. it appears in the robot's view point), its pair $(e_i, pos(e_i))$ is added into the visual memory, and the particle filter $\mathcal{G}_i$ associated with $pos(e_i)$ is initialised by uniformly distributing the

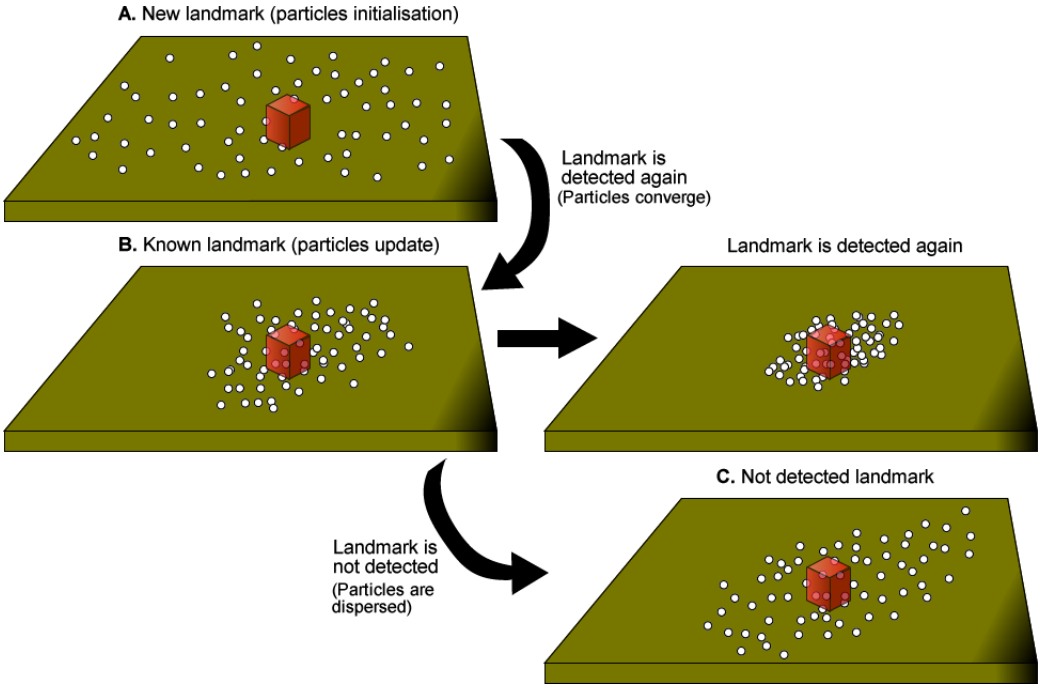

Figure 4.7: Maintaining the visual memory when: A. A new landmark is detected. B. A known landmark is detected again. C. A known landmark is not detected for some time. See text for details.

particles across the table (as illustrated in Figure 4.7A).

- **Known landmark:** Every time a landmark e_i is detected again, its location is estimated and with this new evidence its posterior distribution $pos(e_i)$ is updated. In our case, the corresponding particle set $\mathcal{G}_i$ gets updated (Figure 4.7B). How the particle filter gets updated is explained below.

- **Not detected landmark:** If a landmark does not appear in the robot's view point for 1.5 seconds, small Gaussian noise with zero mean and 1.0 cm of standard deviation is added to its current estimate (Figure 4.7C). This noise models temporal decline in visual working memory. This is particularly useful for the case when objects move across the table, which can happen when the robot accidentally hits and object with its hand or a grasped object.

By accessing the visual memory, the robot employs these location estimates in order to decide what manipulation action(s) to perform and how to control its gaze.

4.4.1.3 Particle Filter Update

When the robot detects a *known* landmark e_i (i.e. the landmark appears insider the robot's view point), then its posterior distribution $pos(e_i)$ is updated with the new *evidence* about its just estimated location. This new evidence is denoted by ω, which in belief estimation techniques (as those presented in Section 2.7) this evidence is referred to as *observation*. Throughout this chapter we will make use of the term *observation* to refer specifically to an estimated location *(x,y)* of a particular landmark, thus $\omega = (\omega_x, \omega_y)$. The term *observation* must not be confused with a *fixation*. As explained above, a fixation centres a landmark in the robot's FoV, the image obtained can actually contain more than one landmark, and an *observation* ω will result for each landmark.

The posterior distribution is updated following the *resampling* process of particle filters [64], which makes the particles converge as more *observations* are made (see Section 2.7.2). First the weight of each particle is calculated: $weight(g^j) = p(\omega|g^j)$, that represents the probability of the *observation* ω given the particle g^j, where j is used to index the particles. Recall that each particle g^j represents a possible *(x,y)* location for a particular landmark. This location is used, along with the current camera pose and the current fixation point, to index the observation model in order to obtain one of the bivariate Gaussian densities. The resulting density is centred at the location given by the particle g^j, then the probability density is calculated for the *observation* ω using the probability density function (pdf) of the selected Gaussian density. This probability density is the weight assigned to the particle g^j:

$$weight(g^j) = \frac{1}{\sigma_{om}\sqrt{2\pi}} e^{-\frac{(\omega-\mu_{om})^2}{2\sigma_{om}^2}} \tag{4.2}$$

Where μ_{om} and σ_{om} refer to the mean and standard deviation respectively, of the Gaussian density that has been indexed from the observation model *om*, and ω is the *observation*.

Figure 4.8 illustrates how the weights are obtained for different particles (white circles). Let us assume that we want to calculate the weight of the particle in the

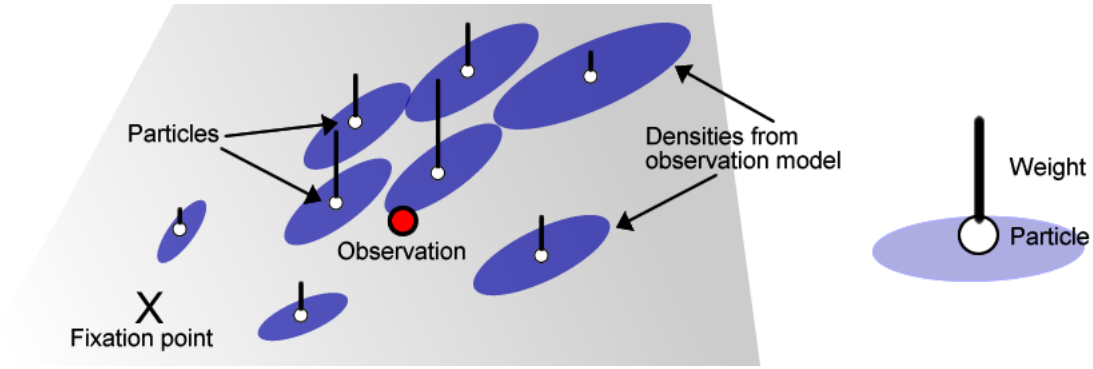

Figure 4.8: Representation of how the weights of different particles (white circles) are calculated according to the location of the fixation point (marked by X), the *observation* (red circle), and the corresponding Gaussian distribution obtained from the observation model. The weight of each particle (vertical line) is calculated using the probability density function in terms of the particle (see text for details).

upper right corner. First, based on the fixation point (marked with an X), the current camera pose (not shown) and the particle, the observation model is indexed to obtain the corresponding Gaussian density (shown here as a 2D ellipsoidal projection) centred at the particle. Then, the probability density is calculated in terms of the *observation* (red circle), and this value is assigned as weight to the particle (represented as a vertical line). Recall that the *observation* is where the robot has detected the landmark to be in the current fixation. Notice that densities closer to the fixation point are smaller because the particles are closer to the centre of the robot's field of view (FoV). Also, notice that those particles closer to the *observation* have a larger weight. If, for example, the *observation* was closer to the fixation point, it would mean that the landmark is centred in the FoV and most of the particles would have a large weight. On the other hand, if the observation is far from the fixation point then the particles would have a small weight.

Once all the weights for all particles are calculated, a new particle set is formed by drawing particles with replacement from the old set with probability proportional to their weight (i.e. particles with large weight are more likely to be chosen for the new set). However, it is possible that some particles are selected more than once. To avoid this some noise is added to each particle of the new set according to the corresponding bivariate Gaussian density indexed when its weight was calculated[8].

[8]Further details about this added noise, along with other alternatives, will be discussed in Section

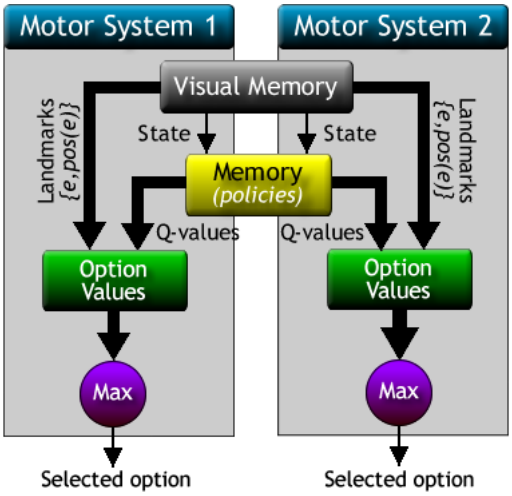

Figure 4.9: Schematic view of the manipulation action (option) selection mechanism.

As mentioned above, the manipulation tasks used in this thesis do not involve dynamic objects; they are expected to remain static. Nevertheless, sometimes the robot hits objects accidentally which causes the objects to move and change their location. To account for these cases, 10% of the particles are replaced with normally distributed random values [64].

4.4.2 Manipulation Action Selection

The robot selects options (i.e. manipulation actions) for each manipulation motor system ms by considering the location information of those landmarks within its reach. The list of manipulable objects is determined during run-time. As the manipulation motor systems work independently in the pick & place task, options are selected in parallel using the *Q-MDP algorithm* [65] (the selection process is illustrated in Figure 4.9). The idea is to select the manipulation action that provides the highest value based on the current location uncertainty of the available landmarks. This is achieved by the following equation:

$$o_{ms} = \arg\max_{o \in \mathcal{O}_{ms}} \frac{1}{J} \sum_{g^j \in \mathcal{G}_i} value(g^j, o) \tag{4.3}$$

Where g^j is a particle from the set $\mathcal{G}_i$ of the landmark e_i, and J is the number of

5.1, when the gaze control models are characterised.

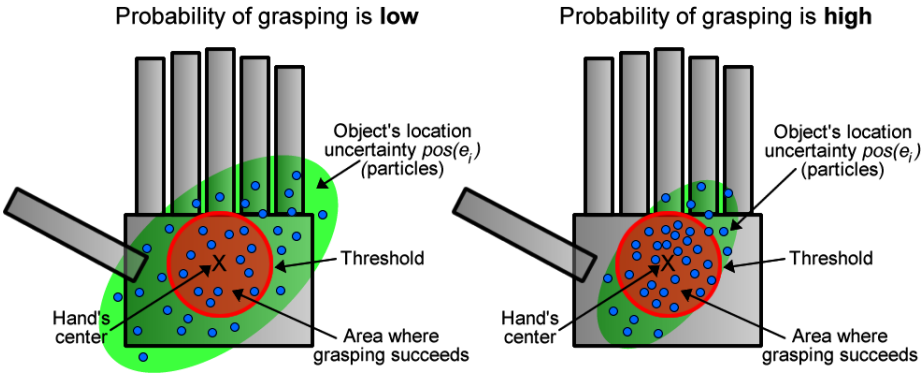

Figure 4.10: Representation of the probability of "success" or "failure" for option *Grasp* (see text for details).

particles. Then, $value(g^j, o)$ is the value of performing the manipulation action o on landmark e_i, assuming g^j is the landmark's location. In this work we define that an option o can either "succeed" or "fail". An option is said to "succeed" if the offset between the centre of the hand and the location given by particle g^j is less than or equal to some threshold (e.g. 1.0 cm). An option is said to "fail" if the offset is greater than the threshold. If an option "succeeds" then $value(g^j, o)$ takes the *Q-value* of $Q_{ms}(s, o)$ (from the learnt policy π_{ms}). If the option "fails" then it takes the *Q-value* of the second best option and is punished by adding -1 units of reward. These values are summed over all particles and then averaged by J.

This procedure maps the continuous information needed for low-level control to the discrete state space of each manipulation motor system (defined in Table tab:state-taski), by determining the likelihood of "success" and "failure" of options. Recall that the iCub arm controllers receive as input the 3D desired hand location, but the state space is discrete. For instance, Figure 4.10 illustrates the probability of "success" for option *Grasp* (the same principle applies for reaching). We define an area where grasping is said to succeed which is determined by a threshold value (red circle). According to Eq. 4.3, a particle lying inside this area will get the value of $Q_{ms}(s, o)$. On the other hand, a particle lying outside the threshold area will get the value of the second best option and will also be punished with -1 units of reward. At the end all these values are summed and averaged by the number of particles J. Therefore, the more particles lying inside the threshold, the higher the value for the option being evaluated (option

Grasp in this example). In other words, we say that the probability of "success"is high if the number of particles inside the threshold is big (right hand side figure). Notice that the number of particles lying inside the threshold area is determined by the spread of the distribution $pos(e_i)$ (green ellipsoid), which represents the location uncertainty of landmark e_i. In order to increase the probability of "success" of manipulation actions the location uncertainty needs to be minimised. This is achieved by fixating the landmark that is relevant to the task at a given moment. The *fixation selection process* is determined by the gaze control models which we now describe in turn.

4.5 Models of Gaze Control

The selection and the probability of "success" of manipulation actions depend on having the correct location information of each landmark. But only some information needs to be known to complete each step of the task, and the robot can choose which information to gather by fixating a specific landmark (i.e. by deciding *where to look*). Moreover, as explained in Section 1.3 we also deal with the problem of *gaze allocation*, because gaze control needs to be shared amongst our two manipulation motor systems (i.e. the robot's arms). Next, we describe our three one-step look ahead models of gaze control (introduced in Section 1.5) that aim to select fixation locations, where a fixation location corresponds to a specific landmark listed in the robot's visual memory (Section 4.2).

Because our models predict the benefit of looking at some landmark, which includes: i) the selection of that landmark, ii) the saccade to the selected landmark, and iii) the analysis of that fixation location; then we will use the term *perceptual actions*. Thus, the aim of our models of gaze control is to select a perceptual action $p_i \in \mathcal{P}$, associated to landmark e_i, that includes fixating that landmark (i.e. move the robot cameras so that the landmark appears in the centre of the robot's field of view), and calculating the value of the manipulation actions given the predicted location information of that landmark.

As explained in Section 1.5, predicting the benefit of the possible fixation locations

is based mainly on two parameters: positional uncertainty of landmarks and the value of performing manipulation actions. Depending on how the robot reasons about these two quantities at least three gaze control models emerge. First, the gaze control model based on uncertainty reduction (*Uncertainty* gaze scheme) is described, followed by the model of gaze control based on rewards and uncertainty (*Rew+Unc* gaze scheme), and finally the gaze control model based on rewards, uncertainty and gain (*Rew+Unc+Gain* gaze scheme).

4.5.1 Gaze Control based on Uncertainty

Our first gaze control model aims to maximise the reduction of positional uncertainty only. It predicts one step into the future the location uncertainty that would remain if each landmark e_i in visual memory, for each manipulation motor system, is fixated. This means that for each motor system ms the model first selects the perceptual action that will most reduce positional uncertainty, and this is denoted as p_{ms}:

$$p_{ms} = \arg\max_{p_i \in P} \left\{ \frac{1}{H} \sum_{\omega_h \in \Omega_i} dist(\mathcal{G}_i) - dist(\mathcal{G}_i^{p_i}, \omega_h) \right\} \tag{4.4}$$

The robot "imagines" that each perceptual action p_i is taken, i.e. each known landmark e_i is "fixated" (the imaginary fixation point for landmark e_i is the mean of the cloud of particles ($mean(\mathcal{G}_i)$), where $\mathcal{G}_i$ is the particle set representing the location uncertainty of that landmark. For each perceptual action p_i, imaginary *observations* ω_h (i.e. imaginary evidence) are sampled using the observation model (Section 4.4.1), where Ω_i is the set of *observations* produced by p_i, and H is the number of *observations*. As explained above an *observation* represents an estimated location *(x,y)* of a particular landmark. However in this case these observations are imaginary, since the robot is assuming to be fixating a landmark. Everytime that the observation model needs to be indexed we make use of the imaginary fixation point, the estimated camera pose and by randomly selecting a particle g^j from $\mathcal{G}_i$.

We first calculate the *current* spread in uncertainty using function $dist(\mathcal{G}_i)$. The

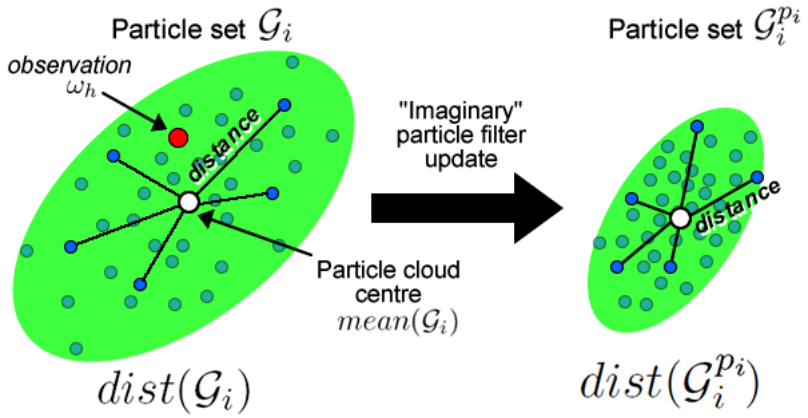

Figure 4.11: Measurement of the spread of the current particle set ($\mathcal{G}_i$), and the "imaginary" particle set ($\mathcal{G}_i^{p_i}$) using the Euclidean distance from each particle to the mean of the cloud.

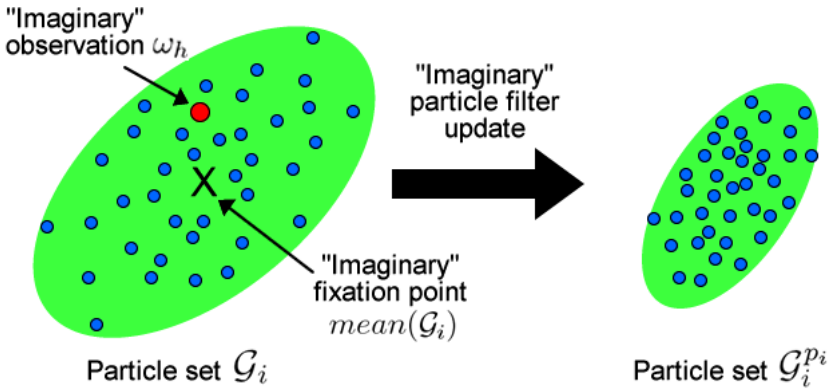

Figure 4.12: Representation of an "imaginary" particle filter update based on the "imaginary" observation ω_h.

spread is the average Euclidean distance between each particle $g^j \in \mathcal{G}_i$ and the mean of the cloud ($mean(\mathcal{G}_i)$). This is illustrated in Figure 4.11 (left hand side).

Then, each observation ω_h is used to perform an imaginary particle filter update (see Section 4.4.1), and the *predicted* spread in uncertainty is calculated by $dist(\mathcal{G}_i^{p_i}, \omega_h)$, assuming p_i is taken (Figure 4.11 right hand side). This update is depicted in Figure 4.12. The subtraction indicates the *residual* positional uncertainty, which is averaged over all *observations* for perceptual action p_i.

Once each manipulation motor system has selected its candidate perceptual action p_{ms}, gaze is *allocated* (or assigned) to the manipulation motor system that maximises the reduction in uncertainty, denoted as ms_f:

$$ms_f = \arg \max_{ms \in MS} \{p_{ms}\} \tag{4.5}$$

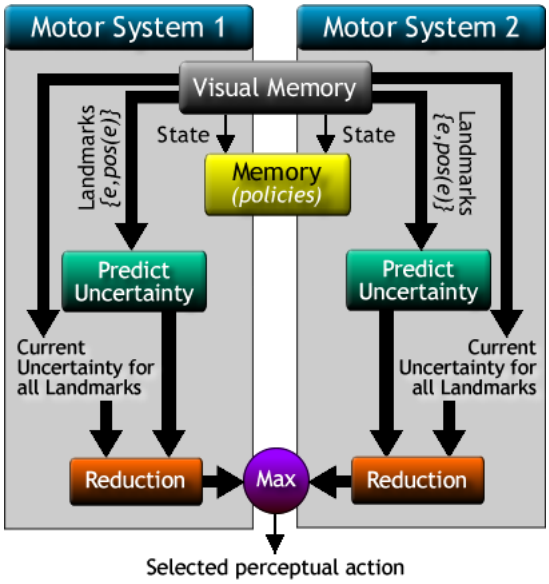

Figure 4.13: Schematic view of the gaze control model based on uncertainty reduction.

Finally, the selected manipulation motor system ms_f *executes* its previously selected perceptual action p_{ms}.

The fixation selection process for this model of gaze control is shown in Figure 4.13. This model favours perceptual actions (i.e. fixations) that obtain new information. Nevertheless, this new information does not imply that is relevant to the task. For instance, if the uncertainty about one of the containers is higher than the rest of the landmarks, the robot will choose to look at this container, even if the demands of the task have nothing to do with that container (see the example in Section 1.5).

4.5.2 Gaze Control based on Rewards and Uncertainty

This model integrates tasks rewards (i.e. the values of performing manipulation actions) into the decision process. The model predicts the value of performing manipulation actions (or options) that would be obtained *if* each landmark, for each manipulation motor system, is fixated the next time step. Thus, for each manipulation motor system ms, we calculate the *predicted* value of each option assuming perceptual action p_i is taken, and the maximum predicted value is selected (denoted as $V_{ms}^{p_i}$):

$$V_{ms}^{p_i} = \frac{1}{H} \sum_{\omega_h \in \Omega_i} \max_{o \in O_{ms}} \left(\frac{1}{J} \sum_{g^{j,\omega_h} \in \mathcal{G}_i^{\omega_h}} value(g^{j,\omega_h}, o) \right) \tag{4.6}$$

As the previous model, the robot first "imagines" that perceptual action p_i is actually taken by "fixating" on the mean of the cloud of particles ($mean(\mathcal{G}_i)$) of landmark e_i. For a perceptual action p_i, imaginary *observations* ω_h (i.e. imaginary landmark locations) are sampled using the observation model (as in the previous model), where Ω_i is the set of *observations* produced by perceptual action p_i, and H is the number of *observations*.

Basically, for each *observation* ω_h we verify which manipulation action o produces the maximum value the next time step. This is similar to Eq. 4.3 but here we need to update the particle filter with the imaginary *observation* ω_h. Thus, g^{j,ω_h} is a particle from the set $\mathcal{G}_i^{\omega_h}$, and J is the number of particles. $value(g^{j,\omega_h}, o)$ is the value of performing option o on landmark e_i, assuming g^{j,ω_h} is the predicted landmark's location. As explained in Section 4.4.2, if option o "succeeds" (i.e. if particle g^{j,ω_h} lies within the threshold area) then it takes the *Q-value* of $Q_{ms}(s, o)$. If the option "fails", it takes the *Q-value* of the second best action minus 1 units of reward. These values are summed over all particles and then averaged by J.

Then, for each manipulation motor system ms, we calculate the values $V_{ms}^{p_i}$ of each perceptual action associated to that motor system, and select the one with the maximum value (denoted as $M_{ms}^{p_i}$):

$$M_{ms}^{p_i} = \max_{p_i \in \mathcal{P}} \{V_{ms}^{p_i}\} \tag{4.7}$$

Once each motor system selects its value $M_{ms}^{p_i}$, gaze is *allocated* to the manipulation motor system with the highest value. Then this motor system *executes* its associated perceptual action, denoted as p_{ms}:

$$p_{ms} = \arg\max_{p_i \in \mathcal{P}} \left\{ \max_{ms \in MS} \{M_{ms}^{p_i}\} \right\} \tag{4.8}$$

Figure 4.14 shows a schematic view of the fixation selection process for this model. The main disadvantage of this model is its tendency to fixate landmarks with low lo-

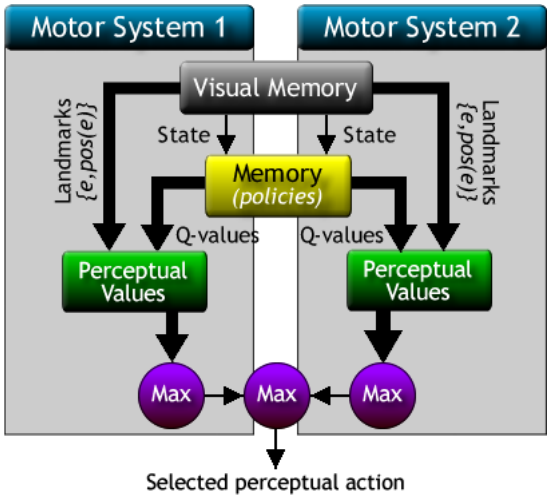

Figure 4.14: Schematic view of the gaze control model based on rewards and uncertainty.

cation uncertainty and high predicted value, which will not typically provide much new task information (as exemplified in Section 1.5).

4.5.3 Gaze Control based on Rewards, Uncertainty and Gain

Our third gaze control model is similar to the previous scheme, the main difference resides in how gaze is *allocated*. First, for each manipulation motor system ms, we calculate the values $V_{ms}^{p_i}$ of each perceptual action p_i, using Eq. 4.6. Second, Eq. 4.7 selects the value $M_{ms}^{p_i}$ for each motor system ms. However, instead of using $M_{ms}^{p_i}$ for gaze allocation directly as the previous model, here we calculate the *gain* that each motor system ms is expected to obtain if it is given access to perception:

$$gain_{ms}^{p_i} = M_{ms}^{p_i} - \max_{o \in O_{ms}} \{V_{ms}^{o}\} \tag{4.9}$$

$M_{ms}^{p_i}$ is the maximum value of performing a particular option, for manipulation motor system ms, if perceptual action p_i is taken (i.e. the *predicted* value one step ahead in time). V_{ms}^{o} is the value of executing option o in the *current* time step. The second term determines the current maximum value of performing some manipulation action, which essentially is calculated during the *manipulation action selection* using Eq. 4.3. Thus, this value can be cached. The subtraction determines how much the robot *gains* if gaze is allocated to the motor system ms. Therefore, gaze is *allocated* to the motor system

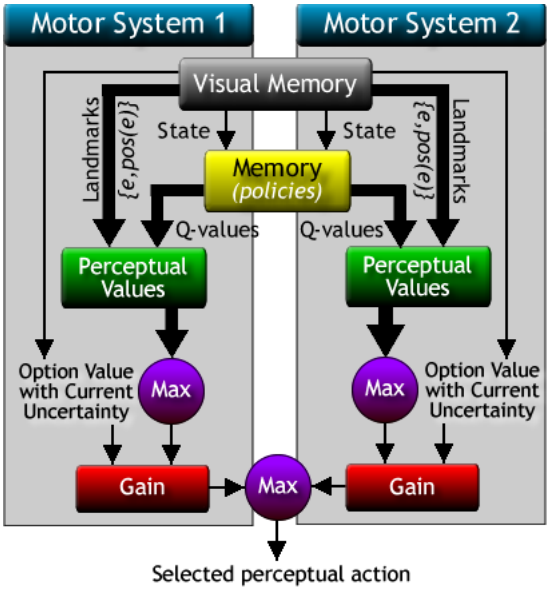

Figure 4.15: Schematic view of the gaze control model based on rewards, uncertainty and gain.

with the highest gain. Then this motor system *executes* its associated perceptual action, denoted as p_{ms}:

$$p_{ms} = \arg\max_{p_i \in \mathcal{P}} \left\{ \max_{ms \in MS} \{gain_{ms}^{p_i}\} \right\} \tag{4.10}$$

Figure 4.15 illustrates the fixation selection process of this strategy. This model tries to overcome the weakness of the preceding model by fixating landmarks that provide high predicted value and relevant task information.

4.6 Summary

This chapter described in detail the complete robot control architecture, its different control modules and their interaction. The control modules are: i) the *visual memory*, used to store the landmarks' ids and their corresponding location information; ii) the *visual analysis* module, which captures images from the right camera and calculates the 3D location of those landmarks inside the robot's field of view; iii) the *manipulation action selection* module, which is in charge of selecting options (i.e. manipulation actions) for each manipulation motor system; and iv) the *Fixation selection* or *perceptual action*

selection module, which decides where to look and how to allocate gaze.

The system has two temporal phases: the learning and the execution phase. During learning, the robot is trained to perform the task assuming all information about the task is available (i.e. the location of all landmarks). During execution, the robot should act under incomplete and uncertain information. It is here where the robot should control its gaze efficiently in order to accomplish the task (i.e. fixate landmarks in order to estimate their location).

The topic of this thesis lies within the *fixation selection* module. Three one-step look ahead models of gaze control were formulated and explained. These models reason about positional uncertainty of landmarks and the value of performing manipulation actions (i.e. rewards) in different ways. Therefore, the next chapter presents a series of experiments that characterise and compare these gaze control models in terms of task performance, in order to determine which one is more efficient.

Chapter 5

Analysis of the Gaze Control Models

The previous chapter described in detail our three candidate models of gaze control that can be used by an agent/robot in order to gather relevant task information for the selection of manipulation actions. This chapter presents a series of experiments with the aim of analysing the behaviour, the advantages and weaknesses, of these models of gaze control. To do so, the gaze control strategies are characterised using the pick & place task (introduced in Section 1.6)[1], in terms of task performance (by counting the number of objects that are put inside the containers), by varying three environmental variables [58]:

Reach/grasp sensitivity: Refers to the degree of positional accuracy in the manipulation actions. This sensitivity or accuracy is specified by a threshold value that directly affects the probability of "success" or "failure" of the manipulation actions as explained in Section 4.4.2[2].

[1]Recall that the pick & place task consists of picking up objects from the table top and then placing them inside one of two containers. A new object appears at a random position on the table: i) every 60 seconds, ii) every time an object is put inside a container, and iii) whenever an object falls from the table. The goal is to put as many objects as possible inside the containers.

[2]Recall that in this thesis, manipulation actions are said to "succeed" if the number of particles (representing the position of a landmark) lying inside the threshold area is big. On the other hand, manipulation actions are said to "fail" if the number of particles lying inside the threshold area is small. Please refer to Section 4.4.2 for a more detailed explanation.

Observation noise: Refers to the noise in the position estimates as specified by the observation model (Section 4.4.1.1). The observation model represents this noise as a function of the camera pose, the fixation point, and the landmark's location.

Field of View (FoV): The camera's field of view determines the amount of the scene captured in a single image that the robot acquires during a fixation.

Even though the aim is to compare our three gaze control models, two more gaze schemes are presented that serve as common baseline and provide a more general analysis [43, 57]:

Random gaze control: This scheme randomly chooses to fixate one of the landmarks currently listed in the robot's visual memory.

Round Robin gaze control: This strategy loops through the list of landmarks currently stored in the visual memory fixating each landmark one by one.

The following sections present the results and analysis for each of the environmental variables just described. For purposes of brevity, we will refer to the gaze control model based on uncertainty reduction (Section 4.5.1) as the *Uncertainty* gaze strategy; the gaze control model based on rewards and uncertainty (Section 4.5.2) will be referred to as the *Rew+Unc* gaze strategy; finally, the gaze control model based on rewards, uncertainty and gain (Section 4.5.3) will be referred to as the *Rew+Unc+Gain* gaze strategy. Before we present the experiments of each environmental variable, this chapter starts by analysing different ways in which noise is added to the particle filters (associated to each landmark location), after every update as mentioned in Section 4.4.1.3. In particular, we analyse the differences between uniform and non-uniform spatial acuity in the field of view.

5.1 Spatial Acuity

As explained in the previous chapter (Section 4.2), particle filters are used to keep track of the location of landmarks, where each particle represents a possible location *(x,y)*

on the table for the given landmark. Every time the robot performs a fixation, those landmarks appearing inside the robot's view point are detected and triangulated, in order to estimate their 3D location[3]. This means that the particle filter associated with each visible landmark is updated, and in Section 4.4.1.3 we described how this update occurs. Recall that the particles are updated by means of a *resampling* mechanism, where the weight of each particle is calculated based on: i) the current fixation location, ii) the new observation (i.e. the new evidence about where the landmark is), and iii) the corresponding bivariate Gaussian density given by the observation model. We explained that, *after* the update, in order to avoid having individual particles pointing to exactly the same *(x,y)* location, some noise is added to each particle using the same bivariate Gaussian density previously selected. Typically, some noise must be added every step into the particle filter to avoid it becoming trapped in local minima [64]. In our case, the kind of noise to be added is influenced by our observation model and the size of the field of view as a proportion of the workspace.

The observation model, defined in Section 4.4.1.1, characterises the location uncertainty that results from the triangulation process in the iCub simulation in a rather large workspace. However, the workspace in the pick & place task is relatively small and just a few Gaussian densities from the observation model are actually used.

Also, an important question is how to model the observational effect of non-uniform spatial acuity within the field of view. The robot simulation generates images with an essentially uniform spatial acuity across the field of view, which means that the robot does not need to centre its cameras to a landmark. Therefore, our models of gaze control will cease to be important since obtaining a full view of the workspace is enough to get all the task relevant information reliably. However, we know that in the human visual system spatial acuity falls off sharply as we move outwards from the fovea to the periphery of the retina. As explained in Section 3.1.1, it is due to this property that the human eye needs to move the fovea to specific parts of the scene in order to gather reliable information. As stated in Chapter 1, we do not follow a biological approach,

[3]Recall that Gaussian noise with zero mean and 1.0 cm of standard deviation is added to those landmarks that are not inside the current view point for 1.5 seconds.

nevertheless we should consider this modelling issue when selecting the noise function to be added each step to prevent the particle filter becoming trapped in local minima. In this section we test three noise models. The first is suitable for modelling systems with uniform spatial acuity (Model 1), the other two models (Models 2 and 3) are suitable for modelling systems with spatial acuity that falls off from the centre of the field of view.

- **Model 1 - Uniform spatial acuity:** After the particle filter update, the standard deviation of the cloud of particles is measured. Normally distributed noise is added to each particle with zero mean and standard deviation equal to the 10% of the cloud's standard deviation. Thus the particle filter adds noise that is modulated by the belief state, and noise can be the same across the visual field. This noise is used to escape the local minima.

- **Model 2 - Stepped spatial acuity:** The same noise as in Model 1 is added to the particle filter, to escape local minima. In addition, normally distributed noise (zero mean and 1.0 cm of standard deviation) is added to those landmarks inside the FoV but not in the centre of the camera (where we defined the centre to cover 10°). This models two regions: i) the "fovea" (0°to 10°) having higher spatial acuity, and ii) the "periphery" of the field of view ($> 10°$) having lower spatial acuity. However, note that the spatial acuity is *uniform* across the "periphery" of the field of view.

- **Model 3 - Smoothly varying spatial acuity:** Noise is added to each particle according to the bivariate Gaussian density that gives the observation model (Section 4.4.1.1) for that particle's position in the visual field. By re-employing the observation model in this way we are able to model a smooth fall off in spatial acuity as we move from the centre of the field of view to the periphery.

The modelling assumption here is quite important. As explained above we would expect gaze control to cease to be important if the robot can gather equally reliable information about all locations in the field of view. Gaze control methods will thus

be more differentiated as the spatial acuity of the periphery of the field of view falls off more rapidly. We measure this effect below as we vary the reach/grasp sensitivity environmental variable.

5.1.1 Results

The three noise models were tested based on the environmental variables defined at the beginning of this chapter. Three values were used to vary the reach/grasp sensitivity threshold: 0.5, 1.0 and 1.5 cm. The other two environmental variables are set to their default values. The observation noise is set to 1.0, meaning that an unmodified version of the observation model was used. The FoV is set to 60°x40°, where the angles refer to the complete horizontal and vertical fields (see Figure 5.6). Figure 5.1A-C shows the average number of objects correctly placed in the containers for Model 1, 2 and 3 respectively, whilst performing the pick & place task. A total of 10 trials of 5 minutes each were performed for the *Rew+Unc+Gain*, *Random* and *Round Robin* gaze schemes, for each set of results (the *Rew+Unc* and *Uncertainty* gaze strategies were not used for these experiments). The error bars in the graphs represent the 95% confidence intervals.

The results for Model 1 (Figure 5.1A) show the effect in task performance when the FoV has uniform spatial acuity. The *Rew+Unc+Gain* gaze strategy is much better than the two baselines when the threshold is small (0.5 cm), then it is slightly better for the default threshold value (1.0 cm), but it is overrun by *Round Robin* for the threshold value of 1.5 cm. The results for Model 2 (Figure 5.1B) show the effect in task performance when the FoV has a stepped spatial acuity. In this case the *Rew+Unc+Gain* gaze scheme is better for all three threshold values. The same happens for Model 3 (Figure 5.1C), that shows the effect in task performance when the FoV has a smoothly varying spatial acuity.

The two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain* gaze strategy against the two baselines for all noise models. Using Model 1, for the threshold values of 0.5 and 1.0 cm the differences are statistically significant at $p < .01$. However, the value of 1.5 is not statistically significant. Following Model 2 and 3, for

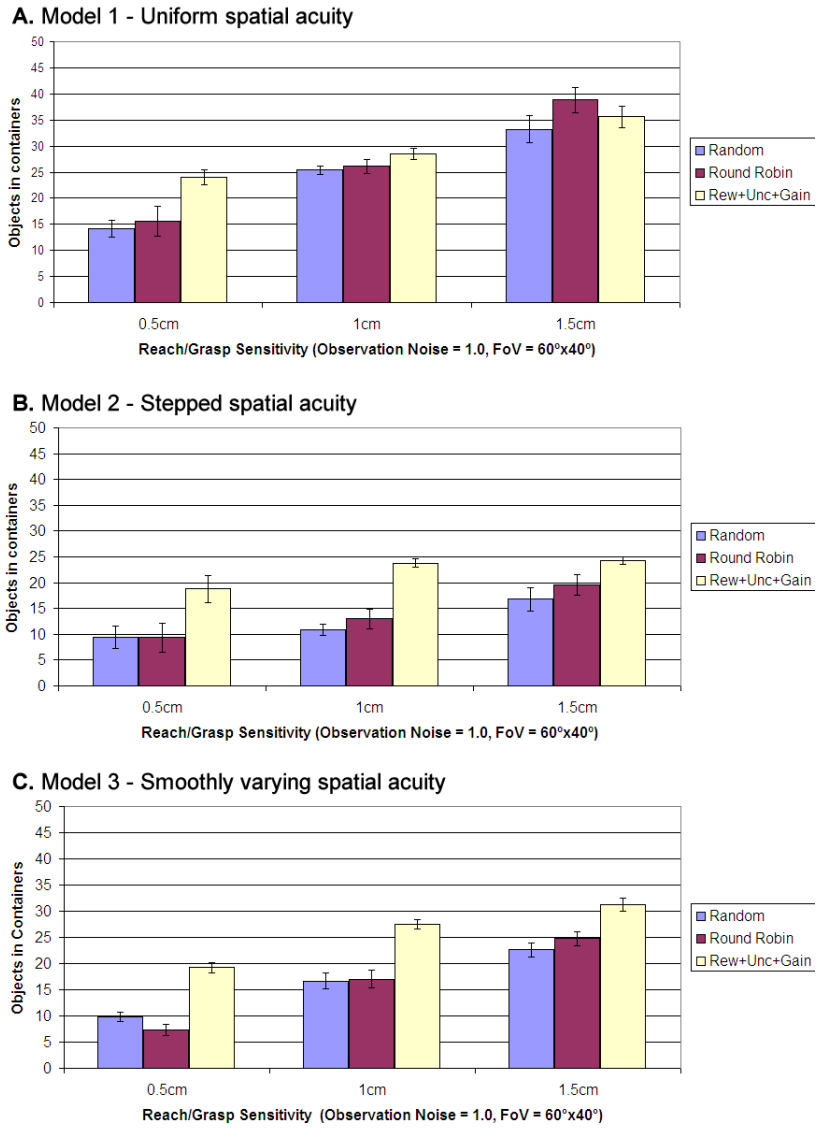

Figure 5.1: Effect in task performance when the field of view has: A) Uniform spatial acuity (Model 1), B) Stepped spatial acuity (Model 2), and C) Smoothly varying spatial acuity (Model 3); whilst the reach/grasp sensitivity varies. The observation noise = 1.0 and the field of view = 60°x40°. The error bars represent the 95% confidence intervals.

all threshold values the differences are statistically significant at p <.003.

5.1.2 Discussion

When the FoV has a uniform spatial acuity (Model 1) the robot can gather equally reliable information about all locations in the field of view. This is even more important since the default FoV used by the simulator (i.e. 60°x40°) is large enough to cover most of the robot's workspace. In this case, note that a rational gaze control strategy is more efficient if higher accuracy in the manipulation actions is required (e.g. when the

threshold is 0.5 cm). However, gaze control quickly ceases to be important, since for the threshold value of 1.0 cm the *Rew+Unc+Gain* gaze scheme is just barely better than the two baselines. For a threshold of 1.5 cm a simple gaze strategy (e.g. *Round Robin*) is enough to have a very good performance. This is because it really does not matter where the centre of the cameras are, since we get the same information across the FoV.

This effect is countered if the FoV has non-uniform spatial acuity. Model 2 has a stepped spatial acuity, where uniform noise is added to the location estimates of those landmarks *not* in the centre of the camera (i.e. $> 10°$). As it is now more important for the landmarks to appear in the centre of the cameras in order to obtain reliable information, we can immediately see that the two baselines do not provide a good gaze strategy. Similarly, Model 3 shows almost the same results as Model 2, where smoothly varying spatial acuity is modelled. Recall that instead of adding the same noise to a landmark appearing in the "periphery", Model 3 uses the same observation model to add noise which makes spatial acuity to fall off smoothly from the centre to the periphery.

From the graphs in Figure 5.1, we can first compare the effects between uniform and non-uniform spatial acuity. Notice that the average number of objects placed in the containers is higher in Model 1. This further demonstrates that by having a FoV with uniform spatial acuity, and in this case, one that covers a large portion of the workspace, the robot does not really need to move the cameras. Thus, the benefit of gaze control for the selection of fixation locations disappears. Secondly, we can compare the effects between the non-uniform models (Models 2 and 3). Notice that the difference in task performance between the *Rew+Unc+Gain* gaze scheme and the two baselines is practically the same for each threshold value in Models 2 and 3. What changes is the overall average number of objects. Note that it increases when Model 3 is employed. This is interesting because it shows that there is an overall change between stepped and smooth spatial acuity, but the proportions within each threshold values are similar.

Since the aim of this thesis is to formulate models of gaze control for the selection of fixation locations, we will follow a model of non-uniform spatial acuity in the FoV. Specifically, we will make use of Model 3 since it offers a better way to differentiate be-

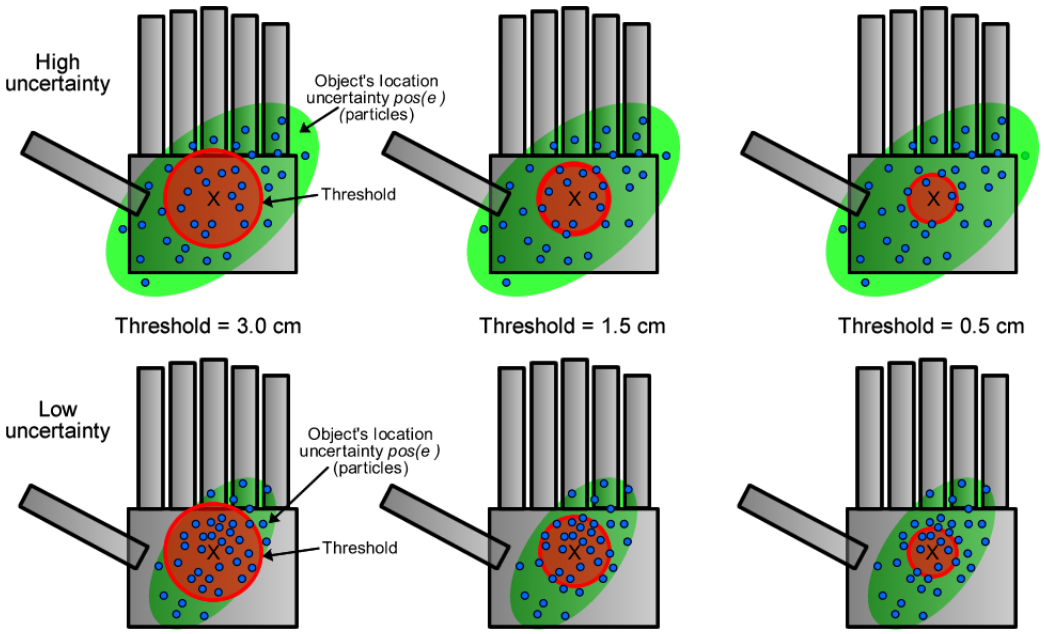

Figure 5.2: Representation of different scenarios for the reach/grasp sensitivity condition. The top row presents cases where the location uncertainty is high (represented by the large green ellipsoids), whereas the bottom row presents cases with low uncertainty (represented by the small green ellipsoids). The red circle determines the threshold values of 3.0, 1.5 and 0.5 cm. (See text for details)

tween our proposed models of gaze control. Furthermore, Model 3 is qualitatively similar although not the same as the human eye. Appendix A presents further experiments that compare Models 1 and 3.

5.2 Reach/Grasp Sensitivity

The first environmental variable to be analysed is the sensitivity in the manipulation actions. As explained in Section 4.4.2, the predicted "success" or "failure" of options is determined by how many particles lie inside the threshold area. This threshold defines the accuracy with which the robot must be able to reach for, grasp and release objects, in order for those actions to "succeed". Figure 5.2 depicts different scenarios when the location uncertainty of an object is high (represented by the large green ellipsoids in the top row), and when the location uncertainty is low (represented by the small green ellipsoids in the bottom row), for the threshold values of 3.0, 1.5 and 0.5 cm (represented by the red circle). A reaching or grasping action is said to have a high probability of "success" if the proportion of particles inside the threshold area is large.

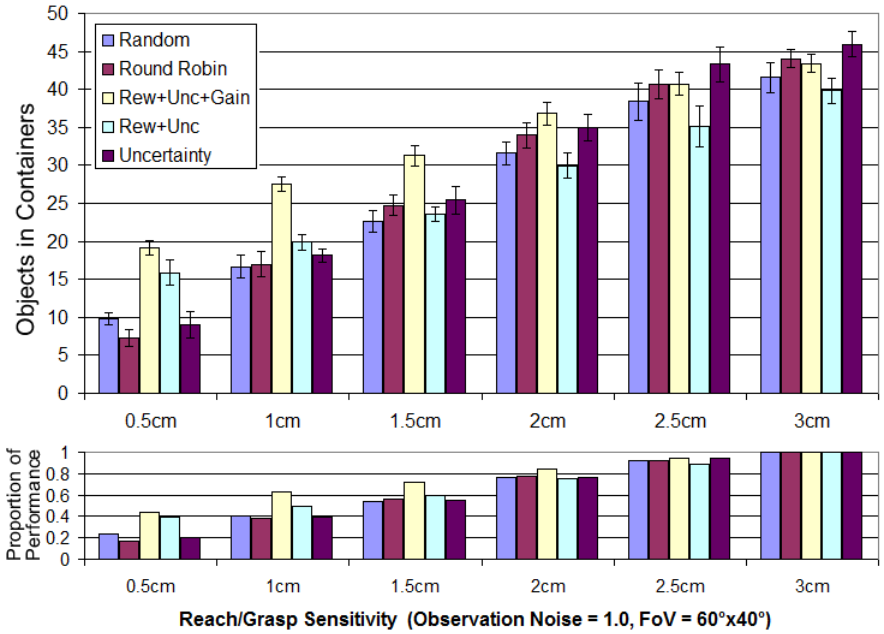

Figure 5.3: Results for reach/grasp sensitivity analysis, with observation noise = 1.0 and the field of view = 60°x40°. The top graph shows task performance. The lower graph shows the proportion of actual performance compared to the best case for each strategy. The error bars represent the 95% confidence intervals.

Notice that as the threshold area gets bigger (e.g. 3.0 cm), more particles will lie inside it, even if the uncertainty about a landmark's location is high[4]. On the other hand, as the threshold gets smaller (e.g. 0.5 cm), reducing the landmark's location uncertainty becomes critical. Based on these cases it is expected that, as the threshold value gets smaller sensitivity to positional uncertainty rises and task performance should decrease.

5.2.1 Results

For the experiment six threshold values were defined: 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0 cm. Figure 5.3 shows the average number of objects correctly placed in the containers for all five gaze strategies. A total of 15 trials of 5 minutes each were performed for each gaze scheme. The observation noise is set to 1.0, meaning that an unmodified version of the observation model was used, and the FoV is set to 60°x40°. This FoV is the default in the simulator and in our experiments, where the angles refer to the complete

[4]The release action works in the same way as reach and grasp. In this case, because the container is bigger than the objects, the accuracy may be in principle not as important as for reach and grasp. However, we force the robot to release the objects using the same threshold value as for the other actions. The release area is actually defined to be close to one of the bottom corners of the containers, not the container's centre.

horizontal and vertical fields (see Figure 5.6). The error bars in the graph represent the 95% confidence intervals.

As expected, the task performance of all gaze strategies drops as the sensitivity increases, i.e. when the threshold moves towards 0.5 cm. The *Rew+Unc+Gain* gaze strategy outperforms all other strategies, except when sensitivity is 2.5 and 3.0 cm, where the *Uncertainty* gaze strategy performs better. The *Rew+Unc* gaze strategy performs better than the *Uncertainty* strategy for the values 0.5 and 1.0 cm, but in general the *Uncertainty* strategy has the performance no better than *Random* and *Round Robin*. The lower graph of Figure 5.3 depicts the proportion of actual performance compared to the best case for each strategy. This determines the robustness of the gaze schemes to changes in the sensitivity of the manipulation actions. In this case, the *Rew+Unc+Gain* strategy is more robust to those changes.

First, the two-way ANOVA test was used where the threshold values and the gaze strategies were defined as factors for the test. For both factors the differences are statistically significant at $p < .0001$. Second, the two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain* gaze strategy against each strategy[5]. For the values of 0.5, 1.0 and 1.5 cm the differences are statistically significant at $p < .0001$. For the value of 2.0, the *Rew+Unc+Gain* strategy is not statistically significant against the *Uncertainty* scheme. Finally, for the values of 2.5 and 3.0 cm only against the *Rew+Unc* scheme the differences are statistically significant at $p < .002$.

5.2.2 Discussion

In terms of task performance it is clear that the *Rew+Unc+Gain* gaze strategy is the best option when more accuracy is required, i.e. for the threshold values of 0.5 and 1.0 cm. The rest of the strategies start to catch up as the threshold value increases. In particular, for the threshold values of 2.5 and 3.0 cm, even the performance of *Random* and *Round Robin* matches the performance of our models of gaze control. The main reason for this is that the default FoV covers a large part of the robot's workspace, thus

[5]For these tests we have chosen to focus on the *Rew+Unc+Gain* gaze strategy because, as it will be shown, it is in general the best scheme in terms of performance and robustness.

more landmarks appear inside the image taken in every fixation. Therefore, even if the robot decides to look (i.e. to centre its cameras) at a task irrelevant landmark, there might be other landmarks inside the FoV that are task relevant. When the threshold value is small (e.g. 0.5 cm), landmarks need to appear in the camera's centre in order to estimate their location reliably and in consequence the robot will be able to manipulate them successfully. As the threshold value increases, the threshold area gets bigger and more particles will lie inside it. This makes it more likely for manipulation actions to "succeed", even if the location uncertainty of a landmark is big.

The *Uncertainty* gaze strategy performs better as the threshold value increases. This suggests that selecting fixation locations based only on uncertainty reduction is enough when the requirements of the task are simplified. In the case of the *Rew+Unc* gaze strategy, recall that it might prefer to look at landmarks that do not provide new task information but produce high value, so it will tend to look at a landmark more than needed. This behaviour is good when the threshold values are small, since more accuracy is required. However, it is a bad behaviour when the threshold value increases, because it wastes time looking at landmarks from which no more relevant information can be extracted. In fact, when following the *Rew+Unc* strategy, the workload between arms fluctuates during the trial. The robot ends up working with only one arm for some periods of time. If the location uncertainty of the landmarks associated with one of the arms increases, then the expected value for those landmarks gets smaller. As the *Rew+Unc* strategy looks at landmarks with high value, it will sometimes favour only one arm.

5.3 Observation Noise

This environmental variable refers to the observation model, explained in Section 4.4.1, that is used to maintain the particle filters of each landmark and represents the uncertainty associated to the triangulation process. By scaling the observation noise the location uncertainty about landmarks will also increase, making it harder for the robot

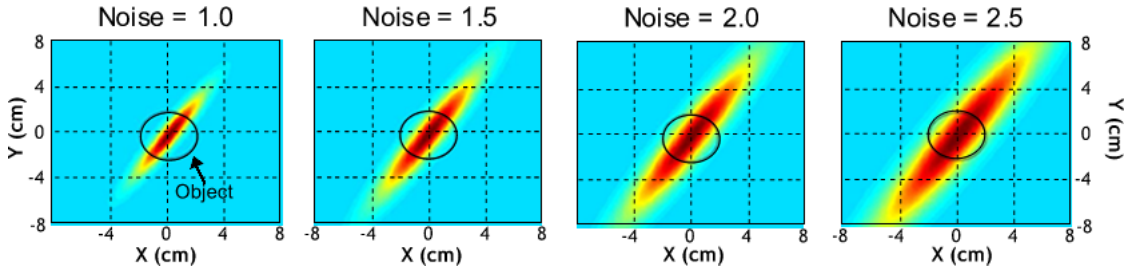

Figure 5.4: An example of how different levels of noise affect one of the densities from the observation model.

to successfully manipulate objects. Figure 5.4 shows what one of the densities from the observation model looks like when scaled with different levels of noise. A factor of 1.0 means an unchanged version of the observation model. As the observation noise increases it is expected that the task performance of all gaze schemes will fall.

5.3.1 Results

For these experiments, the observation model is scaled by the factors: 1.0, 1.5, 2.0 and 2.5. Figure 5.5 presents the average number of objects correctly placed in the containers for all gaze strategies, where reach/grasp sensitivity is set to 1.0 cm and the FoV is 60°x40°. A total of 15 trials of 5 minutes each were performed for each gaze strategy. The error bars in the graph represent the 95% confidence intervals.

As expected, the task performance of all gaze strategies decreases as observation noise increases. The *Rew+Unc+Gain* gaze strategy outperforms all other strategies in terms of task performance, whereas the second best is the *Rew+Unc* gaze scheme. The lower graph of Figure 5.5 shows the proportion of actual performance compared to the best case for each strategy. This determines the robustness of the gaze schemes to variations of the observation noise. In this case the *Rew+Unc* gaze strategy is the most robust in these experiments, closely followed by the *Rew+Unc+Gain* gaze strategy.

The two-way ANOVA test was used with the noise values and gaze strategies as factors. For both factors the differences are statistically significant at $p < .0001$. Then, the two-tailed unpaired t-test was used to compared the results of the *Rew+Unc+Gain* scheme against each strategy. All the differences are statistically significant at $p < .0001$.

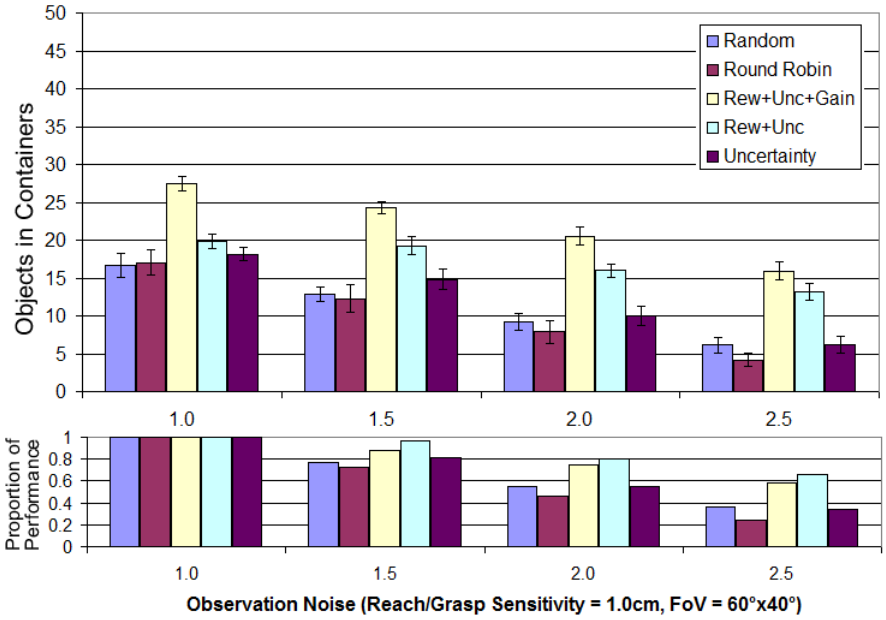

Figure 5.5: Results for the observation noise analysis, with reach/grasp sensitivity = 1.0 cm and the field of view = 60°x40°. The top graph shows task performance. The lower graph shows the proportion of actual performance compared to the best case for each strategy. The error bars represent the 95% confidence intervals.

5.3.2 Discussion

In terms of task performance, the *Rew+Unc+Gain* gaze strategy is the best choice. However, it is interesting to see that the *Rew+Unc* gaze strategy is the second best choice. As pointed out before, this strategy tends to fixate landmarks more often than needed. But this behaviour turns out to be good when the observation noise is large. This is the main reason why the *Rew+Unc* gaze strategy is more robust to observation noise than the rest of the strategies. In contrast, the *Uncertainty* gaze strategy has a similar task performance to *Random* and *Round Robin*. As the observation noise increases, landmarks need to be fixated several times for each manipulation action. However, the behaviour of the *Uncertainty* scheme is likely to select different fixation points every time step, which makes it difficult to fixate on a landmark continuously.

5.4 Field of View

The field of view (FoV) refers to the camera's angles that determine the size of the image captured during a fixation. A large FoV results in images with more visual information

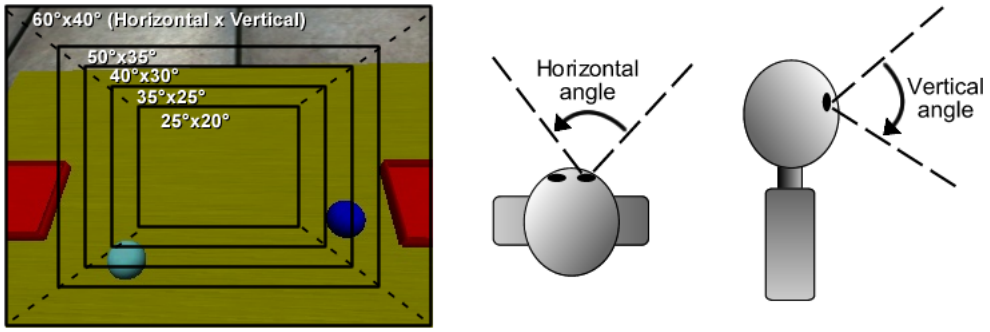

Figure 5.6: Changes in the FoV with respect to an image captured with the right camera. The angles in the FoV correspond to the complete horizontal and vertical planes.

(i.e. more landmarks may appear in the image). Thus, if the FoV narrows we would expect to see a reduction in task performance since there is less information in the view point. Figure 5.6 shows an image taken from the right camera and different sizes for the FoV. Notice how some objects and the containers no longer appear in the image as the FoV decreases. The angles that determine the FoV refer to the complete horizontal and vertical fields.

5.4.1 Results

For these experiments we vary the horizontal and vertical angles of the FoV with the values: (60°x40°), (50°x35°), (40°x30°), (35°x25°), and (25°x20°). Figure 5.7 presents the average number of objects correctly placed in the containers for all gaze strategies, where the reach/grasp sensitivity is set to 1.0 cm and the observation noise is set to 1.0. A total of 15 trials of 5 minutes each were performed for each gaze strategy. The error bars in the graph represent the 95% confidence intervals.

Again as expected, the task performance drops down for all gaze strategies as the FoV narrows. The *Rew+Unc+Gain* gaze strategy outperforms all other strategies, and the second best is the *Rew+Unc* gaze strategy. The lower graph of Figure 5.7 shows the proportion of actual performance compared to the best case for each strategy. This determines the robustness of the gaze schemes to changes in the FoV. In this case *Round Robin* and the *Rew+Unc* gaze strategy are more robust. Even though the *Rew+Unc+Gain* gaze strategy performs better than the rest of the strategies, it is not robust to changes

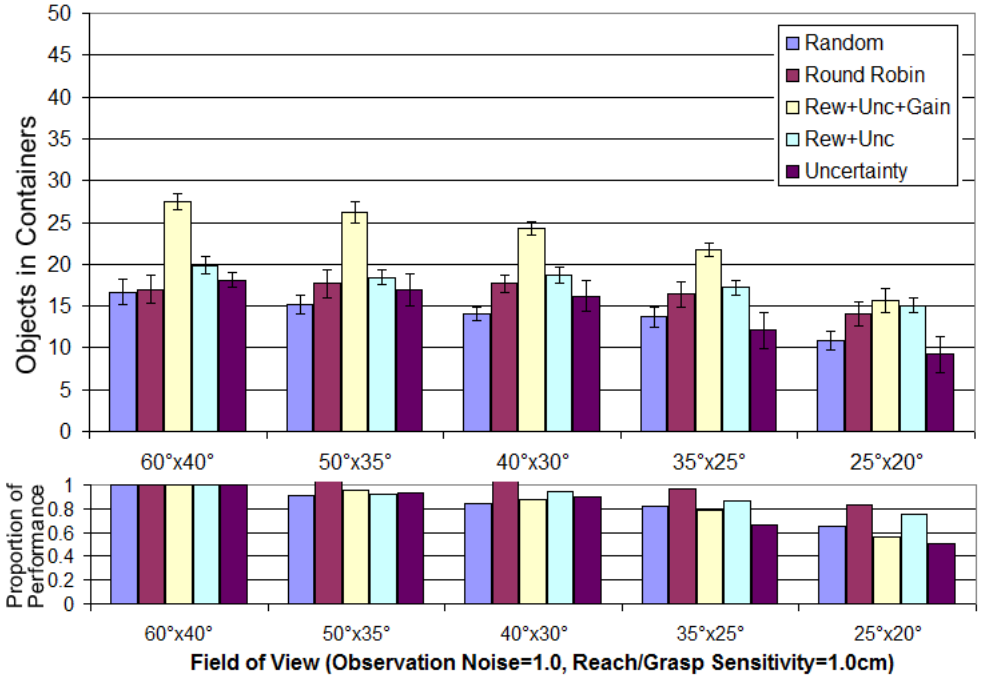

Figure 5.7: Results for the FoV analysis, with reach/grasp sensitivity = 1.0 cm and observation noise = 1.0. The top graph shows task performance. The lower graph shows the proportion of actual performance compared to the best case for each strategy. The error bars represent the 95% confidence intervals.

in the FoV.

The two-way ANOVA test was used with the FoV values and gaze strategies as factors. For both factors the differences are statistically significant at p <.0001. Then, the two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain* scheme against each strategy. All differences are statistically significant at p <.0001, except when the FoV is (25°x20°), where only against *Round Robin* and the *Rew+Unc* strategy the differences are not statistically significant.

5.4.2 Discussion

Once again the *Rew+Unc+Gain* gaze strategy is the best choice in terms of task performance. However, it is affected by changes in the FoV. This is because, as the FoV gets smaller the likelihood of finding new objects during a fixation gets smaller as well. Finding new objects inside the robot's current view point is crucial, as these objects are stored in the visual memory and are considered to be fixated in the next time step. Therefore, if the FoV is small then the robot should deliberately search for new objects.

In other words, *active visual search* [10] becomes an important issue. For this experiment a simple visual search heuristic was implemented, where every time the visual memory is empty the robot systematically searches for a new object across the table. It starts in the lower part of the right hand side of the table, and then follows a mow-the-lawn pattern[6].

It turns out that *Round Robin* and the *Rew+Unc* gaze strategies are more robust to changes in the FoV than the rest. *Round Robin* does not show a big variation in task performance because its predetermined behaviour makes gaze move across the table continuously, which in turn makes it possible to indirectly find new objects. The *Rew+Unc* gaze strategy is robust to changes in the FoV because, as explained above, sometimes it favours only one arm. Thus, even if the FoV narrows, one arm can still keep performing the task.

5.5 Conclusions

The experiments presented in this chapter helped us characterise our three candidate models of gaze control in terms of three environmental variables: reach/grasp sensitivity, observation noise and field of view (FoV). Two more gaze schemes, namely *Random* and *Round Robin*, were also considered in order to have a common baseline for our candidate models. The experiments analysed in this chapter provide interesting conclusions that are summarised here:

- In general, the *Uncertainty* gaze strategy performed in some cases no better than the *Random* or *Round Robin* gaze schemes. These results suggest that it is inefficient to control gaze reasoning only about the reduction in location uncertainty of landmarks.

- A more efficient method for the control of gaze comes when task rewards (i.e. the value of manipulation actions) are considered along with the location uncer-

[6]This visual search heuristic is never needed when the FoV is (60˚x40˚) or (50˚x35˚). This means that in these two cases the FoV is large enough to find new objects during the execution of the task

tainty of landmarks. As the *Rew+Unc* and the *Rew+Unc+Gain* gaze schemes demonstrate.

- The *Rew+Unc+Gain* gaze strategy is, in general, the best in terms of task performance for all environmental conditions. It dominates other methods in 13 of the 15 experimental conditions. It is also robust to variations in the sensitivity of the manipulation actions and observation noise. The only weakness is that it is affected by changes in the FoV.

- The *Rew+Unc* gaze strategy is the second best strategy in terms of performance and robustness. Recall that the difference between this scheme and the *Rew+Unc+Gain* gaze strategy is just the way in which *gaze allocation* is resolved. These results suggest that using the perceptual *gain* for gaze allocation is indeed a better choice (as implemented by the *Rew+Unc+Gain* scheme).

- Finally, the problem of visual search has been identified when the FoV narrows, in particular for the *Rew+Unc+Gain* gaze scheme. The proposed visual search heuristic explained above is not preemptive. It makes the robot stop performing the task until a new object is found, which is highly inefficient. Chapter 6 will present an integrated account of a more efficient and predictive visual search process that works together with the gaze controller, in order to increase the robustness of the *Rew+Unc+Gain* gaze scheme to changes in the FoV.

Chapter 6

Integration of an Active Visual Search Process

In the previous chapter, our proposed gaze control models were characterised in terms of three environmental conditions. The results showed that the *Rew+Unc+Gain* gaze strategy (described in Section 4.5.3) is, in general, the best scheme in terms of task performance for all conditions. However, we found that this gaze strategy was not particularly robust to changes in the field of view (FoV) (Section 5.4). As the FoV narrows there is less visual information appearing in the current view point and in consequence task performance is expected to drop. To counter this problem a simple visual search heuristic was implemented. Nevertheless, this heuristic was not sufficient to maintain a good task performance for the *Rew+Unc+Gain* gaze scheme. Therefore, one of the main conclusions of the previous chapter was that a more efficient visual search process had to be devised in order to increase the robustness of this particular model of gaze control in terms of changes in the FoV.

The main contribution of this chapter is to integrate a probabilistic *active visual search* process into the *Rew+Unc+Gain* gaze control model. The main question that we face here is: *How is a decision made when to perform visual search whilst performing a task?* We will see that most of the previous work has dealt with the visual search problem alone. However, in our case, the robot is already engaged in a manipulation

task, and it has to decide when is the best time to perform a visual search action. Thus we want to show how the *Rew+Unc+Gain* gaze scheme can naturally account for visual search as it is commonly performed in natural tasks.

This chapter is organised as follows. First, previous work for the problem of visual search is presented, along with a discussion of some visual search models and their similarities to the one proposed here. Second, a quick summary of the visual search heuristic, from Chapter 5, is provided for comparison. Third, the active visual search process is described and integrated into the *Rew+Unc+Gain* gaze strategy. Fourth, the same FoV experiment used in Section 5.4 is employed in order to analyse and compare the integrated model against the *Rew+Unc+Gain* scheme with the heuristic (along with the other gaze strategies). Concluding remarks are given at the end of the chapter.

6.1 Related Work

In general, visual search can be stated as the problem of finding a landmark in a visual scene [10]. A visual search behaviour takes place, for example, when we try to look for a book in our desk, pedestrians in a road intersection, a spoon whilst we are cooking, a friend in the street, etc. This section is organised in a similar way as Chapter 3. First, a summary of human visual search findings from the psychophysical literature is presented, and serve as motivation for our work. Second, a discussion of some computational models of visual search is provided. In particular, we focus on models of visual search implemented in robots.

6.1.1 Human Visual Search

Visual search tasks have been extensively employed in psychophysical human experiments with the goal of investigating visual preattentive and attentional processes [37, 217]. These studies have led to the formulation of classic theories of visual search, such as the feature integration theory [218], and the guided search theory [219]. Typical psychological experiments consist of a display where subjects had to find a particular

object from a discrete set of spatially separated distractors. Two types of search processes have been identified: i) parallel, and ii) serial search processes. *Parallel search* refers to the case where subjects are able to find the target object in an almost instantaneous decision time, no matter how many distractors are on the display. This type of search is used to investigate the preattentive process in humans, and is closely related to the concept of visual saliency (see Section 3.2); where things seem to "pop-out" from the visual scene. On the other hand, *serial search* refers to the case where subjects need to focus on different items in order to correctly discriminate between a distractor and the target. These theories, however, and most of the visual search literature disregard saccadic eye movements [10].

In many situations, particularly in real-world scenarios, visual search cannot be accomplished without performing saccades. Recent studies have considered eye movements as an integral part of the visual search task [95, 220–222]. Most of the experiments are still similar to the ones described in the paragraph above, where subjects search for a target item in a display, but now subjects are free to move their eyes. It has been found that there are two rules or factors that determine how gaze is deployed during a visual search task. The first factor refers to the *similarity* between the distractors and the target. Before the target is found, saccades are typically performed to items that look similar to the target object. The second factor refers to the *proximity* rule, which states that it is likely that the next saccade will be performed to items that are close to the current fixation location [10, 37]. Other experiments using photographs of real-world (mostly) indoor scenes provide evidence that saccades during visual search are guided by top-down processes, as opposed to salient image features [223–225]. This is in accordance with the findings presented in Section 3.1.2 that state that gaze is, in general, driven by the demands of the task [5, 16, 18]. It seems that the main top-down information that drives gaze is that of the object being searched for. Another type of top-down information that has been found to influence visual search is the context provided by the scene [226, 227]. Depending on the scene, for example, if it is a kitchen, a library, an outdoor scene, etc. the resulting scanpaths are different, suggesting that

humans are guided by prior knowledge about the spatial arrangement of the scene.

Of great interest to this thesis is the work by Navalpakkam et al. [228], where they investigated the role of reward during visual search. Subjects were given a monetary reward per trial depending on their ability to find a target amongst a set of distractors in synthetic images. Subjects were able to learn fast how to maximise the expected reward per trial. Another important work, already mentioned in Section 3.1.2, is that by Najemnik and Geisler [11, 123]. They found evidence that humans follow an optimal fixation selection strategy during a visual search task, compared to that of an ideal Bayesian observer; with the aim of maximising the information gain about target location on each saccade.

It is important to note that all the experiments just reviewed are based on images or photographs presented on a display, not in real-world settings. Also, subjects had only to perform the visual search task, without pursuing any other goal or sub-task. In Section 3.1.2 we reviewed several studies where subjects performed natural tasks. One such task was the tea-making task explored by Land et al. [109]. This task is interesting because subjects were able to freely move around a kitchen, where the utensils and the ingredients were located. This means that subjects could make decisions about changing their current view point during the task, something impossible to achieve whilst looking at a display. It was reported that during the tea-making task, subjects performed visual search but other actions were interrupted. However, this is also due to the nature of the task; tea-making follows a consistent sequence of actions. This means that the task cannot continue if, for example, the kettle is not found first. Changing the view point may need not only a saccade, but in some cases we may also need to move our head or body. Moving the view point for visual search, particularly whilst performing another task, has not been fully investigated in the psychophysical literature. However, this situation has been explored in robotics and we discuss it next.

6.1.2 Computational Visual Search

There is a large number of computational models of visual search that try to deal with the same problems discussed above, where only single images (typically the ones also shown to the human subjects) are presented to the model [154, 222]. Some of these models are based on saliency detection [229], which are good for parallel search. But most of the models are based on a combination between top-down and bottom-up information (similar to those discussed in Section 3.2). The most common top-down source is the information about the target; this could be the knowledge of target features, or a target template [47, 230]. Recently, context information about the scene has been successfully employed as another source of top-down information [231].

An important extension to the previous models is the use of robotic pan-tilt-zoom camera(s) that can be employed to explore a scene, instead of being restricted to a single image or view point. Vogel and Murphy [46] modelled the visual search problem using a POMDP (partially observable Markov decision process) (see Section 2.3), and proposed two indirect object search strategies: i) search for the target object by first locating places in the scene where it is likely to be found (e.g. desktops, walls), and ii) search first for a spatially adjacent object to the target. This model is not actually implemented in a robotic camera, rather it is simulated using pre-acquired images of different scenes. Nevertheless, the model involves pan and zoom actions, and the POMDP models noisy observations (i.e. evidence) about the location of the target. This model was later extended [187], in order to incorporate bottom-up saliency and scene context information. Whilst Vogel's model planned the following action to be executed, Minut and Mahadevan [188] learnt gaze patterns during a visual search task by means of reinforcement learning (RL) (see Section 2.5). A pan-tilt camera was used, and the task was set in an office environment. The next fixation location was decided based on the most salient point from the current view point. Positive reward was given if the target object was located closer to the centre of the camera. Their results show that the number of saccades needed to find the target object decreases as the training increases. The main drawback of this model is that it has to re-learn the gaze pattern if the target object

is moved to a different location. Nevertheless, it can be used to learn the location of semi-static objects, such as desks, cabinets, bookshelves, etc. and then we could follow a search strategy similar to the one proposed by Vogel and Murphy [46].

A further extension to the previous models is to allow mobile robots to engage in visual search tasks (also known as large-scale visual search). In this case, the dimensionality of the space of view points increases since the robot is now allowed to navigate through the environment [232]. At the same time, object detection and recognition may be more robust, as the target object(s) can be observed from multiple points of view. Typically, whilst performing the searching task, the robot should also take into consideration its energy consumption, thus effective search strategies should be devised. Tsotsos and Shubina [233, 234] provide a good overview of the visual search problem in mobile robots. Their visual search model considers the robot's prior knowledge about the target and the environment, and the performance of the recognition algorithms. They also decompose the search problem into the "where to look next", and "where to move next" processes. For the experiments, they tested their model with different initial probabilities. The selection of the next view point is done following a probabilistic approach, by maximising the probability of finding the target whilst minimising the searching time or a cost function. Following a similar optimisation approach, Saidi et al [235] implemented a visual search process in a humanoid robot. The robot selects the next best view point by trading-off three parameters: its energy consumption, the probability of detecting the target object, and the expected new information of the new view point. Aydemir et al. [236] constrain the search space by considering spatial relations amongst objects in order to perform indirect search. Furthermore, the robot is able to select a specific search strategy by means of a Markov decision process (MDP) (Section 2.2). An extension of this work is that from Hanheide et al. [237], where common-sense knowledge is combined with spatial-relational knowledge to form a probabilistic conceptual map. Also, they solve the navigation and exploration problem by means of a continual planner that switches between classic and probabilistic planning domains for different levels of abstraction.

Other visual search tasks have the aim of recognising several objects in a 3D environment. These objects can be recognised more reliably by moving to different viewpoints in the scene, as in [238,239]. Recently, Zhang and Sridharan [240] propose a hierarchical planning framework based on partially observable Markov decision processes (POMDPs) (Section 2.3). The robot first searches for previously learnt target objects in a 3D environment by means of occupancy grids, where each cell in the grid has associated a probability of finding a target object. Once the robot travels to a particular cell, it processes the scene using image feature matching or using another two-layer POMDP planner that allows the robot to plan visual operations on the scene. Finally, the visual search task may be part of a set of behaviours that the robot is executing. For instance, robots that play football need to search for the ball, players, obstacles, etc. whilst performing some other task(s). Kohlbrecher et al. [195] employ an entropy grid map that combines different sources of information for a humanoid robot. The different parameters used in the estimation of this map determine the prioritisation of the cells that should be considered as next view points.

Our proposed visual search process cannot be directly compared to the visual search processes just described, however the following is a list of similarities and differences between the processes just described and our proposed visual search process:

1. Our visual search process follows a probabilistic approach as does the majority of the works just described.

2. Our work space (i.e. the tabletop) is small compared to those robots situated in a full 3D environment (e.g. an office room).

3. Our robot's oculomotor system can move to obtain different view points, similar to those works that consider only a robotic camera.

4. Our robot does not move in the 3D environment as its whole body remains in the same position.

5. Our robot should perform visual search whilst engaged in a manipulation task, whereas most of the previous work has focused on just the visual search task.

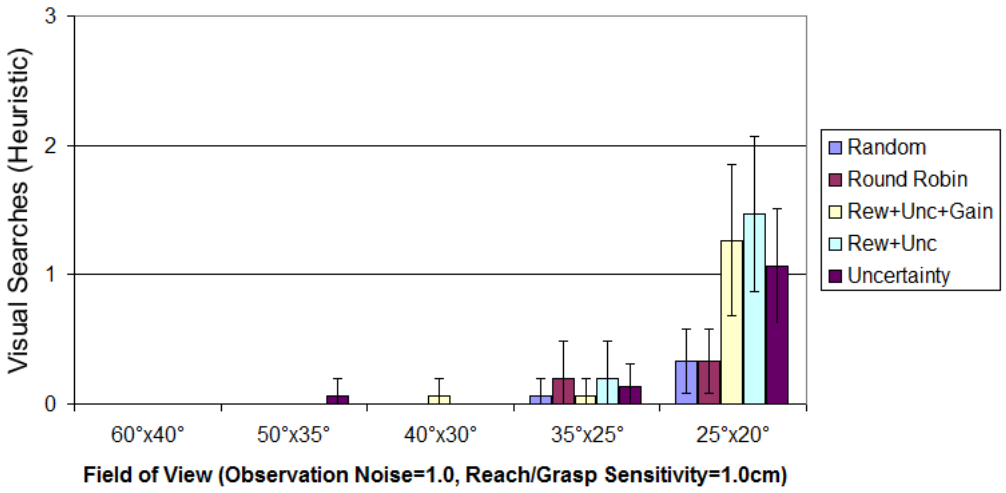

Figure 6.1: Average number of times that the visual search heuristic was employed during the field of view analysis presented in Section 5.4. The error bars represent the 95 % confidence intervals.

6.2 A Simple Visual Search Heuristic

Before we explain our proposed visual search process, let us describe in more detail the simple heuristic employed for the experiments performed in Chapter 5. This visual search heuristic is employed whenever the visual memory is empty. The robot systematically searches for a new object across the table, starting in the lower part of the right hand side of the table, and then following a mow-the-lawn pattern. Once at least one object is found, the searching heuristic stops and the normal execution of the task resumes. Because this heuristic always starts from the table's lower right, objects from the left hand side of the table may not be found for a long time. This will make the left arm remain idle. Of course, the heuristic could be modified to make it more efficient. For instance, the robot could randomly start its search on the left or right side of the table, or be forced to search for at least one object from the right and one from the left. Still, these heuristics will interrupt the normal execution of the manipulation task.

Figure 6.1 shows the average number of times that the visual search heuristic was employed, for all gaze strategies, during the FoV analysis presented in Section 5.4. This graph can also represent the average number of times that the visual memory was empty as the FoV narrowed. As mentioned in Section 5.4, and as shown in the graph, the FoV is typically large enough for finding new objects during the normal execution of the task.

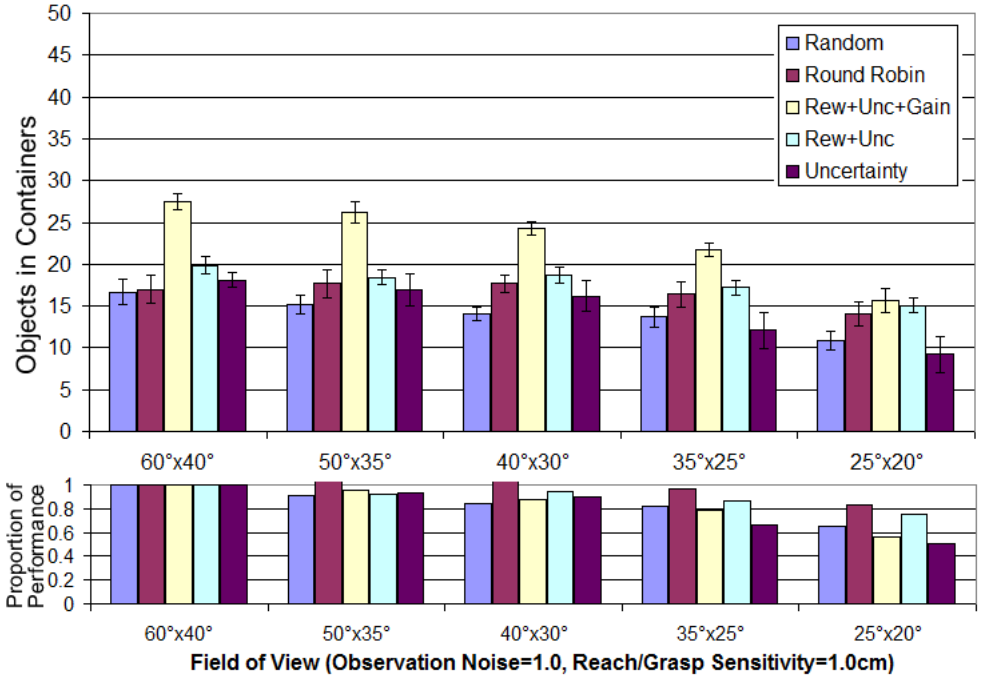

Figure 6.2: Results for FoV analysis, with reach/grasp sensitivity = 1.0 cm and observation noise = 1.0. The top graph shows the task performance and the lower graph the robustness for all strategies. The error bars represent the 95% confidence intervals. (This figure has been reproduced from Section 5.4)

The problem is when the FoV is set to (35°x25°), and (25°x20°). However, note that the number of visual searches is not so big. For instance, there are only 1.26 visual searches in average performed by the *Rew+Unc+Gain* gaze strategy, when the FoV is (25°x20°). Nevertheless, the results presented in Section 5.4 (reproduced here in Figure 6.2), clearly demonstrate that the task performance of the *Rew+Unc+Gain* gaze strategy decreases as the FoV narrows, and finding new objects is key to maintaining good performance.

6.3 Integrating Active Visual Search

The main drawback of the visual search heuristic is that it interrupts the execution of the manipulation task in order to find new objects. To avoid such interruptions the visual search process should be integrated into our existing *Rew+Unc+Gain* gaze control model. This means that visual search can take place along with the selection of fixation locations for the ongoing task. This brings up two main issues: i) how to decide when to perform a visual search action, and ii) how to perform the visual search process.

Here, a visual search action is one that aims to find new object(s) on the tabletop.

The *Rew+Unc+Gain* gaze control model makes decisions about what perceptual action to execute based on the information about those landmarks stored in the visual memory (see Section 4.5). However, the visual search process is not directly associated to a particular landmark, nor is part of the manipulation task. Therefore, in order to integrate the visual search process, the robot should also consider the execution of a visual search action as part of the available perceptual actions. To do so, a visual search action should be assigned with a *value* so that the robot can compare it against the values of the perceptual actions. In fact, this should be done for each manipulation motor system ms, particularly if the motor systems are independent such as in the pick & place task (Section 1.6).

Recall that the *Rew+Unc+Gain* gaze strategy decides to *allocate* gaze to the manipulation motor system with the maximum gain ($gain_{ms}$) when performing perceptual action p_i, as formulated in Section 4.5.3 by Eq. 4.9 (reproduced here for convenience as Eq. 6.1).

$$gain_{ms}^{p_i} = M_{ms}^{p_i} - \max_{a \in A_{ms}} \{V_{ms}^{a}\} \tag{6.1}$$

Where $M_{ms}^{p_i}$ is the maximum value of performing a particular option, for manipulation motor system ms, if perceptual action p_i is taken, and V_{ms}^{o} is the value of executing option o in the *current* time step. The second term determines the current maximum value of performing some option, thus the subtraction determines how much it is *gained* if gaze is allocated to the motor system ms.

We can extend this idea and define the *gain* of performing a visual search action for motor system ms, denoted as $gain_{ms}^{p_{vs}}$. To calculate this value we follow the equation:

$$gain_{ms}^{p_{vs}} = P(obj_{new}|vm_{ms})P(obj_{seeing}|p_{vs}, obj_{new}) \left[\max_{o \in O_{ms}} Q_{ms}(s', o) - \max_{o \in O_{ms}} Q_{ms}(s, o) \right] \tag{6.2}$$

Where $P(obj_{new}|vm_{ms})$ is the probability that there exists a new object on the table

given the number of objects in visual memory reachable by manipulation motor system ms. $P(obj_{seeing}|p_{vs}, obj_{new})$ is the probability of seeing an object given that perceptual action p_{vs} is executed and there is a new object on the table. The difference between the two maximum values is the gain that would be obtained if the visual search action is performed, where the discrete state s' represents the state that results after visual search. Note that this equation considers the discrete states instead of the objects' location (see Section 4.1). This is because the aim of visual search is to find objects and not to manipulate them. Finding a new object only changes the discrete states of the robot. The visual search gain ($gain_{ms}^{p_{vs}}$) is compared along $gain_{ms}^{p_i}$ to determine which motor system has the highest gain. As explained in Section 4.5.3, gaze is allocated to the motor system with the highest gain, then this motor system *executes* its associated perceptual action, denoted as p_{ms}:

$$p_{ms} = \arg \max_{p_{vs}, p_i \in \mathcal{P}} \left\{ \max_{ms \in MS} \{gain_{ms}^{p_i}, gain_{ms}^{p_{vs}}\} \right\} \tag{6.3}$$

Note that this equation is similar to Eq. 4.10 from Section 4.5.3, but here we also compare with $gain_{ms}^{p_{vs}}$. If the highest gain results from $gain_{ms}^{p_{vs}}$ then the visual search action p_{vs} is executed. Next, it is explained in detail how the probabilities are calculated.

6.3.1 Probability of a New Object

The main purpose of visual search is to find new objects to keep the robot busy at all times. In fact, the robot can only manipulate one object at a time with each arm, thus to keep the robot busy, the visual memory should contain information about at least two objects, one for each arm.

The first thing that the robot should take into account is the probability that there exists a new object on the table given the number of objects in the visual memory for each motor system ($P(obj_{new}|vm_{ms})$ in Eq. 6.2). Thus, this probability represents the objects' behaviour. Recall that in the pick & place task (defined in Section 1.6), a new object randomly appears on the table whenever the robot: i) puts an object in

FoV = 60° x 40°

vm_{ms}		obj_{new} = *true*	obj_{new} = *false*
	empty	1	0
	nonEmpty	0.45	0.55

FoV = 50° x 35°

vm_{ms}		obj_{new} = *true*	obj_{new} = *false*
	empty	1	0
	nonEmpty	0.36	0.64

FoV = 40° x 30°

vm_{ms}		obj_{new} = *true*	obj_{new} = *false*
	empty	1	0
	nonEmpty	0.52	0.48

FoV = 35° x 25°

vm_{ms}		obj_{new} = *true*	obj_{new} = *false*
	empty	1	0
	nonEmpty	0.53	0.47

FoV = 25° x 20°

vm_{ms}		obj_{new} = *true*	obj_{new} = *false*
	empty	1	0
	nonEmpty	0.72	0.28

Figure 6.3: The values for $P(obj_{new}|vm_{ms})$ for all field of view values.

the container, ii) every 60 seconds, and iii) if an object falls from the table. Because of this, it can be guaranteed that there will always be at least one object on the table reachable by each arm. Therefore, the probability that there exists a new object should be 1.0 if the visual memory is empty. Thus, the number of objects in the visual memory that need to be considered is: 0 objects, or 1 or more objects. In other words, $vm_{ms} = \{empty, nonEmpty\}$. The values for the variable corresponding to the new object are $obj_{new} = \{true, false\}$.

These conditional probabilities were learnt off-line, using the iCub simulator, for each field of view (FoV) value: (60°x40°), (50°x35°), (40°x30°), (35°x25°), and (25°x20°). The reach/grasp sensitivity was set to 1.0 cm and the observation noise was set to 1.0. A total of 5 training trials of 5 minutes each were performed. Every 10 seconds during each trial, we recorded the number of objects in visual memory and the number of objects on the table for each arm. As explained before, every time that the visual memory was empty there was at least one new object on the table, i.e. $P(obj_{new} = true|vm_{ms} = empty) = 1.0$, for all values of the FoV. In order to obtain $P(obj_{new} = true|vm_{ms} = nonEmpty)$ we checked the number of times that a new object was on the table by subtracting the number of objects on the table by those in visual memory, and then dividing this value by the number of times that the visual memory was not empty. Figure 6.3 presents the values of all the conditional probabilities for each FoV value.

6.3.2 Probability of Seeing a New Object

The previous probability informs the robot the likelihood that a new object has appeared on the table given the number of objects in visual memory. The problem is that objects appear on random locations across the table, so knowing that there is a new object on the table does not mean that the robot will see it. Thus, the robot also needs to take into account the probability of actually finding such new object. More specifically, the robot should know the probability of seeing a new object given that perceptual action p_{vs} is executed and that there exists a new object ($P(obj_{seeing}|p_{vs}, obj_{new})$ in Eq. 6.2). The values that obj_{seeing} can take are *true* and *false*.

Of course, if $obj_{new} = false$ then the probability of seeing an object is zero. So we just need to focus on the case when there is a new object on the table, i.e. $obj_{new} = true$. Finding the new object depends on where the robot would look on the table, which is represented by the perceptual action p_{vs}.

Here, the robot also learns off-line the conditional probabilities for all FoV values: (60°x40°), (50°x35°), (40°x30°), (35°x25°), and (25°x20°). In this case the learning experiment consisted in having the robot looking away from the table, then an object, reachable by one of the arms, was randomly positioned on the table. The robot then performed a perceptual action p_{vs} to look at the table, and it was recorded whether the robot saw the object or not. This was repeated for several minutes until 100 data points were recorded for each FoV value. There are several ways in which the robot could decide where to look (i.e. how the perceptual action p_{vs} can be implemented):

- **Random fixation:** One way is to fixate on some random location on the table. This gives variability to the fixation location, but the outcome is too unpredictable.

- **Centred fixation:** A fixation on the centre of the right/left hand side of the table covers most of the working space (for the right/left arm respectively), which increases the chances of finding an object. However, as the FoV narrows, the likelihood of finding an object decreases. More importantly, by always fixating on the centre, an object lying outside the FoV might never be found.

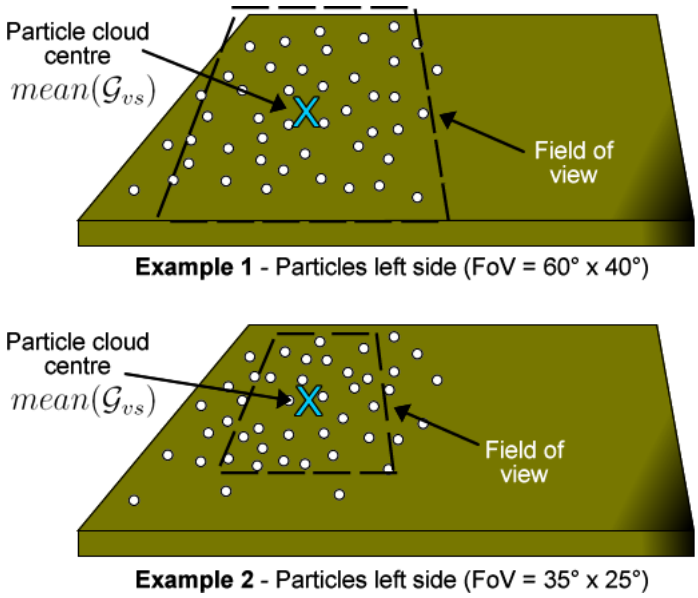

Figure 6.4: Examples showing how the perceptual action p_{vs} is generated for two different field of view values.

- **Hybrid fixation:** Our visual search process performs perceptual actions that combine the random and centred fixations. To do so we employ particles in order to determine the next fixation location. Every time that a perceptual action p_{vs} is executed, a set of particles $\mathcal{G}_{vs}$ is uniformly distributed across the table, similar to the initialisation step of the particle filter in Section 4.4.1.1[1]. Then, the fixation point is set as the mean of the cloud of particles ($mean(\mathcal{G}_{vs})$). Figure 6.4 illustrates this with two examples. The mean of the cloud of particles will tend to be located in the centre of the table. However, because the particles are randomly distributed every time that visual search is executed, the mean of the cloud will be different.

Following the hybrid fixation approach, Figure 6.5 shows the learnt conditional probabilities for $P(obj_{seeing}|p_{vs} = mean(\mathcal{G}_{vs}), obj_{new} = true)$. Notice that for the first three FoV values, (60°x40°), (50°x35°), and (40°x30°), the new object is seen/found with probability of 1.0. For the FoV of (35°x25°), the new object is seen with probability of 0.99. But, for the FoV of (25°x20°) a new object is seen with probability of 0.8. This last case means that the robot might have to perform a few visual search actions (p_{vs}) before finding the new object. Still, these probabilities demonstrate that the hybrid approach

[1]It is important to note that we are not using a particle filter for visual search, as no object is being tracked at this point.

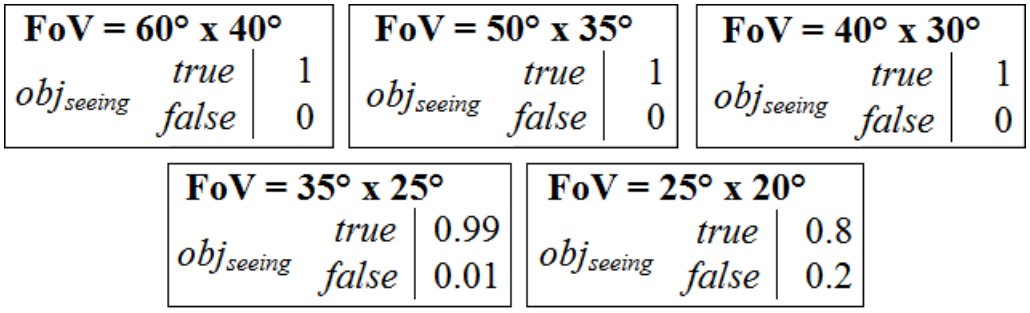

Figure 6.5: The values for $P(obj_{seeing}|p_{vs} = mean(\mathcal{G}_{vs}), obj_{new} = true)$ for all field of view values.

using particles is a good search strategy in general[2].

6.4 Results

To test the visual search process we performed the same experiment as in Section 5.4 using the pick & place task. The horizontal and vertical angles of the FoV were varied with the values: (60°x40°), (50°x35°), (40°x30°), (35°x25°), and (25°x20°). Figure 6.6 shows the average number of objects correctly placed in the containers and compares the *Rew+Unc+Gain* gaze scheme using the heuristic and using our proposed visual search process (*Rew+Unc+Gain+VS* in the figure). The graph also shows the results of the other gaze strategies, obtained in Section5.4, for comparison. The reach/grasp sensitivity is set to 1.0 cm and the observation noise is set to 1.0. A total of 15 trials of 5 minutes each were performed, and the error bars in the graph represent the 95% confidence intervals.

As expected, the visual search process helps to increase the task performance and robustness of the *Rew+Unc+Gain+VS* gaze strategy. Notice that when the FoV is large (i.e. (60°x40°) and (50°x35°)), the task performance is practically the same for both, the *Rew+Unc+Gain* and the *Rew+Unc+Gain+VS* schemes. When the FoV is smallest

[2]Actually, a better way to do the search strategy, especially for the smallest FoV, is not to resample the particles every time a perceptual action p_{vs} is executed. Rather, it is more efficient to select the view point that contains more particles. If an object is not found in that view point, then those particles are deleted, so that the next view point is again the one that contains the most particles. Once an object is found the particles can be redistributed again across the table. To do this, however, it is required a mapping from the future robot configuration to image coordinates. The iCub simulator does not have this implemented, and due to lack of time we could not implemented it. Nevertheless, it is part of our future work (Section 9.2.1).

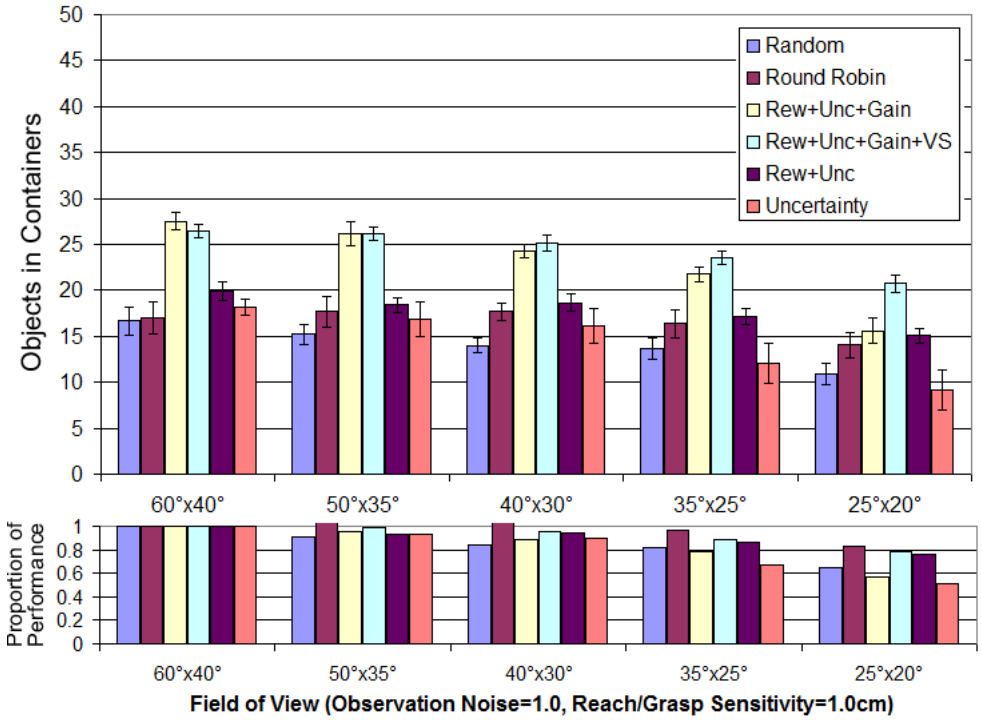

Figure 6.6: Results for the visual search analysis, with reach/grasp sensitivity = 1.0 cm and observation noise = 1.0. The top graph shows the task performance, whilst the lower graph shows the proportion of actual performance compared to the best case for each strategy. The error bars represent the 95% confidence intervals.

(i.e. (25°x20°)), the *Rew+Unc+Gain+VS* gaze strategy performs much better than the *Rew+Unc+Gain* gaze strategy. The lower graph of Figure 6.6 shows the proportion of actual performance compared to the best case for each strategy. This determines the robustness of the gaze schemes to changes in the FoV. In this case *Rew+Unc+Gain+VS* and *Round Robin* strategies are more robust to changes in the FoV. The two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain+VS* scheme against the *Rew+Unc+Gain* strategy. The differences for the first three FoV values are not statistically significant, whilst for the values of (35°x25°) and (25°x20°) the differences are statistically significant at $p < .001$.

6.5 Discussion

First we can compare the average number of visual searches using the heuristic and using the proposed visual search process. Figure 6.7 shows such comparison, along with the averages of the other gaze strategies. As stated above, the heuristic (*Rew+Unc+Gain*)

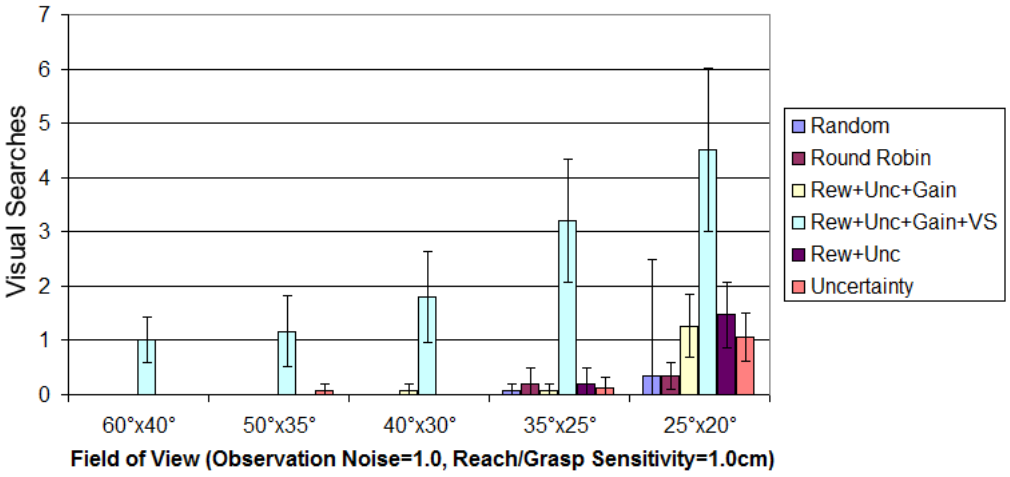

Figure 6.7: Average number of visual searches using the heuristic for all gaze strategies, and using the visual search process for the *Rew+Unc+Gain+VS* gaze strategy. The error bars represent the 95 % confidence intervals.

is never used when the FoV is large. Notice that, in contrast, the visual search process (*Rew+Unc+Gain+VS*) is employed for all FoV values. This means that at some points in time one of the arms might have been left with no objects to work with. Thus, a visual search action was executed. As expected, the number of times that a visual search action is performed increases as the FoV narrows.

It is important to notice that adding the visual search process to the *Rew+Unc+Gain* gaze strategy it does not incur in any considerable cost. This is demonstrated in Figure 6.6 for the FoV values of (60˚x40˚) and (50˚x35˚), where the task performance remains practically the same with and without the visual search process.

The most important thing is that the visual search increases considerably the task performance when the FoV is narrower (25˚x20˚). The *Rew+Unc+Gain* gaze strategy was already the best in terms of task performance, but now with the added visual search it is clearly better than it was before. Obviously, we would also need to add the same visual search process to the *Rew+Unc* and *Uncertainty* gaze schemes for a fair comparison.

In terms of robustness to changes in the FoV the increase is also good. Now it is always better than the robustness of the *Rew+Unc* gaze strategy (using the simple heuristic). However, the robustness of the *Round Robin* gaze strategy is still slightly better. A reason for this is simply because the performance that the *Rew+Unc+Gain+VS*

scheme needs to maintain is much higher than that of *Round Robin*, i.e. the performance of *Round Robin* is always mediocre but is robustly mediocre.

6.6 Conclusions

The problem of visual search is to find a target object, or landmark, in a visual scene. In this chapter we have integrated an active visual search process into our existing *Rew+Unc+Gain* gaze control model. The results demonstrate that with the aid of visual search the integrated gaze model (*Rew+Unc+Gain+VS*) increased its task performance and robustness to changes in the FoV.

It is important to note that the visual search process presented here does not attempt to generalise to all kinds of visual search tasks. In this case our focus was on the pick & place task. Nevertheless, as opposed to most work on visual search, we have provided an integrated account of a visual search process and a task based gaze control model. Future work should be done to determine the usefulness of our visual search process in other manipulation tasks.

Based on our work, we can point out at least three relevant issues related to visual search:

- **What is being searched for:** Typically, the aim is to find one target in a visual scene, as in our task and most of those reviewed at the beginning of this chapter. The agent should acquire or have some prior knowledge about the target in order to find it. A more complex case results when the aim is to search for more than one target object. In our case, the visual search process was formulated with only one target object in mind.

- **The agent's task(s):** Even though there are many situations in which a visual search task may be the only goal of the agent. There are also other situations in which visual search should be executed simultaneously with other of the agent's goal(s) or task(s). This requires some form of coordination or integration mechanism so that gaze can be shared amongst the tasks. This mechanism should

trade-off the information that is currently needed by the task(s) and the information that may be gained by visual search.

- **The search strategy:** The strategy involves the technique(s) employed to actually find the target object(s). The strategy may be constrained according to the agent's task(s) and/or the prior information about the environment. In our case, the task constrains the search to specific objects, and the environment constrains the search to the tabletop. Thus, in the pick & place task the search space is small, compared to other tasks or scenarios where search may take place in one or several rooms. As mentioned above, our search strategy could be more efficient and also more general (In Section 9.2.1 we explain how the current visual search process could be extended).

Chapter 7

Gaze Control in a Bimanual Task

For the pick & place task (Section 1.6) used thus far in the thesis, the assumption is made that the robot's arms act independently, i.e. in *parallel*. Each manipulation motor system executes a policy (i.e. a sub-task) whilst gaze is shared amongst them. However, many real world tasks require the coordination of *concurrent* or *interactive* manipulation actions[1]. For example, passing an object from one hand to the other, lifting an object with both hands, opening a bottle, pouring water into a cup while holding it, etc.

This chapter presents a variation of the pick & place task where the left container is removed from the table. Thus the robot needs to transfer the objects appearing on the table's left side to the right hand in order to put them into the container (Figure 7.1). Based on this new *bimanual* task, our gaze control models are compared in terms of task performance as in Chapter 5.

The implementation of concurrent actions in robots requires efficient and reliable coordination mechanisms between the interacting motor systems [241]. This kind of work is outside the scope of this thesis. This chapter presents a practical, and rather specific, implementation of a bimanual action that allows the robot to transfer objects from one hand to the other. The aim of this chapter is to further analyse our models of gaze control in a different scenario. Concurrent actions force two or more manipu-

[1]Both, parallelism and concurrency refer to processes that can be executed simultaneously. Parallel processes do not depend on each other, so no interaction is needed. In contrast, concurrent processes depend on each other for their correct execution, so normally a coordination mechanism is required to control the processes.

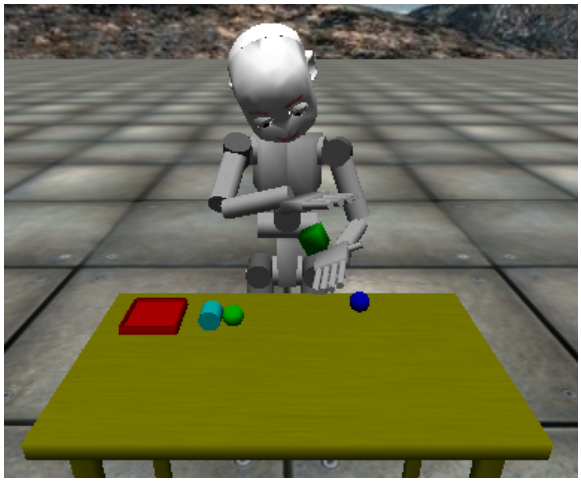

Figure 7.1: The iCub simulator performing the bimanual pick & place task.

lation motor systems to work together in order to successfully achieve some goal. The consequence of this is that gaze should assist, perhaps all or some of, the manipulation motor systems involved in the concurrent action simultaneously.

First, this chapter starts by describing the differences between parallel and concurrent manipulation motor systems and how the gaze control might be affected. Second, we explain how our bimanual task is modelled. Third, it is explained how the robot learns the bimanual task. Fourth, the results of our experiment and the analysis of such results is presented and discussed. Finally, some conclusions are provided.

7.1 Configurations of Manipulation Motor Systems

In order to understand how gaze control can be affected by parallel or concurrent actions, let us first define the possible configurations that can be obtained based on the number of manipulation motor systems considered. In this thesis, three possible configurations are defined: i) single motor system, ii) multiple parallel motor systems, and iii) multiple concurrent motor systems. These configurations are illustrated in Figure 7.2. It is important to point out that, even though this chapter focuses on manipulation, these configurations are intended to be general and include all types of motor systems (i.e. arms, legs, wheels, torso, etc.). For instance, Surdilovic et al. [242] propose a similar classification that is specific to dual-arm operations. In fact, it is also possible to view

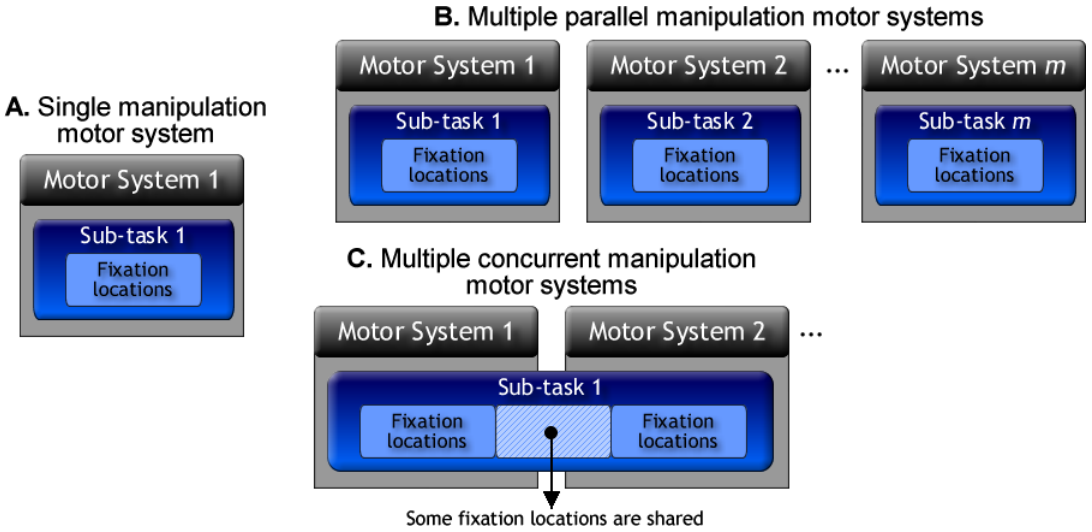

Figure 7.2: Configurations of manipulation motor system. A. Single motor system. B. Multiple parallel motor systems. C. Multiple concurrent motor systems.

each motor system as an agent and treat the multiple motor system problem as a multi-agent system [243].

As explained in Section 4.1, our formulation associates each manipulation motor system with a particular sub-task, and the configurations defined here reflect this fact. In this thesis, we define parallelism and concurrency at the motor level. This, of course, does not rule out the possibility of having parallelism or concurrency within each motor system. For instance, Sprague and Ballard [7] consider a single motor system capable of executing concurrent sub-tasks. By considering multiple parallel or concurrent sub-tasks within each motor system, the number of possible configurations increases. However, our research so far has only focused on manipulation motor systems with single sub-tasks.

7.1.1 Single Manipulation Motor System

The simplest motor configuration is to consider a single manipulation motor system (Figure 7.2A). Notice that, since there is a single sub-task in execution then gaze does not need to be *shared*. Perception will always be available to that motor system, thus the *gaze allocation* problem disappears in this configuration. But note that the problem of *where to look* is still present. The sub-task should still have a set of fixation locations and only one can be selected at a time. In fact, this configuration will be explored in

Chapter 8 in order to model Johansson's task. Here gaze control would be expected to work in the same way as explained in Chapter 4 except for gaze allocation which is not considered.

Some of the works reviewed in Section 3.2 follow this configuration, where only a single task is pursued. For example, for reaching/grasping [35, 196, 197, 244], catching [245] and navigation [191]. As mentioned above, other works consider multi-tasking in a single motor system. For instance, for navigation, collection and avoidance of objects [7], and for localisation and mapping [166].

7.1.2 Multiple Parallel Manipulation Motor Systems

This configuration is the one that has been used in the thesis so far for the pick & place task (Chapter 4) (Figure 7.2B). Each manipulation motor system is assumed to be able to perform a particular sub-task, where each sub-task has its own set of fixation locations. Because we consider that there is only one *perceptual motor system* (which includes the robot's head, neck and eyes) then gaze needs to be shared amongst the manipulation motor systems (or sub-tasks), which brings up the problem of *gaze allocation*.

In this configuration we can identify two cases with respect to parallel sub-tasks. First, the sub-tasks might all contribute towards a common goal. Second, the sub-tasks might have individual and non-related goals. Our pick & place task follows the first case, since the task performance is measured by the total number of objects placed by both hands. For the second case each arm would be doing different sub-tasks, for example an arm could be stacking objects whilst the other arm sorts another kind of object. Notice that even in this last case, gaze still needs to be shared to assist each manipulation motor system.

Srinivasan [246] provides a series of psychophysical experiments for parallel reaching actions in humans, along with the formulation of a two-handed control model for reaching that uses visual feedback information. Beetz et al. [247] showed how two robots cooperate to make pancakes. One of the robots, for example, is able to open the fridge

with one hand and grasp an object with the other. The arms work towards a common goal but there is no direct interaction with both hands. A rather extreme case is when the motor systems are not part of the same "body" but still there is some kind of co-operation in order to contribute to a common goal. For instance, robotic arms working on the same assembly line [248], or multi-agent systems in a foraging task in the special case where agents work independently [249].

7.1.3 Multiple Concurrent Manipulation Motor Systems

Parallel manipulation motor systems run simultaneously and may cooperate to achieve a common goal, however they are not required to interact directly with each other. In fact, parallel motor systems do not necessarily need to know about the existence of the other manipulation motor systems. An extension to this parallel configuration is to allow different motor systems to interact with each other, to some degree, by introducing concurrent manipulation actions.

Concurrency already implies that the motor systems are expected to be running at the same time, but it also implies that motor systems must be aware of each other. This is because some manipulation actions can only be successfully achieved by the interaction of two or more motor systems. This configuration is illustrated in Figure 7.2C. Notice how there is no longer a sub-task associated with each motor system, now there is a shared sub-task where the manipulation motor systems are expected to perform together. As will be explained below, when modelling concurrent motor systems part of the state and action spaces of each manipulation motor system is now shared. In this case the task cannot be subdivided as clearly as for parallel systems.

With respect to the available fixation locations, they can still be associated with a particular manipulation motor system. However, depending on the implementation, it is likely that a subset of fixation locations will be shared or be common amongst some motor systems (see Figure 7.2C). This means that in some situations a single fixation would be able to assist several manipulation motor systems simultaneously. These situations are present in our system because of (at least) two reasons:

1. **Field of view (FoV):** In our bimanual task the field of view is large enough to acquire the complete view of where the concurrent action is taking place. For example, in our task the transfer of an object can be captured by a single fixation location when the robot looks at the object being transferred.

2. **Uniform image raster:** The robot's cameras acquire images with the same resolution across the image. This means that, together with the FoV, a fixation to the centre of where the manipulation action takes place is likely to be sufficient to obtain the information needed to achieve the concurrent action successfully.

Notice that, if the FoV narrows or the rasterisation of images is not homogeneous, then it is likely that a single fixation cannot assist all the manipulation motor systems. Therefore, selection of fixation locations *within* the concurrent action may have to take place.

7.1.3.1 Related Work

On a day-to-day basis we perform a large number of concurrent tasks using our body motor systems. In Section 3.1.2 some psychophysical studies were reviewed where concurrent actions take place. For example the tea-making task [109], where an action such as "fill the kettle with water" depends on the correct interaction between both arms (one to hold the kettle and another to open/close the tap water). The sandwich making task [113] is another example where concurrent actions such as "spread peanut butter on the bread" requires the coordination of both hands. A final example is a stacking task [108] where subjects need to build cup pyramids using both arms. These examples also illustrate the situation where a single fixation location may assist the whole concurrent task, although the inhomogeneity of the human fovea is likely to make the eyes move to different parts of the object(s) that are taking place in the concurrent action.

In the case of robotic platforms, the most common example of concurrent tasks are from those realised by humanoid robots. By having dual-arms, humanoid robots are expected to perform concurrent actions that involve cooperation and interaction of both motor systems. For instance, lifting and moving objects with both arms, handling

tools, assembly of pieces, cloth manipulation, transferring objects, etc. [241,247,250,251]. Similarly to parallel motor systems, multi-agent systems are typically expected to engage in concurrent tasks. Some examples of these tasks are, foraging (with communication amongst the agents), pushing/pulling objects, playing cooperative sports (e.g. soccer), herding, cooperative navigation, etc. [243].

7.2 Modelling the Bimanual Pick & Place Task

The bimanual task is modelled following the same decision-theoretic techniques employed in Section 4.1. Recall, that for the case of parallel motor systems, each manipulation motor system ms was modelled as a semi-Markov decision process (SMDP) (Section 2.4) with a discrete state space and a set of temporally extended actions (options). Here, the same approach is followed except that each manipulation motor system might contain extra state variables that will be used for coordination.

As in Section 4.1, the state space of the bimanual pick & place task is modelled following a factorised representation. Table 7.1 and 7.2 presents the state spaces for the right and left arm respectively, whilst Table 7.3 defines the set of options available to the right arm. The set of options for the left arm is not shown, but it is exactly the same as the right arm except for *moveToContainer* which is not needed. Next to each option their completion time is shown, defined as a Gaussian distribution and specified by the mean completion time and standard deviation (μ_o, σ_o) in seconds. These time distributions are the same as those defined in Section 4.1. Notice that there are a few differences in the state spaces compared to the pick & place task from Chapter 4 (defined in Section 4.1). First, now the state variable *armPosition* contains the extra value *inTransfer*. Second, each arm contains a state variable that indicates the state of the other arm: $leftTransferStatus$ and $rightTransferStatus$ for the right and left arm respectively. Third, the value *onContainer* for the state variable *armPosition* has been removed from the left arm, since there is no left container.

Since our main purpose is to analyse the behaviour of our gaze control models con-

Table 7.1: Factorised state space for the right arm.

State Variable	*State Value*
armPosition	{onObject, onTable, onContainer, inTransfer}
handStatus	{grasping, empty}
transferStatus	{ready, notReady}
leftTransferStatus	{ready, notReady}

Table 7.2: Factorised state space for the left arm.

State Variable	*State Value*
armPosition	{onObject, onTable, inTransfer}
handStatus	{grasping, empty}
transferStatus	{ready, notReady}
rightTransferStatus	{ready, notReady}

Table 7.3: Options for the right arm.

Options	*Avg. completion time (secs)*	*Stand. deviation (secs)*
moveToObject	2.65	0.56
moveToTable	2.95	0.45
moveToTransfer	2.65	0.56
moveToContainer	3.25	0.33
graspObject	2.87	1.08
releaseObject	1.0	0.2
noAction	0.01	-

sidering concurrent actions, the problem of transferring an object from one hand to the other has been simplified[2]. This simplification consists of pre-defining a "transfer area" where the hands would move in order to pass the object. This area is located in the centre front of the robot. When the right or left hand reach this transfer area the variable *armPosition* takes the value of *inTransfer*. At the same time the variable *leftTransferStatus* or *rightTransferStatus* will take the value *ready*. In order to move to the transfer area the arms employ the option $moveToTransfer$. A further simplification consist in predefining the orientation of the left hand whilst executing $moveToTransfer$. In this case, the hand changes its orientation so that the palm of the hand ends up facing upwards. That way the right hand can move above the object in order to pick it up (see Figure 7.1). This simplification works fine for our current goal, however this concurrent action should be formulated in a principled way in order to learn or plan a coordination mechanism for object transfer (e.g. see [250, 251]). Sections 7.3 and 7.4 describe how this transfer is learnt and executed by the robot.

7.2.1 Related Work

Our modelling approach is similar to the *concurrent action model* (CAM) proposed by Rohanimanesh and Mahadevan [252]. In CAM all the agent's sub-tasks or goals are modelled using a single SMDP with options. An application of CAM for modelling a multi-agent system is presented in [253] and [254]. Rohanimanesh and Mahadevan also proposed a more complex model called the *coarticulation* model [252]. This model makes an explicit decomposition of sub-tasks, as in our approach, but it is more general as it allows for the modelling of a much larger space of concurrent problems. Coarticulation allows for sub-optimal actions to be executed in order to concurrently satisfy multiple goals. As opposed to our approach, coarticulation does not make an explicit separation between motor systems. Mausam and Weld [255] model parallel planning problems with concurrent MDPs (CoMDPs). They also allow for the execution of durative actions but instead of using SMDPs, as in this thesis, they employ a technique called *concurrent*

[2]Recall that we perform a further simplification for the problem of grasping, where the robot uses the magnet-like function, as explained in Section 4.1.

probabilistic temporal planning (CPTP). This technique allows for the formulation of durative actions as CoMDPs. Younes and Simmons [256] have proposed the model called generalised semi-Markov decision process (GSMDP), which extends SMDPs to model concurrent durative actions explicitly.

Other approaches involve the use of partially observable Markov decision processes (POMDPs) (Section 2.3). Erez et al. [45] defined a POMDP model for hand-eye coordination in a simulated reaching task, where two "hands" must reach to a common point whilst passing a pair of obstacles. They model the joint behaviour of eye and hands in the POMDP, whereas our model splits the decision problem into physical and perceptual action selection. Roth et al. [257] employs POMDPs for modelling a multi-agent system.

7.3 Learning the Task

Learning parallel sub-tasks is relatively straightforward because each sub-task can be learnt separately and then executed together. However, by allowing concurrent actions, some degree of interaction is expected between different manipulation motor systems. This interaction means that all manipulation motor systems should be considered during learning. Even though we have simplified in some way the transfer of objects, the robot still needs to learn how to coordinate both arms in order to successfully pass objects between hands.

In Section 4.3 we explained how a separate computational learning programme was implemented for learning the pick & place task, instead of using the iCub simulator directly. Here the same approach is followed, and the same *SMDP Q-learning* algorithm is employed (see Section 4.3). Each arm learns its own policy as before, the main difference here is that both policies are learnt simultaneously. This is because the state of one arm affects the state of the other and vice-versa (via the state variables *leftTransferStatus* or *rightTransferStatus*).

Learning is once again carried out under an assumption of complete observability,

i.e. the visual memory contains the complete list of landmarks (i.e. objects and the container) and their true location. The number of objects that appear on the table is limited to 10. A minimal time-to-goal strategy is followed. For any option taken the robot receives -1 units of reward, and 0 units of reward when the task is completed, i.e. when all objects are put in the containers. A special issue found at this point was that by using this reward function the robot learnt to first put all reachable objects with the right arm and then started transferring objects from the left side. This policy is of course valid, however it would be more interesting to interleave the transfer of objects. This can be done in two ways: i) modifying the reward strategy, or ii) modifying the arrival of objects during the task (particularly for the right hand side). In this Chapter we follow the first option, whilst the second is left for future work[3]. Therefore, if the left arm is ready to transfer an object ($armPosition_{left} = inTransfer$), every option performed by the right arm receives -5 units of rewards except for the option *moveToTransfer*. This is to prioritise the transferring of objects, otherwise the right arm will first put all the reachable objects and then start transferring objects from the left side. By prioritising, both sub-tasks are interleaved.

7.3.1 Related Work

Most of the previous work in learning is related to multi-agent systems (Busoniu et al. [258] provides a comprehensive survey of multi-agent reinforcement learning algorithms). Guestrin et al. [259] developed a coordinated reinforcement learning approach for multi-agent systems using Markov decision processes (MDPs), where each agent knows only part of the complete state of the system. Thus, agents need to share information amongst them. In our case, all motor systems belong to the same agent so the state of each motor system is known to the others. Banerjee et al. [260] proposed an on-policy concurrent reinforcement learning algorithm using MDPs for a multi-agent system in a competitive games domain. As opposed to Guestrin's model, here the agents know the state of the

[3]If objects stop appearing for a while on the right hand side, then the robot would have time to transfer objects coming from the left hand. Thus the robot would learn how to interleave both sub-tasks and the reward strategy would remain the same. The same arrival strategy must need to be implemented during the execution phase.

others. These models make use of MDPs which do not allow the modelling of temporally extended actions as our approach does.

Our learning approach is similar to that of Rohanimanesh and Mahadevan [252], as they also use the SMDP Q-learning algorithm. For the concurrent action model (CAM) a single policy is learnt for all sub-tasks, but for their coarticulation model a policy is learnt for each sub-task. However, as explained above, the coarticulation model does not separate the motor systems explicitly as we do. Ghavamzadeh et al. [254] provides an extension to CAM for multi-agent systems using SMDPs (MSMDPs). Furthermore, the authors propose a hierarchical reinforcement learning algorithm which decomposes the task in a similar way as our approach, but where sub-tasks are performed by agents. This provides a more general learning framework than the one we follow. Sub-tasks can be defined to be cooperative or non-cooperative. The framework is able to learn the sub-task, their priorities and how they can be coordinated. In future work our gaze control models could be integrated into this framework.

7.4 Executing the Task

Once the task is learnt the robot is now in charge of maintaining the visual memory (as explained in Section 4.4), i.e. the robot has to look at the landmarks in order to estimate their location by means of the gaze control strategies described in Section 4.5.

The robot executes the bimanual task in the same way as the pick & place task. The key difference is the transfer of objects from the left to the right hand side. For example, to transfer an object the robot should perform the following steps:

1. The left hand (having grasped an object) moves to the predefined transfer area (using the option $moveToTransfer$), this sets $armPosition_{left} = inTransfer$ and $leftTransferStatusright = ready$, thus notifying the right arm about the state of the left arm. Since the transfer area has been predefined, the left arm knows exactly where to move. Also, as mentioned in Section 7.2, whilst the left arm moves to the transfer are the palm changes its orientation so that it ends

up facing upwards. In this way the object being grasped by the left hand can be picked up by the right hand (see Figure 7.1).

2. Once the right arm notices that the left arm is ready to transfer an object, it will "try" to move to the transfer area (using the option $moveToTransfer$). In this case the right arm must estimate the location of the object to be transferred by looking at it (in the same way as objects on the table are grasped). Only if the right hand reaches the object to be transferred correctly (i.e. the hand's centre lies within certain threshold value with respect to the object's centre, as explained in Section 4.4.2), then it sets the state variables $armPosition_{right} = inTransfer$ and $rightTransferStatusleft = ready$, to notify the left arm.

3. Only when both arms are ready to transfer the object ($rightTransferStatusleft = ready$ and $leftTransferStatusright = ready$) is grasped by the right arm and released by the left.

In terms of the gaze control strategy, the only difference with the pick & place task is that in this case both arms share one fixation location, i.e. the object to be transferred. However, the fixation selection process for our three strategies remains the same as before. Next we present and discuss the results of an experiment where the robot performs the bimanual task.

7.5 Results

In order to test the bimanual task, a similar experiment to the one described in Section 5.2 was performed. Here, the sensitivity in the manipulation actions was varied with three threshold values: 0.5, 1.0 and 1.5 cm. Figure 7.3 shows the average number of objects correctly placed in the right container for all five gaze strategies. A total of 10 trials of 5 minutes each were performed for each gaze strategy. The observation noise is set to 1.0 and the FoV is set to 60°x40°. The error bars in the graph represent the 95% confidence intervals.

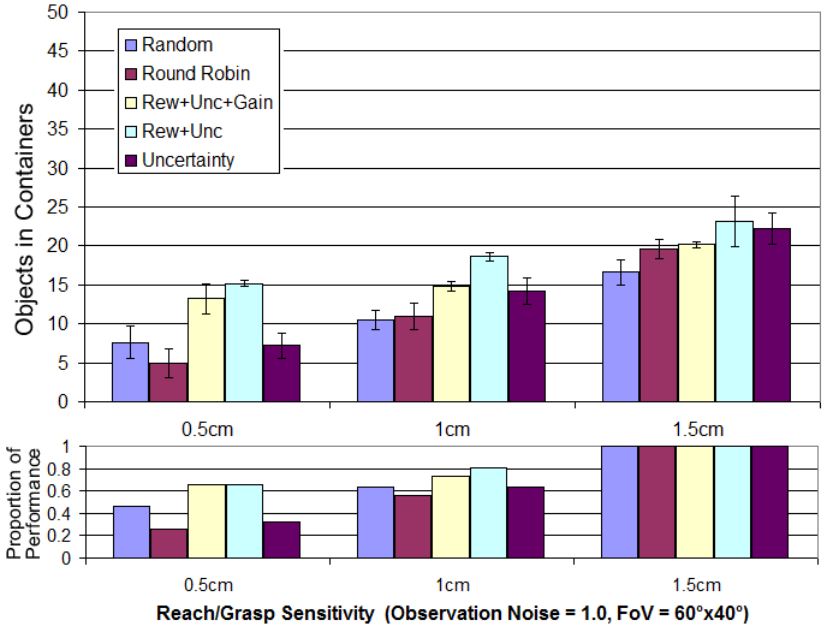

Figure 7.3: Results for the bimanual task when varying the reach/grasp sensitivity, with observation noise = 1.0 and field of view = 60°x40°. The top graph shows the task performance and the lower graph shows the proportion of actual performance compared to the best case for each strategy. The error bars represent the 95% confidence intervals.

Similar to the results obtained in Section 5.2, the performance of all gaze strategies drops as the sensitivity increases. In terms of task performance the *Rew+Unc* gaze scheme outperforms all other strategies, whilst the second best is the *Rew+Unc+Gain* gaze strategy, for the threshold values of 0.5 and 1.0 cm. For the value of 1.5 cm, the *Uncertainty* gaze scheme is better than *Rew+Unc+Gain*. The lower graph of Figure 7.3 shows the proportion of actual performance compared to the best case for each strategy. This determines the robustness of the gaze schemes to changes in the sensitivity values. In this case the *Rew+Unc* gaze strategy is the most robust to changes in the threshold values, closely followed by the *Rew+Unc+Gain* gaze scheme.

The two-way ANOVA test was used with the threshold values and gaze strategies as factors. For both factors the differences are statistically significant at $p < .0075$. Then, using the two-tailed unpaired t-test, the results of the *Rew+Unc+Gain* gaze strategy were compared against the rest of the schemes. For the threshold value of 0.5 cm, all differences are statistically significant at $p < .005$, except against the *Rew+Unc* gaze strategy. For the threshold value of 1.0 cm, the differences for *Random* and *Round Robin* are statistically significant at $p < .0007$, except when compared against the *Rew+Unc*

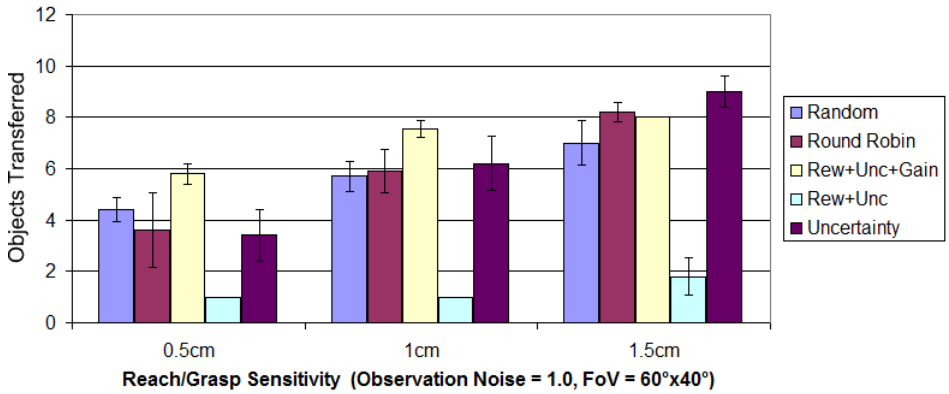

Figure 7.4: Average number of objects transferred under each gaze control strategy. The error bars represent the 95% confidence intervals.

and the *Uncertainty* strategies. Finally, for the threshold value of 1.5 cm only the difference against *Random* is statistically significant at *Random* at p <.003.

7.6 Discussion

The results show that in terms of task performance and robustness the *Rew+Unc* gaze strategy is better than *Rew+Unc+Gain*. However, a careful examination of the results reveal some interesting points that need to be taken into consideration.

Even though the transfer of objects was learnt to be a priority (Section 7.3), it turns out that the behaviour of the *Rew+Unc* gaze scheme disregards almost completely the transfer of objects[4]. Figure 7.4 presents the average number of objects transferred under each gaze strategy. Notice that the *Rew+Unc* strategy transfers the least number of objects and just a few of them on average. In fact, for the threshold values of 0.5 and 1.0 cm there is only one transfer per trial. By favouring only one arm its performance increases with respect to the rest of the strategies, since passing of objects takes more time. This behaviour emerges because, as explained in Section 5.2.2, this gaze scheme is likely to look more often at landmarks with high predicted value. As the object to be transferred becomes more uncertain because it is not fixated, then its predicted value

[4]During the learning phase the number of objects is limited to 10, and since the task is accomplished until after all objects are put in the container, then the robot has to transfer objects. On the other hand, during the execution phase objects keep appearing on the table every minute, every time an object is put in the container, or every time an object falls from the table. This means that the right arm can continuously find new objects to manipulate, so no transfer is necessary to continue the task. This is why the *Rew+Unc* gaze scheme can disregard the transfer of objects.

drops, therefore it might be the case that the gaze strategy will never choose to look at that object.

There are two ways in which these results can be analysed: i) in terms of task performance, and ii) in terms of the balance of workload between the arms. First, if only the task performance is taken into consideration, then the *Rew+Unc* gaze strategy provides the best overall performance. It is interesting to see that following the *Rew+Unc* scheme, the robot exploits the fact that all objects yield the same value. Therefore, it makes sense to use only one arm since it can place objects faster in the container. However, if the results are analysed in terms of the workload of each arm, then the *Rew+Unc* gaze strategy is actually the worst. From Figure 7.4 it is clear that the *Rew+Unc* scheme transfers the least number of objects in average per trial. Therefore, if we were to look at the average number of transferred objects as well the task performance, then we might conclude that the *Rew+Unc+Gain* and the *Uncertainty* schemes produce better results.

7.7 Conclusions

This chapter examined the behaviour of our models of gaze control in a bimanual task, which is a variation of the pick & place task employed in previous chapters of this thesis. The bimanual task allows for the transfer of objects from the left hand to the right. This action is considered to be concurrent as both hands interact with each other in order to successfully achieve the transfer. The aim of the chapter was not centred on the concurrent coordination of the manipulation motor systems, but on the gaze control analysis.

Even though the task was learnt in such way that the transfer of objects had higher priority, our results suggest that the behaviour of each gaze control strategy determines the number of objects being transferred. This situation also emerges because all objects provide the same value. It was found that the *Rew+Unc* gaze strategy had the best overall performance. However, closer examination showed that the workload of each arm under the *Rew+Unc* scheme is not balanced, as only around 1 object is transferred

per trial. This means that the right arm is doing all the work. Therefore, if we take into account the average number of transferred objects, then we might conclude that the *Rew+Unc+Gain* and the *Uncertainty* schemes yield a more balanced workload between the two arms.

As mentioned above, during the learning phase there are other ways in which the task could be learnt. One way we discussed is that instead of changing the reward function, we can modify the way in which objects arrive during the task. In fact, the workload can be somehow controlled by changing the arrival of objects. Therefore, we can hypothesise that the workload resulting from the *Rew+Unc* scheme could be balanced. Thus, it may be possible to compare all gaze strategies in terms of task performance alone.

Chapter 8

Analysis of Human Gaze Data

So far, our candidate models of gaze control have been characterised and analysed following an engineering approach. One of the main conclusions has been that the model of gaze control based on rewards, uncertainty and gain (*Rew+Unc+Gain*) (formulated in Section 4.5.3) is the best option in terms of task performance and robustness, in terms to three environmental variables (Chapter 5), compared to our two other candidate models (*Rew+Unc* and *Uncertainty* schemes). However, it is also one of our main interests to determine whether or not our candidate gaze control models are able to reproduce behavioural human data.

This chapter presents the results obtained for the simulation of a psychophysical experiment devised by Johansson et al. [1] to study the eye-hand coordination in humans. Thus, this chapter starts by explaining Johansson's task in detail. Then, a description of how the same task is modelled and implemented in the iCub simulator is provided. Finally, a comparison and analysis of the results obtained with our candidate models of gaze control and the behavioural data presented by Johansson et al. [1,8] are discussed.

8.1 Johansson's Task

Chapter 3 presented an account of the several psychophysical experiments that try to understand how gaze and actions are coordinated by humans. Tasks such as, copying blocks [21], sorting virtual objects [23], sorting clothes [107], making a jelly sandwich

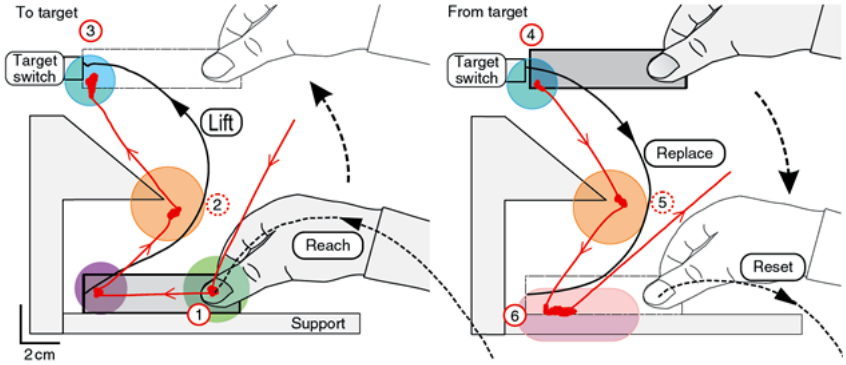

Figure 8.1: Schematic view of the manipulation task as defined in [1] (Figure taken from [8]) (Figure courtesy R. Johansson).

[113], making a cup of tea [109], playing cricket [114], playing squash [116] and driving [22, 118], they all involve some kind of manipulation of objects and the control of gaze. Of particular interest is the work of Johansson et al. [1], investigating the precise spatio-temporal relationship between gaze and actions in a simple manipulation task. One of the main findings was that gaze fixations specify landmark positions to which the manipulation actions are subsequently directed. This supports the hypothesis that gaze is deployed in order to guide manipulation actions.

Figure 8.1 shows a reproduction of the schematic view of the manipulation task as presented in [1]. Subjects have to reach for and grasp the bar located on the table. Then they have to move the bar and touch the target switch with the tip of the bar whilst avoiding the obstacle in the middle of the path. After touching the target, the bar had to be placed again on top of the support surface. The task is divided into the following six action phases: *pre-reach, reach, load & lift, target switch, replace & unload*, and *reset* (these phases are illustrated in Figure 8.1). The proposed fixational landmarks were defined by looking at the places where subjects fixated the most during the performance of the task. These places are represented by the coloured ellipsoids in Figure 8.1, where its shape determines the extension at which subjects fixated on that particular area. Based on these coloured areas, there are five proposed landmarks: *the bar, the tip of the bar, the obstacle, the target* and *the support surface*. For our simulation, the same action phases and landmarks are considered.

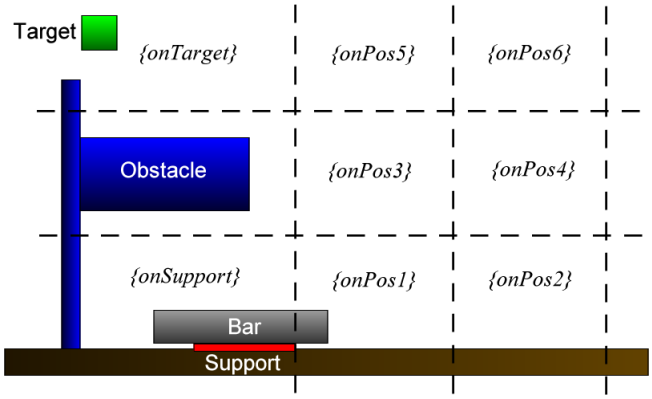

Figure 8.2: The setup for the simulation of Johansson's task defined in a discrete state space.

8.2 Simulating the Task

Following the same procedure as in Section 4.1, the task is first modelled using a discrete state representation, which is defined in Table 8.1. Figure 8.2 shows a graphical representation of the values for the state variable *armPosition*. Notice that this task requires the use of only one arm (right arm), thus only $\mathcal{S}_{right\_arm}$ needs to be defined (as opposed to the pick & place task used so far in the thesis).

Table 8.1: Factorised state space for Johansson's task

State Variable	*State Value*
armPosition	{onTable, onBar, onTarget, onPos1, onPos2, onPos3, onPos4, onPos5, onPos6, onSupport}
handStatus	{graspingBar, handEmpty}
targetStatus	{targetTouched, targetUntouched}

The set of options available for the right arm ($\mathcal{O}_{right\_arm}$) is defined in Table 8.2. Next to each option is the average completion time in seconds, and its standard deviation. These time values are *assumed* to be the same as those used in the pick & place task defined in Section 4.1. Recall that to obtain these time values for the pick & place task we executed all options in sequence for 60 minutes. Because the location of objects is not discrete, the completion time for each manipulation action varies. This variation results because actions do not start exactly from the same position at all times. However, because the workspace in the pick & place task and Johansson's task is similar we consider that all the *moveTo*X options are the same as the option *moveToObject* from the pick & place task. The option *moveToObject* can move short distances (e.g. when

an object is next to a container), or not so short distances (e.g. when an object is in the table's centre, the farthest from a container). Thus, this option already summarises different distances that are similar to the different positions in Johansson's task (shown in Figure 8.2).

Table 8.2: Options for the right arm (Johansson's task).

Options	*Avg. completion time (secs)*	*Stand. deviation*
moveToBar, moveToTarget, moveToPos1, moveToPos2, moveToPos3, moveToPos4, moveToPos5, moveToPos6	2.65	0.56
graspBar	2.87	1.08
releaseBar	1.0	0.2
noAction	0.01	-

The visual memory is implemented in the same way as in Section 4.2. The only difference is that the landmarks e_i used in this task are the bar, the tip of the bar, the obstacle, the target and the support surface. Associated with each landmark is its location distribution defined by $pos(e_i)$, and represented by a particle filter. In this respect, the location uncertainty of the bar and the support surface are treated in the same way as the objects and containers from the pick & place task. However, the location uncertainty of the tip of the bar, the obstacle and the target is treated slightly different. As explained in Section 8.1, the bar needs to avoid the obstacle and then touch the target. Therefore, the robot needs to estimate the distance between the tip of the bar and the obstacle, and the tip of the bar and the target, respectively. This is achieved by looking at the tip of the bar and then the obstacle (or the target) to estimate their respective location first and then their relative distance[1].

Once the task has been modelled, the robot learns to perform the task via reinforcement learning, in the same way as explained in Section 4.3[2]. Learning takes place under the assumption of complete observability, i.e. the visual memory contains the complete

[1]Rather than looking at one landmark first and then the other, we could try to estimate the distance between both landmarks in one fixation. However, we follow the former approach because our visual memory has been defined in such way that each landmark is associated with a particular particle filter. To consider the latter approach we would have to associate one particle filter to multiple landmarks.

[2]Section 4.3 explained how a special computational learning programme was implemented separately from the iCub simulator in order to learn the task. Here the same learning programme was modified in order to learn Johansson's task.

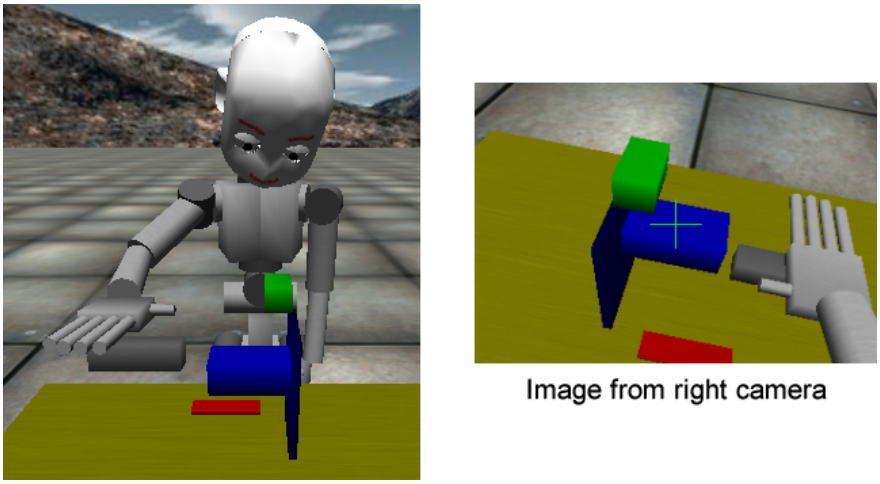

Figure 8.3: The robot performing Johansson's task in the iCub simulator and the current fixation point.

list of landmarks and their true location. Again, a minimal time-to-goal strategy is followed. For any option taken the robot receives -1 units of reward, and 0 units of reward when the task is completed, i.e. when the robot puts the bar back on the support surface. The only special case that needs to be considered for this task is that the robot must have touched the target switch with the bar in the middle of the task. If the bar is placed on the support surface but did not touch the switch then the task is considered a failure. Figure 8.3 shows a snapshot of the robot performing the task.

During the execution phase the robot is in charge of maintaining the visual memory, i.e. it has to fixate the different landmarks in order to estimate their locations by means of the gaze control strategies described in Section 4.5. As mentioned above, only by reducing the location uncertainty of the tip of the bar, the obstacle and the target, the robot will be able to safely avoid the obstacle and touch the target switch, respectively.

8.3 Results

We begin this section by presenting the results obtained by Johansson et al. Then we present and compare the results from our gaze control models from a qualitative and quantitative perspective.

8.3.1 Johansson's Results

Figure 8.4A shows the results obtained by Johansson et al. [1][3]. The data was collected from 10 human subjects, each performing 4 trials. The graph presents the probability of fixating on a given landmark during each action phase (the colours correspond to the coloured areas shown in Figure 8.1). The horizontal line represents time, and the length of each action phase represents its median duration, calculated from all the trials. Subjects performed a median of 16 fixations per trial, and the median total duration of the task was 7.8 seconds.

8.3.2 Qualitative Comparison

The same procedure as explained by Johansson et al. [1] was followed in order to define the duration of the action phases. A total of 20 trials were performed for each of our gaze control strategies, where the reach/grasp sensitivity was set to 1.0 cm, the observation noise to 1.0, and the FoV to 60°x40°. This means that we used an unmodified version of the observation model (Section 4.4.1.1), and the FoV that is the default in the simulator.

Figure 8.4 compares the results of the human data (A) provided by Johansson et al. along with our three candidate gaze strategies (B-D), and *Random* and *Round Robin* schemes (E, F). The first thing to notice is how the *Rew+Unc+Gain* (B) and the *Rew+Unc* (C) gaze strategies produce the same relative ordering of gaze and manipulation actions as the human data (A). A median of 8 and 8.5 fixations per trial were performed by the *Rew+Unc+Gain* and the *Rew+Unc* schemes respectively, with a median duration of the task of 42.7 and 42.5 seconds respectively. The second thing to notice is that the rest of the gaze control strategies do not produce the same pattern as the human data. Table 8.3 presents the median fixations per trial and the median duration of the task for all gaze strategies and the human data.

[3]This graph is taken from [8] where the experiment was revisited. We use this graph as it summarises all the results in a single plot, as opposed to the ones presented in the original paper [1].

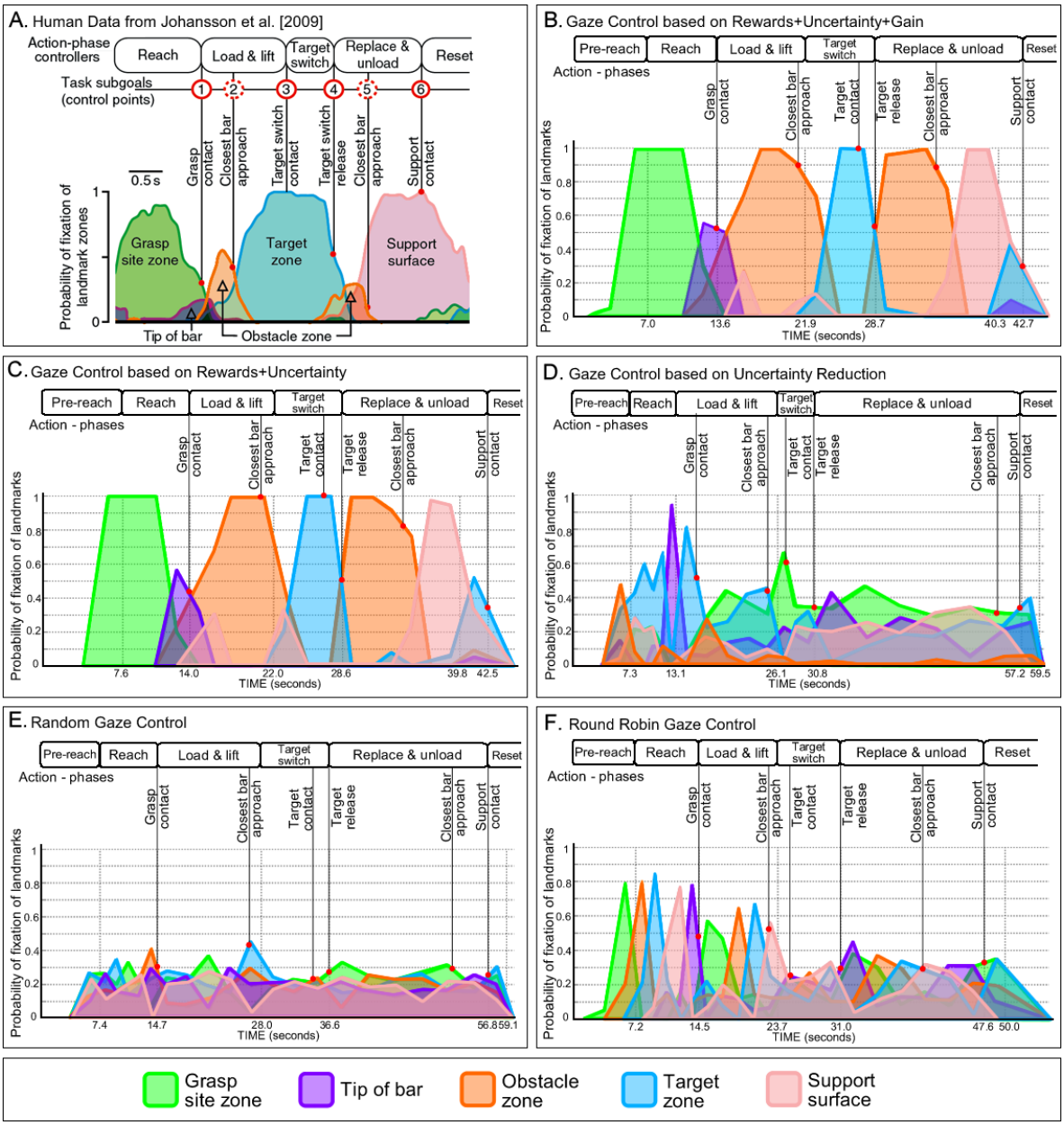

Figure 8.4: A. Behavioural human data as presented in [8] (Figure courtesy R. Johansson). B. Behavioural data obtained by the *Rew+Unc+Gain* gaze strategy. C. Data from the *Rew+Unc* gaze strategy. D. Data from the *Uncertainty* gaze strategy. E. Data from the *Random* gaze strategy. F. Data from the *Round Robin* gaze strategy. Action phases in the human data are divided in bins of 100 msec, compared to bins of 2-3 sec in our graphs (See text for details)

Table 8.3: Statistics for Johansson's task

Gaze Strategy	*Median fixations/trial*	*Median task duration (secs)*
Human Gaze	16	7.8
Rew+Unc+Gain	8	42.7
Rew+Unc	8.5	42.5
Uncertainty	22	59.5
Random	24	59.1
Round Robin	23.5	50

8.3.3 Quantitative Comparison

In order to quantitatively compare our candidate gaze strategies with the human data we make use of the *Levenshtein* distance (also known as *edit* distance) [261]. The Levenshtein distance measures how different two string sequences are by determining the number of edit operations that need to be employed to transform one string ($st1$) into the other ($st2$). These edit operations are: insertion, deletion or substitution of a single character. Thus, for each pair of characters belonging to each string ($st1_i, st2_j$) the Levenshtein distance is obtained as:

$$Lev(i,j) = min \begin{cases} Lev(i,j-1)+1 \\ Lev(i-1,j)+1 \\ Lev(i-1,j-1)+ \begin{cases} 0 & if(i=j) \\ 1 & otherwise \end{cases} \end{cases} \tag{8.1}$$

Here the Levenshtein distance is used to measure the differences between fixation sequences. According to the human data (Figure 8.4A), the most likely fixation sequence that a subject may employ whilst performing the task is, to first look at the grasp site, then at the tip of the bar, then the obstacle, the target, once again the obstacle, and finally the support surface. This sequence is given by the peaks of each one of the landmark's probability distributions, as shown in the human data graph (Figure 8.4A). By using a letter to represent each landmark, the human fixation sequence can be encoded into the string: *"gbotos"* (e.i. grasp site (g), bar tip (b), obstacle (o), target switch (t), support surface (s)). Based on this string, the Levenshtein distance of each trial is calculated, and the results for each gaze strategy are averaged.

Table 8.4 shows the average Levenshtein distance for all the gaze strategies. The values closer to zero represent a closer similarity to the original sequence, which is the case for *Rew+Unc+Gain* and *Rew+Unc* gaze strategies. Figure 8.5 presents the same averages graphically, where the error bars represent the 95% confidence intervals. A one-way ANOVA test was used with the gaze strategies as a factor. The differences are statistically significant at $p < .0001$.

Table 8.4: The averaged Levenshtein distance for all gaze control models.

Gaze Strategy	*Levenshtein distance*
Rew+Unc+Gain	1.55
Rew+Unc	1.85
Uncertainty	18.45
Random	24.45
Round Robin	17.3

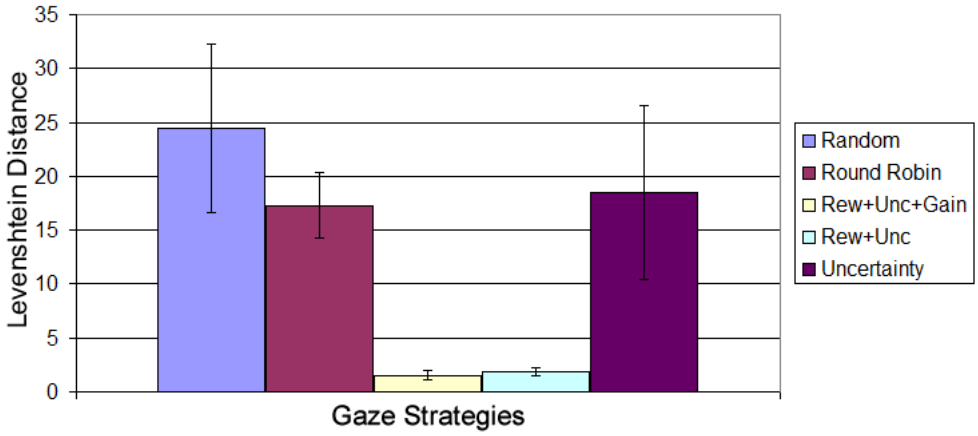

Figure 8.5: The averaged Levenshtein distance for all gaze control strategies. The error bars represent the 95% confidence intervals.

8.4 Discussion

The *Rew+Unc+Gain* and the *Rew+Unc* gaze strategies (Figure 8.4B and C) produce the same gaze pattern because for this particular task there are no differences between them. As explained in Section 4.5.3, the only difference between these two schemes is the way in which the gaze allocation problem is dealt with. As only one arm is used in this task the *gaze allocation* problem does not exist. Thus, both strategies lead to the same behaviour.

The *Rew+Unc+Gain* gaze strategy matches the same relative ordering of gaze and manipulation actions as the human behavioural data. For example, notice how the *grasping contact* event occurs after the grasping site has been fixated as in the human data, with the same happening for the rest of the events. In the case of the resulting statistical data, the robot performed a median of 8 fixations per trial, compared to 16 made by humans; and the median total duration of the task was 42.6 seconds for the robot, compared to 7.8 seconds in humans (Table 8.3). As is expected, the robot is

slower in terms of saccade execution and processing time. The robot takes on average 2 seconds to saccade, 0.65 seconds in deciding where to look, and depending on the number of objects in the visual memory it can take from 0.01 to 0.07 seconds to update the particle filters. This is in contrast to saccades made by humans that can take as little as a few milliseconds.

The shape of the probabilities are different in both the *Rew+Unc+Gain* and the human data graphs (see Figure 8.4A and B). In the case of the human data, every fixation was assigned to a given landmark if the fixation landed close to that landmark. However, the classification of some fixations was difficult to determine if they landed between two landmarks or far from all of them. In our case, it is possible to know exactly where the robot has decided to look. This is one of the reasons why our graphs present a high fixation probability for some landmarks. For example, compare the probability of fixating the obstacle by humans and the robot. The probability that humans fixate the obstacle as the bar moves towards the target switch is just above 50%, whereas the robot fixates the obstacle with a probability of 100%. Another reason is the effect of peripheral vision in humans, Johansson et al. [1] reported that the task could be achieved with no saccadic eye movements involved, although subjects reported that it was more difficult. Nevertheless, this suggests that in some cases the landmarks do not need to be foveated in order to accomplish the task.

Another notable difference between the human data and the rest of the graphs is that the probabilities from the human data are more curved compared to our graphs. The reason for this is the interpolation of the data points. Human subjects performed more fixations per trial and also completed the task faster than the robot. So in order to calculate the time of each action phase, the human data was divided into bins of 100 msec, whereas for our graphs the data was divided into bins of 2 to 3 seconds.

Recall that the behaviour of the *Uncertainty* gaze strategy is to fixate landmarks where information uncertainty can be reduced (Section 4.5.1). By following this strategy the robot tends to look at objects that are located far from each other. This is because whilst the robot fixates on one object the uncertainty of the other, unseen objects,

increases. This situation occurs during the *Reach* action phase (Figure 8.4D), where the target zone is fixated and then gaze moves to the tip of the bar, which are on extreme sides of the work space. Also, notice that the probability of fixating the obstacle is small. This is because the obstacle is in the centre of the work space, constantly appearing in the field of view, and in consequence getting updated in almost every fixation.

In the case of the *Random* gaze strategy, the likelihood of fixating some landmark has almost uniform probability (Figure 8.4E). Finally, the *Round Robin* gaze strategy shows a clear fixation pattern at the beginning of the task. However, by not getting support from gaze the action phases fail at different times. Thus, the gaze pattern gets distorted as the task progresses.

8.5 Conclusions

The main goal of this chapter was to determine the goodness of fitness of our candidate models of gaze control to human behavioural data. This was achieved by simulating the psychophysical task devised by Johansson et al. [1].

Our qualitative and quantitative results have demonstrated, that both the *Rew+Unc+Gain* and the *Rew+Unc* gaze schemes reproduced the same relative ordering of gaze and manipulation actions as the human data. However, it was pointed out that because only one arm is employed in the task, the behaviour of the *Rew+Unc+Gain* and the *Rew+Unc* gaze strategies is actually the same.

The results obtained by the *Uncertainty* gaze strategy reinforce our previous conclusion (Section 5.5) that ignoring task rewards for the control of gaze produces an inefficient behaviour.

It is, of course, not possible to conclude that the *Rew+Unc+Gain* gaze control model fully explains the human data. First, further experiments should be done where more motor systems are employed in order to differentiate between the *Rew+Unc+Gain* and the *Rew+Unc* gaze schemes. This can be done for example by: i) simulating psychophysical tasks which involve two arms [107,108,113], or ii) possible experiments with

varying rewards, where high reward landmarks have low positional uncertainty ($HRLU$) and low reward landmarks high uncertainty ($LRHU$); in this case *Rew+Unc* is expected to favour the $HRLU$ landmarks and *Rew+Unc+Gain* the $LRHU$ landmarks. Second, detailed experimental tasks should be devised in order to isolate and analyse the role of uncertainty information and task rewards in humans (e.g. [11, 26, 36, 130]). Nevertheless, our results suggest that gaze control models based on reward and uncertainty are candidate models that are worthy of further analysis.

Chapter 9

Conclusions and Future Work

This thesis addressed two questions: i) what mechanisms a rational decision maker could employ to select a gaze location given limited information and limited computation time during the performance of a task; and ii) how humans might select the next gaze location. Previous work has suggested that human eye movement behaviour is consistent with decision making mechanisms for fixation selection that are Bayes' rational [11, 12], or that try to maximise reward [13, 14]. This thesis investigates these claims further by means of two goals:

1. **Engineering science goal:** Our primary goal was to formulate and characterise principled methods for gaze control that deal with the problem of *fixation selection* during the performance of manipulation tasks. Three one-step look ahead models of gaze control were formulated. Our first model chooses the fixation location that maximises the reduction of location uncertainty (*Uncertainty* scheme). Our second model incorporates task rewards and selects the fixation that maximises the value of performing an action by reducing location uncertainty (*Rew+Unc* scheme). Our third model is similar to *Rew+Unc* but it maximises the gain that results when a manipulation motor system is given access to perception (*Rew+Unc+Gain* scheme). These models were implemented on a simulated humanoid robot, and were characterised using two variations of a pick & place task.

2. **Human behavioural goal:** Our secondary goal was to compare our models of

gaze control to human data to determine whether or not our models could provide some insights into how humans select fixation locations. This was done through the implementation of a third manipulation task based on Johansson's task [1].

9.1 Main Results

Throughout the thesis our three models of gaze control were characterised and compared in terms of task performance in three manipulation tasks. In order to have a common baseline for comparison, two more gaze strategies were implemented: *Random* and *Round Robin*. This section presents our main results[1]:

1. Based on the models of gaze control presented in this thesis, perhaps our main conclusion is that, models that incorporate both rewards and information uncertainty perform better in manipulation tasks, compared to models that employ information uncertainty only or that are based on heuristics (e.g. *Random* and *Round Robin*). This, of course, does not rule out the possibility that other models of gaze control may provide better performance. Still, it is clear from our experiments that reasoning about rewards and information uncertainty provides an elegant and efficient method for the selection of fixation locations.

2. Our *Rew+Unc+Gain* gaze strategy has, in general, the best overall task performance in terms of sensitivity in the manipulation actions, observation noise and changes to the field of view (FoV). The difference between this model and *Rew+Unc* is the way in which gaze is *allocated* to a manipulation motor system. Thus, this suggests that allocating gaze by calculating the *gain* that each manipulation motor system may receive is an efficient solution.

3. On the other hand, the performance of the *Uncertainty* gaze strategy is, in general, similar to that of *Random* and *Round Robin*. Reasoning only about information uncertainty might be good for purely information gathering tasks (e.g. visual

[1]For more specific discussion please refer to the conclusions section in Chapters 5, 6, 7 and 8.

search). However, when the agent/robot is expected to perform a physical task (e.g. manipulation tasks), controlling gaze based on information uncertainty only is, in most cases, sub-optimal.

4. As explained in Section 3.2.3.3, our gaze control models build from the work of Sprague and Ballard [7, 43]. However, there are key differences with respect to our models: i) we deal with the problems of *where to look* and *gaze allocation* together (Section 1.3), ii) we consider a separate oculomotor system to direct gaze to specific locations, iii) we consider multiple manipulation motor systems, iv) we consider manipulation actions with variable duration, v) our gaze control models reason about the remaining uncertainty that would result after a fixation location is selected, as opposed to Sprague's model that assumes that all uncertainty disappears after a fixation is made, vi) we incorporate an active visual search process into the gaze control mechanism. Because of these differences it is not possible to compare directly our models to Sprague's model.

5. One of the main conclusions from the characterisation of the models (Chapter 5) was that an efficient *visual search* process should be considered in order to maintain the performance and robustness of the *Rew+Unc+Gain* gaze scheme to changes in the FoV. Most works have only dealt with the visual search problem in isolation. Here we argue that, in many circumstances, visual search should be carried out at the same time as other task(s) are being performed by the agent. This situation brings up the problem of how to integrate visual search into the gaze control process.

6. The active visual search approach we formulated in this thesis helped the robot to increase its performance and robustness to changes in the FoV (Chapter 6). We could argue that the way in which the agent decides whether or not to execute a visual search action (i.e. by calculating the visual search gain) might generalise to other kinds of tasks. Nevertheless, the search strategy was formulated specifically to the pick & place task, and in consequence it is not expected to generalise to

other kinds of tasks.

7. Our models were also tested in a bimanual task, that extends the pick & place task by forcing the robot to transfer objects from one hand to the other, as one of the containers was removed (Chapter 7). Bimanual actions are interesting from the point of view of manipulation motor control because a coordination mechanism should be implemented. In this thesis, however, our interest was mainly in the behaviour of the gaze controller. Typically, bimanual actions would make the robot's arms share at least one fixation point. In terms of gaze control, that the motor systems share fixation points is convenient because, in some instances, the gaze allocation problem might even disappear.

8. The relative ordering of gaze and manipulation actions produced by existing behavioural human data [1] was compared against the behaviour produced by our gaze control models. Only the *Rew+Unc+Gain* and *Rew+Unc* gaze schemes were able to match the human data. This provides further evidence that rewards and uncertainty allow for better gaze control mechanisms. Of course, how humans select the next fixation location is still an open question that requires further research and interdisciplinary collaboration.

9.2 Future Work

Throughout the presented research work we had to make several assumptions and simplifications in order to focus on the gaze control problem. Therefore, there are many ways in which the current work can be extended. This section discusses several of these possible extensions organised as follows: i) the formulation of new models of gaze control, ii) the implementation of our proposed models in a real humanoid robot, iii) the implementation of visual processing algorithms into the current system, iv) the implementation of other tasks, v) the design of psychophysical experiments to understand the importance of rewards and uncertainty, and vi) the discussion of further extensions.

9.2.1 New Models of Gaze Control

One of the most important things to be done is to formulate additional models of gaze control for comparison with the ones presented in this thesis.

9.2.1.1 Models Based on Current Information

This thesis focused only on one-step lookahead gaze control models. This allowed us to formulate the three present models based on rewards and for uncertainty. One clear extension is to formulate models that use only the *current* or *present* information, rather than a one-step lookahead prediction. It is possible to identify three such models of gaze control:

- **Gaze control based on current uncertainty:** Instead of predicting the reduction in uncertainty one-step ahead (as our *uncertainty* gaze scheme), this model could select the next fixation location using the current location uncertainty about landmarks. The model would simply choose to fixate the landmark with the current maximum uncertainty in order to minimise it.

- **Gaze control based on reward:** This is a very interesting model since uncertainty is not taken into account. The model would choose to fixate the most rewarding landmark every time step. This can be done by using the learnt action-value function (i.e. the *Q-values*). For some tasks, such as the pick & place task, by disregarding uncertainty several objects may have the same expected return and the model might have to select one randomly.

- **Gaze control based on reward and current uncertainty:** This model reasons about rewards and the current uncertainty together. For example, if an object has high location uncertainty then its value will be small (as explained in Section 1.5). On the other hand, if the location uncertainty is low then the value of the object will be closer to the one given by the current Q-value associated with that object. One possible issue with this model (that needs to be verified) is that, if the robot happens to fixate a non-relevant object (because the rest have high uncertainty

and their values are small), then it might remain fixating that object forever since the uncertainty of the others will never get reduced. If this issue happens it could be handled with heuristics.

Characterising and comparing these models using the same experiments presented in this thesis would allow us to determine whether or not one-step lookahead models are better than models using current information alone.

9.2.1.2 N-Step Lookahead Models

The next extension would be to formulate models able to make predictions more than one step into the future. The implementation of these models is not as straightforward as the previous ones. Here, it is necessary to predict the value of a sequence of manipulation actions. This is more complex and is likely to take longer periods of decision time.

For example, from a psychophysical perspective, experiments such as the tea-making task [109] reported that subjects performed fixations with the goal of localising objects for future use.

It is difficult to say whether the use of n-step lookahead models using the manipulation tasks presented in this thesis would result in higher performance. The workspace of our manipulation tasks is too small and the number of landmarks that the robot keeps in visual memory is not too large. Therefore, it is likely that the benefit of making predictions n-steps into the future will be small or nonexisting. Other kinds of tasks would have to be devised in order to test this benefit.

9.2.1.3 Saliency Based Models

All the previous models are still based on rewards and for uncertainty. However, it is also important to compare to models that have a different way of selecting fixation locations. For instance, models that decide where to look based on bottom-up saliency [145]. This kind of model does not use task information so it it could work as a common baseline (as *Random* and *Round Robin* do) [262].

9.2.1.4 Multiple Sources of Uncertainty

Formulating new models of gaze control is important for testing and comparison. However, another significant extension is within the models themselves. Our proposed models employed only a single source of uncertainty, namely the locations of objects. However, in real-world tasks there are multiple sources of uncertainty associated with the objects, and the robot should be able to deal with them as well (e.g. the object's size, colour, shape, category, etc.).

The gaze control models should be able to integrate these sources of uncertainty, and then use such information in order to decide the next fixation location. This may involve the creation of different observation models for each object property. By handling multiple sources of uncertainty, the gaze models could be employed for a larger number of tasks.

9.2.1.5 Reward Strategy

As explained in Section 4.3, our pick & place task assigns the same reward to all objects. Further experiments can be performed with variations of the reward strategy. For instance, red objects could be assigned higher reward. This goes hand-in-hand with the types of tasks we would like the robot to perform. For example, a sorting task where objects are grasped according to their shape or colour. These experiments could help us to further differentiate between our *Rew+Unc+Gain* and *Rew+Unc* schemes.

9.2.1.6 Active Visual Search

As mentioned in Chapter 6, there can be further extensions to our current active visual search process. First, the search strategy, defined in Section 6.3.2, could be more efficient. Instead of resampling the particles every time a visual search action is executed, it would be more efficient to select the view point that contains more particles. If an object is not found in that view point, then those particles are deleted, so that the next view point is again the one that contains the most particles. Once an object is found the particles can be redistributed again across the table. Second, the number of objects

to search for could vary, rather than be restricted to just one object. To do this, the robot would need to reason about the gain to be obtained by knowing about two or more objects.

9.2.2 Real Robot Implementation

In order to fully characterise and test our proposed models of gaze control, they should be implemented in a real humanoid robot. To accomplish this, several issues need to be taken into account:

- **The robot:** Since our current system has been implemented using the iCub simulator, a logical step is to use a real iCub robot [54] to test our models. Even though the simulator employs the same controllers as for the real robot, there are still several things that need to be taken care of and which are discussed below. Another possibility is to port our current system to ROS (Robot Operating System) [263], which offers a generic communication platform that would allow us to test our models in different humanoid robots (e.g. Nao [264]).

- **Oculomotor System:** As explained in Section 1.2, we have assumed that the robot was already able to control the oculomotor system. In fact, our focus was just on saccadic eye movements and only one camera was employed. However, an interesting extension could be to incorporate into our models the possibility to perform smooth pursuit and vergence. In the case of vergence, we should extend our system to consider binocular information [38].

- **Observation model:** It is not clear whether the observation model, defined in Section 4.4.1.1 could also be used for the real robot. We suspect that, in fact, the noise after triangulation is even higher in the real robot. Therefore, it is likely that another observation model should be learnt.

- **Environment:** One of the best advantages of the simulator is the possibility of manipulating the environment with ease. For instance, objects can appear during

the execution of the simulation, and the precise coordinates of all objects are available for testing purposes or to implement safety mechanisms. In the real environment this is more difficult to achieve, thus we need to come up with ways in which the same manipulation tasks can be performed by the real robot.

- **Grasping:** One of the biggest simplifications in this thesis had to do with the problem of grasping. As explained in Section 4.1, the robot simulation employs a magnet-like function for grasping objects. Further work needs to be done so that both the simulated and the real robot are able to grasp objects [265].

- **3D Coordinates:** Another simplification was that the locations of landmarks were represented only by their 2D coordinates. This was mainly because the table's plane is known, so one coordinate is given to the robot. Nevertheless, the robot should be able to reason about locations in 3D space [266]. This extension is likely to affect the observation model and how grasping is performed.

9.2.3 Visual Processing

We pointed out in Section 4.4.1.1 that the simulator was directly used to detect landmarks lying within the field of view. Therefore, no image processing was employed in this thesis. Another logical extension is to integrate image processing techniques into our current system.

- **Object detection and recognition:** At the moment, since we only keep track of the location of objects, an object detection mechanism might be sufficient to implement in our system. However, later on the robot might be required to keep track of other object properties (e.g. size, shape, type, etc.), in this case object recognition mechanisms should also be implemented [267]. The two main issues that these algorithms would have to deal with are, real-time execution and occlusions.

  In addition, we may also require an object identification process. This process is straightforward in the simulator, however this process should also be integrated if

the system is to be implemented in the real robot.

No image processing technique is perfect, therefore the models of gaze control would have to deal with the uncertainty generated by these visual mechanisms. These are other sources of uncertainty besides those already mentioned above.

- **Bottom-up saliency:** Some of the work reviewed in Chapter 3 combined bottom-up and top-down information for the selection of fixation locations. We could argue that if the agent has specific tasks to perform then bottom-up information is not required. However, it can also be argued that, a robot expected to perform in long-term scenarios should include some sort of "curiosity" mechanism, and bottom-up saliency may act as the trigger for such mechanism [268].

9.2.4 Tasks

The kind of tasks that the robot is expected to perform constrain much of the topics that we have delineated above.

- **Task decomposition:** The formulation of our models of gaze control largely depends on the way the task is decomposed and represented. This thesis followed a two-level hierarchical task decomposition where the number of tasks was equal to the number of motor systems. More levels in the hierarchy could allow the robot to reason about other kinds of tasks, and different ways to organise its motor systems [16, 252]. In particular, it would be interesting to use the hierarchical decomposition proposed by Ghavamzadeh et al. [254]. Even though their decomposition is for multi-agent systems, it might be possible to be used for multiple motor systems. In addition, non-hierarchical representations could also be explored.

- **Selection of tasks:** In this thesis we assumed that the robot can only execute one task at a time. In long-term scenarios or for multi-purpose robots, this cannot be the case. The robot should be able to perform a variety of tasks and select the one more appropriate for the current situation [7].

- **Parallelism and concurrency:** Building from the previous two points, a robot should be expected to perform multiple tasks simultaneously in parallel or concurrently[2]. This is specially expected from those robots with several degrees of freedom (e.g. humanoid robots) [252] (or multi-agent systems [254]). The overall system, along with the gaze control mechanism, might need to be modified depending on what the robot is expected to do.

- **Types of tasks:** In this thesis we have focused only on manipulation tasks, but we could try to model other tasks such as a navigation task [7]. Nevertheless, there are several variations that can be performed from our current pick & place task, depending on some of the topics already covered. For example, sorting objects in different containers according to colour or size, obstacles could be placed on the table, containers could move, etc.

9.2.5 Human Behavioural Experiments

As reviewed in Section 3.1.2, some psychophysical studies have already provided evidence that rewards and uncertainty are used for the control of gaze. Still, further experiments could be devised in order to analyse the specific role of rewards and uncertainty in a similar way to the gaze strategies presented in this thesis during manipulation tasks.

A further extension related to this topic is the possible implementation of our gaze control models following neurocomputational algorithms [136, 171].

9.2.6 Further Extensions

Finally, at least two more extensions can be incorporated into the current system:

- **Information-gathering manipulation actions:** Typically, manipulation actions are assumed to be performed in order to achieve some task. However, some manipulation actions could be used to gather information about the environment.

[2]Both, parallelism and concurrency refer to processes that can be executed simultaneously. Parallel processes do not depend on each other, so no interaction is needed. In contrast, concurrent processes depend on each other for their correct execution, so normally a coordination mechanism is required to control the processes.

For instance, moving objects that occlude the robot's view, or moving the robot to different locations in order to get a better view of something [239]. These actions are not part of the task but they could be essential to the successful achievement of such task. Our system could be extended in order to allow this kind of action.

- **Multi-sensory integration:** The focus of this thesis and many other research projects is vision alone. However, the integration of multiple sensory information could provide much more information to the robot. For example, tactile and visual information could be combined in order to reduce uncertainty about the location, or other properties, of objects. It is likely that this integration would change the way in which gaze is deployed, since the robot does not need to rely on visual information alone [269].

9.3 Summary

This chapter presented a quick overview of the thesis, our conclusions, and future lines of work. The primary goal of the thesis was to formulate principled methods for the control of gaze. These models of gaze control were implemented in a simulated humanoid robot, and they were characterised in terms of task performance during the execution of three manipulation tasks. The aim of the gaze control models is to select fixation locations based on rewards and for uncertainty. Our results demonstrated that these kind of models provide an efficient way to decide where to look and how to allocate gaze. We also demonstrated that those models based on rewards and uncertainty match human behavioural data. Nevertheless, other ways in which location fixation can be selected cannot be ruled out and further research is needed.

Appendix A

Further Spatial Acuity Results

In Section 5.1 we presented an analysis that compared three different models of spatial acuity in the field of view (FoV). The three noise models were defined as:

- **Model 1 - Uniform spatial acuity:** After the particle filter update, the standard deviation of the cloud of particles is measured. Normally distributed noise is added to each particle with zero mean and standard deviation equal to the 10% of the cloud's standard deviation. Thus the particle filter adds noise that is modulated by the belief state, and noise can be the same across the visual field. This noise is used to escape the local minima.

- **Model 2 - Stepped spatial acuity:** The same noise as in Model 1 is added to the particle filter, to escape local minima. In addition, normally distributed noise (zero mean and 1.0 cm of standard deviation) is added to those landmarks inside the FoV but not in the centre of the camera (where we defined the centre to cover 10°). This models two regions: i) the "fovea" (0°to 10°) having higher spatial acuity, and ii) the "periphery" of the field of view (> 10°) having lower spatial acuity. However, note that the spatial acuity is *uniform* across the "periphery" of the FoV.

- **Model 3 - Smoothly varying spatial acuity:** Noise is added to each particle according to the bivariate Gaussian density that gives the observation model

(Section 4.4.1.1) for that particle's position in the visual field. By re-employing the observation model in this way we are able to model a smooth fall off in spatial acuity as we move from the centre of the field of view to the periphery.

Section 5.1 compared all three models in terms of variations of reach/grasp sensitivity. One of the main conclusions was that gaze control ceases to be important when the FoV has a uniform spatial acuity. Therefore, we decided to use a non-uniform model for all of our experiments. In this case we chose Model 3 as it offers a better way to differentiate between our proposed models of gaze control. Furthermore, it is qualitatively similar although not the same as inhomogeneity of the human eye. As discussed in Section 5.1, Model 2 and 3 produce very similar results in terms of the difference in task performance between the *Rew+Unc+Gain* gaze scheme and the two common baselines (*Random* and *Round Robin*). Therefore, this appendix presents further experiments that only compare Model 1 and 3 whilst varying the observation noise and the FoV.

A.1 Observation Noise

The observation model was scaled by the factors: 1.0, 1.5, 2.0 and 2.5. Figure A.1 presents the average number of objects correctly placed in the containers for the *Rew+Unc+Gain*, *Random* and *Round Robin* gaze strategies, where reach/grasp sensitivity is set to 1.0 cm and the FoV is 60°x40°. A total of 10 trials of 5 minutes each were performed for each gaze strategy. The error bars in the graph represent the 95% confidence intervals.

The results for Model 3 (Figure A.1B) are the same as those shown in Figure 5.5, but here only the results of the *Rew+Unc+Gain*, *Random* and *Round Robin* gaze schemes are presented. In this case the *Rew+Unc+Gain* gaze strategy outperforms the two baselines and the overall task performance of all schemes decreases as the observation noise accrues. In contrast, notice how for Model 1 the overall task performance remains practically the same (Figure A.1A). Increasing the observation noise has no effect when the FoV has uniform acuity. This further suggests that a single fixation obtains a

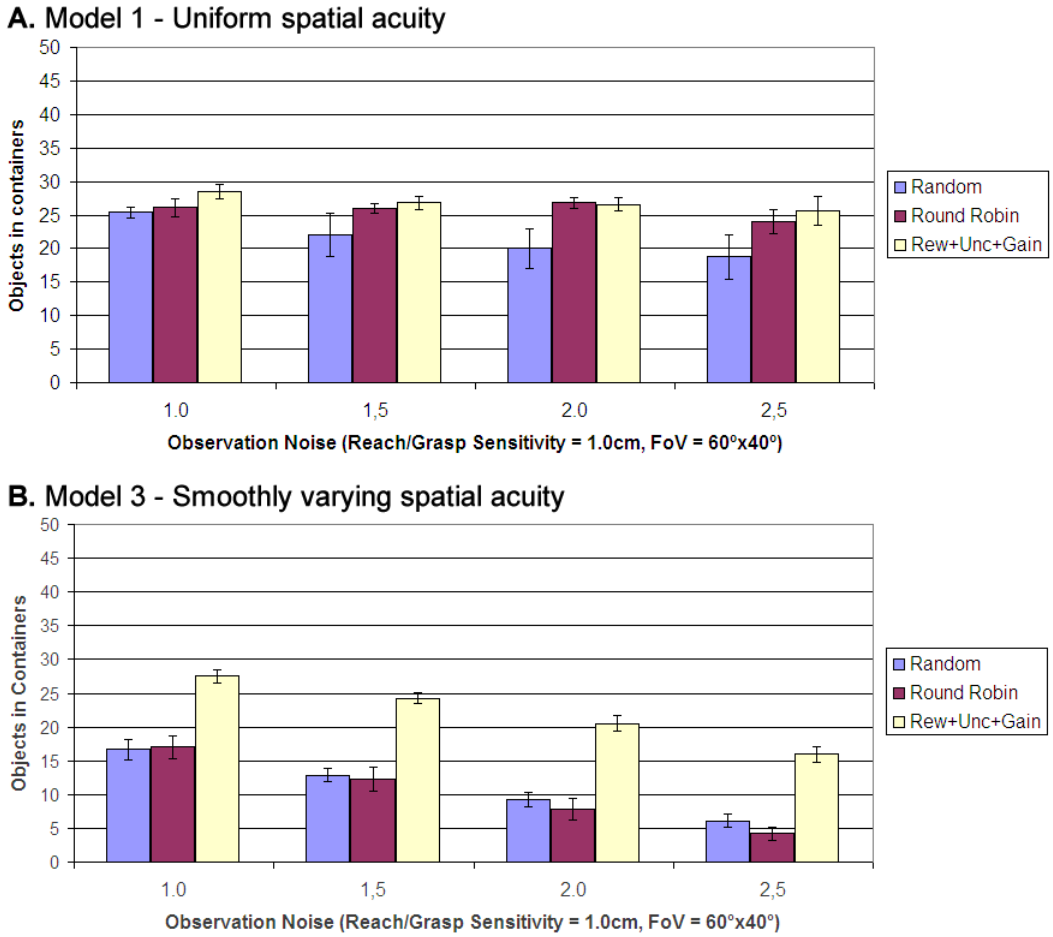

Figure A.1: Effect in task performance when the FoV has: A) Uniform spatial acuity (Model 1), B) Smoothly varying spatial acuity (Model 3); whilst the observation noise varies. Reach/grasp sensitivity = 1.0 cm and the field of view = 60°x40°. The error bars represent the 95% confidence intervals.

considerable amount of information that is not affected by extra noise in the observation model. Once again, the benefit of choosing fixation locations disappears in Model 1.

The two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain* gaze strategy against the two baselines for both noise models. Using Model 1, when compared to *Random* the differences are statistically significant for all observation noise factors at $p < .009$. However, when compared to *Round Robin* only for the first factor (1.0) the difference is statistically significant at $p < .013$. Following Model 3, for all factors the differences are statistically significant at $p < .0001$.

A.2 Field of View

For this experiment we vary the horizontal and vertical angles of the FoV with the values: (60°x40°), (50°x35°), (40°x30°), (35°x25°), and (25°x20°). Figure A.2 presents the average number of objects correctly placed in the containers for the *Rew+Unc+Gain*, *Random* and *Round Robin* gaze strategies, where the reach/grasp sensitivity is set to 1.0 cm and the observation noise is set to 1.0. A total of 10 trials of 5 minutes each were performed for each gaze strategy. The error bars in the graph represent the 95% confidence intervals.

The results for Model 3 (Figure A.2B) are the same as those shown in Figure 5.7, but here only the results of the *Rew+Unc+Gain*, *Random* and *Round Robin* gaze schemes are presented. In this case the *Rew+Unc+Gain* gaze strategy outperforms the two baselines and the overall task performance of all schemes decreases as the FoV narrows. This time, the overall performance when Model 1 is used also decreases as the size of the FoV gets smaller (Figure A.2A). This is to be expected since having a uniform spatial acuity does not eliminates the effect of a decreasing FoV. The benefit of a gaze control strategy is slightly better as the size of the FoV determines the amount of information to be captured. Nevertheless, the benefit is not as clear as with a non-uniform spatial acuity.

The two-tailed unpaired t-test was used to compare the results of the *Rew+Unc+Gain* gaze strategy against the two baselines for both noise models. Using Model 1, when compared to *Random* the differences are statistically significant for all FoV values at $p < .01$, except when the FoV is (25°x20°). When compared to *Round Robin* only for the first two FoV values the difference is statistically significant at $p < .02$.

Following Model 3, for all FoV values the differences are statistically significant at $p < .0001$, except when the FoV is (25°x20°) where the difference against *Round Robin* is not statistically significant.

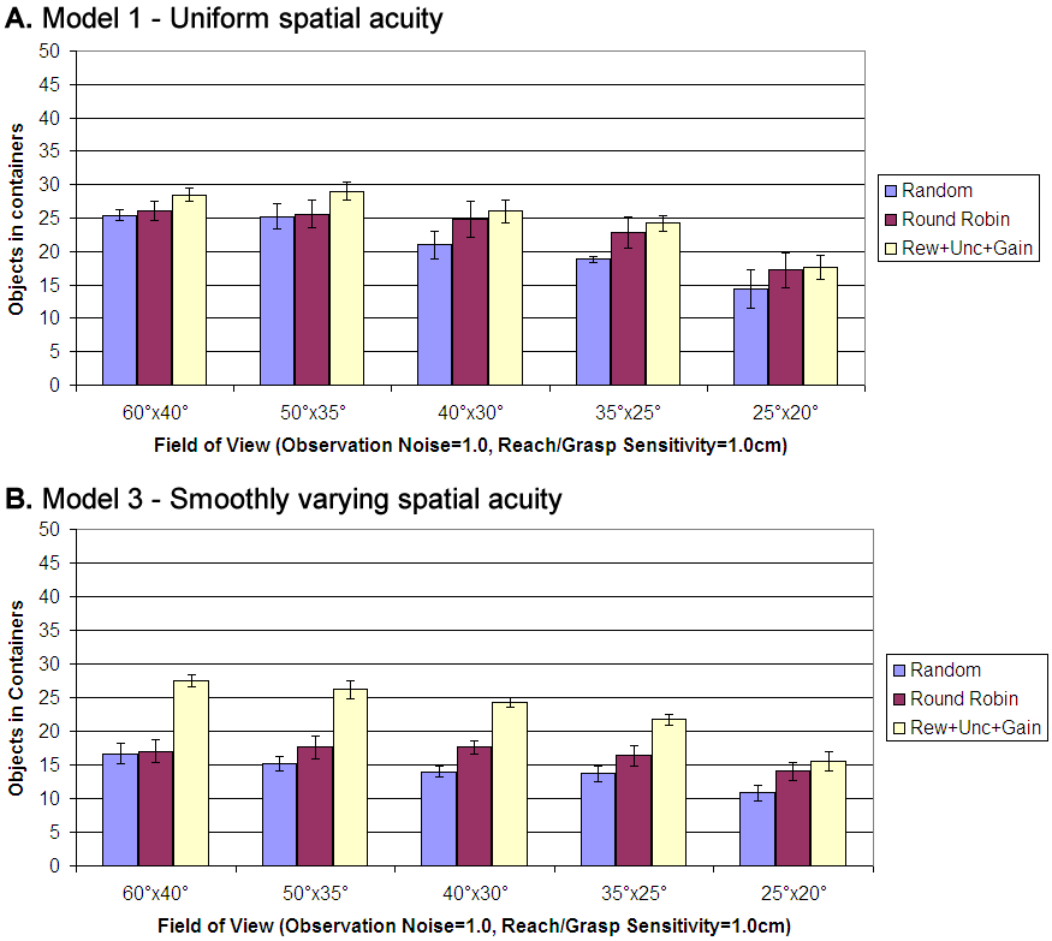

Figure A.2: Effect in task performance when the FoV has: A) Uniform spatial acuity (Model 1), B) Smoothly varying spatial acuity (Model 3); whilst the field of view varies. The observation noise = 1.0 and the field of view = 60°x40°. The error bars represent the 95% confidence intervals.

A.3 Conclusions

These experiments provide further evidence that a gaze control strategy for the selection of fixation locations ceases to be important when the field of view has a uniform spatial acuity. Therefore, in order to test our models of gaze control and the importance of fixation task-relevant landmarks (i.e. moving the robot's cameras so that landmarks appear in the camera's centre), we need to consider a field of view with non-uniform spatial acuity.

Bibliography

[1] R. Johansson, G. Westling, A. Backstrom, and J. R. Flanagan, "Eye-hand coordination in object manipulation," *Journal of Neuroscience*, vol. 21, no. 17, pp. 6917–6932, 2001.

[2] N. E. Institute, "Diagram of the eye." `http://www.nei.nih.gov/health/eyediagram/eyeimages3.asp`, 2012.

[3] M. Land, "Eye movements and the control of actions in everyday life," *Prog. in Retinal and Eye Research*, vol. 25, no. 3, pp. 296–324, 2006.

[4] B. Tatler, N. Wade, H. Kwan, J. Findlay, and B. Velichkovsky, "Yarbus, eye movements, and vision," *i-Perception*, vol. 1, no. 1, pp. 7–27, 2010.

[5] A. Yarbus, *Eye movements and vision*. New York: Plenum Press, 1967.

[6] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in Cognitive Sciences*, vol. 9, no. 4, pp. 188–194, 2005.

[7] N. Sprague, *Learning to coordinate visual behaviors*. PhD thesis, The University of Rochester, New York, May 2004.

[8] R. S. Johansson and J. R. Flanagan, "Sensorimotor control of manipulation," *Encyclopedia of Neuroscience*, vol. 8, pp. 593–604, 2009.

[9] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[10] J. Findlay and I. Gilchrist, *Active Vision*. Oxford: Oxford Univ. Press, 2003.

[11] J. Najemnik and W. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, no. 7031, pp. 387–391, 2005.

[12] A. Yuille and D. Kersten, "Vision as Bayesian inference: Analysis by synthesis?," *Trends in cognitive sciences*, vol. 10, no. 7, pp. 301–308, 2006.

[13] V. Navalpakkam, C. Koch, A. Rangel, and P. Perona, "Optimal reward harvesting in complex perceptual environments," *Proceedings of the National Academy of Sciences*, vol. 107, no. 11, pp. 5232–5237, 2010.

[14] A. Schutz and K. Gegenfurtner, "Dynamic integration of saliency and reward information for saccadic eye movements," *Journal of Vision*, vol. 10, no. 7, pp. 551–551, 2010.

[15] E. Kowler, "Eye movements: The past 25 years," *Vision research*, vol. 51, no. 13, pp. 1457–1483, 2011.

[16] M. Hayhoe and C. Rothkopf, "Vision in the natural world," *Wiley Inter. Reviews: Cognitive Science*, vol. 2, no. 2, pp. 158–166, 2010.

[17] A. Schütz, D. Braun, and K. Gegenfurtner, "Eye movements and perception: A selective review," *Journal of Vision*, vol. 11, no. 5, pp. 1–30, 2011.

[18] M. Land, "Vision, eye movements, and natural behavior," *Visual neuroscience*, vol. 26, no. 1, pp. 51–62, 2009.

[19] D. Ballard, "Animate vision," tech. rep., University of Rochester, 1990.

[20] C. Rothkopf, D. Ballard, and M. Hayhoe, "Task and context determine where you look," *Journal of Vision*, vol. 7, no. 14, pp. 1–20, 2007.

[21] D. Ballard, M. Hayhoe, F. Li, and S. Whitehead, "Hand-eye coordination during sequential tasks [and discussion]," *Phil. Trans. Royal Society of London. Series B: Biological Sciences*, vol. 337, no. 1281, pp. 331–339, 1992.

[22] M. Land and D. Lee, "Where we look when we steer," *Nature*, vol. 369, no. 6483, pp. 742–744, 1994.

[23] J. Triesch, D. Ballard, M. Hayhoe, and B. Sullivan, "What you see is what you need," *Journal of Vision*, vol. 3, no. 1, pp. 86–94, 2003.

[24] C. Rothkopf and D. Ballard, "Image statistics at the point of gaze during human navigation," *Visual neuroscience*, vol. 26, no. 1, pp. 81–92, 2009.

[25] R. Steinman, Z. Pizlo, T. Forofonova, and J. Epelboim, "One fixates accurately in order to see clearly not because one sees clearly," *Spatial Vision*, vol. 16, no. 3-4, pp. 225–241, 2003.

[26] B. Sullivan, L. Johnson, C. Rothkopf, D. Ballard, and M. Hayhoe, "The effect of uncertainty and reward on fixation behavior in a driving task," *Journal of Vision*, vol. 12, no. 9, pp. 1259–1259, 2012.

[27] T. Prescott, K. Gurney, and P. Redgrave, "Basal ganglia," in *The Handbook of Brain Theory and Neural Networks* (M. Arbib, ed.), p. 147151, MIT Press, 2002.

[28] P. Glimcher, "The neurobiology of visual-saccadic decision making," *Annual review of neuroscience*, vol. 26, no. 1, pp. 133–179, 2003.

[29] O. Hikosaka, "Basal ganglia mechanisms of reward-oriented eye movement," *Annals of the New York Academy of Sciences*, vol. 1104, no. 1, pp. 229–249, 2007.

[30] W. Schultz, "Multiple reward signals in the brain," *Nat. Reviews Neuroscience*, vol. 1, no. 3, pp. 199–207, 2000.

[31] A. Yu and P. Dayan, "Uncertainty, neuromodulation, and attention," *Neuron*, vol. 46, no. 4, pp. 681–692, 2005.

[32] K. Doya, "Metalearning and neuromodulation," *Neural Networks*, vol. 15, no. 4, pp. 495–506, 2002.

[33] P. Dayan and A. Yu, "Ach, uncertainty, and cortical inference," in *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, (Vancouver,Canada), pp. 189–197, 2002.

[34] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA: MIT Press Cambridge, 1998.

[35] C. Karaoguz, T. Rodemann, and B. Wrede, "Optimisation of gaze movements for multitasking using rewards," in *Proceedings of the 2011 IEEE/RSJ IROS*, (USA), pp. 1187–1193, Oct. 2011.

[36] L. Renninger, P. Verghese, and J. Coughlan, "Where to look next? Eye movements reduce local uncertainty," *Journal of Vision*, vol. 7, no. 3, pp. 1–17, 2010.

[37] R. Snowden, P. Thompson, and T. Troscianko, *Basic vision: An introduction to visual perception*. Oxford: Oxford Univ. Press, 2006.

[38] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal, "Biomimetic oculomotor control," *Adaptive Behavior*, vol. 9, no. 3-4, pp. 189–207, 2001.

[39] U. Pattacini, *Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub*. PhD thesis, Istituto Italiano di Tecnologia, Genova, Genova,Italy, 2010.

[40] S. Ullman, "Visual routines," *Cognition*, vol. 18, no. 1-3, pp. 97–159, 1984.

[41] J. Chambers, K. Gurney, M. Humphries, and T. Prescott, "Mechanisms of choice in the primate brain: A quick look at positive feedback," in *Modelling Natural Action Selection* (T. J. P. J. J. Bryson and A. Seth, eds.), pp. 45–52, AISB Press, 2005.

[42] P. Roelfsema, "Elemental operations in vision," *Trends in cognitive sciences*, vol. 9, no. 5, pp. 226–233, 2005.

[43] N. Sprague, D. Ballard, and A. Robinson, "Modeling embodied visual behaviors," *ACM Transactions on Applied Perception*, vol. 4, no. 2, pp. 1–23, 2007.

[44] M. Sridharan, R. Dearden, and J. Wyatt, "HiPPo: Hierarchical POMDPs for planning information processing and sensing actions on a robot," in *International Conference on Automated Planning and Scheduling (ICAPS)*, (Sydney, Australia), pp. 346–354, Sept. 2008.

[45] T. Erez, J. Tramper, W. Smart, and S. Gielen, "A POMDP model of eye-hand coordination," in *Proceedings of the 25th Conference on Artificial Intelligence*, (California), pp. 952–957, Aug. 2011.

[46] J. Vogel and K. Murphy, "A non-myopic approach to visual search," in *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*, (Montreal, Quebec, Canada), pp. 227–234, May 2007.

[47] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*. New York, NY: Springer-Verlag, 2006.

[48] A. Guillot and J. Meyer, "The animat contribution to cognitive systems research," *Cognitive Systems Research*, vol. 2, no. 2, pp. 157–165, 2001.

[49] T. Prescott, F. M. González, K. Gurney, M. Humphries, and P. Redgrave, "A robot model of the basal ganglia: Behavior and intrinsic processing," *Neural Networks*, vol. 19, no. 1, pp. 31–61, 2006.

[50] B. Webb, "Using robots to understand animal behavior," *Advances in the Study of Behavior*, vol. 38, pp. 1–58, 2008.

[51] B. Webb, "Animals versus animats: Or why not model the real iguana?," *Adaptive Behavior*, vol. 17, no. 4, pp. 269–286, 2009.

[52] T. Ziemke, "On the role of robot simulations in embodied cognitive science," *AISB journal*, vol. 1, no. 4, pp. 389–399, 2003.

[53] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: The prototype of the iCub humanoid robot simulator," in *Proc. 8th Workshop on Performance Metrics for Intelligent Systems*, (Maryland), pp. 57–61, Aug. 2008.

[54] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: An open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*, (MD, USA), pp. 50–56, Aug. 2008.

[55] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Taipei, Taiwan), pp. 1668–1674, Oct. 2010.

[56] R. Smith, "Open dynamics engine." `http://www.ode.org`, 2007.

[57] J. Nunez-Varela, J. Wyatt, and B. Ravindran, "Where do I look now? Gaze allocation during visually guided manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Minnesota,MN), pp. 4444–4449, May 2012.

[58] J. Nunez-Varela, J. Wyatt, and B. Ravindran, "Gaze allocation analysis for a visually guided manipulation task," in *Proceedings of Simulated of Adaptive Behaviour SAB2012*, (Denmark), pp. 44–53, Aug. 2012.

[59] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: Wiley-Interscience, 1994.

[60] M. L. Littman, *Algorithms for sequential decision making*. PhD thesis, Brown University, Rhode Island, May 1996.

[61] M. Hauskrecht and B. Kveton, "Linear program approximations for factored continuous-state Markov decision processes," *Advances in Neural Information Processing Systems*, vol. 16, no. 1, pp. 1–8, 2004.

[62] L. Li and M. Littman, "Lazy approximation for solving continuous finite-horizon MDPs," in *Proc. National Conf. on AI*, (California), pp. 1175–1180, Aug. 2005.

[63] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[64] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press Cambridge, 2008.

[65] A. R. Cassandra, *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, Rhode Island, May 1998.

[66] N. Roy, *Finding approximate POMDP solutions through belief compression*. PhD thesis, Carnegie Mellon University, Pittsburgh, Aug. 2003.

[67] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *AI Journal*, vol. 112, no. 1, pp. 181–211, 1999.

[68] A. McGovern, D. Precup, B. Ravindran, S. Singh, and R. Sutton, "Hierarchical optimal control of MDPs," in *Proc. 10th Yale Workshop on Adaptive and Learning Systems*, (New Haven,CT), pp. 186–191, 1998.

[69] M. Hauskrecht, N. Meuleau, L. Kaelbling, T. Dean, and C. Boutilier, "Hierarchical solution of Markov decision processes using macro-actions," in *Proc. 14th Conf. on Uncertainty in AI*, (Madison,WI), pp. 220–229, 1998.

[70] D. Precup, *Temporal abstraction in reinforcement learning*. PhD thesis, Univ. of Massachusetts Amherst, Massachusetts, May 2000.

[71] R. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1998.

[72] J. Wyatt, *Exploration and inference in learning from reinforcement*. PhD thesis, University of Edinburgh, Edinburgh,UK, 1998.

[73] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement learning: A survey," *Journal of AI Research*, vol. 4, no. 1, pp. 237–285, 1996.

[74] S. Ishii, W. Yoshida, and J. Yoshimoto, "Control of exploitation–exploration meta-parameter in reinforcement learning," *Neural networks*, vol. 15, no. 4, pp. 665–687, 2002.

[75] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[76] S. Bradtke and M. Duff, "Reinforcement learning methods for continuous-time Markov decision problems," *Adv. in Neural Inf. Proc. Sys.*, vol. 8, pp. 393–400, 1995.

[77] D. Bertsekas and J. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 560–564, 1995.

[78] M. Littman, A. Cassandra, and L. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Proceedings of the 12th International Conference on Machine Learning*, (San Francisco,CA), pp. 362–370, 1995.

[79] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," tech. rep., Adaptive Syst. Lab., McMaster Univ., 2003.

[80] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[81] S. Julier and J. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *AeroSense'97*, (Orlando,FL), pp. 182–193, 1997.

[82] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[83] R. Rubinstein, *Simulation and the Monte Carlo method*. New York, NY: John Wiley and Sons, 1981.

[84] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York, NY: Springer, 2001.

[85] I. Rekleitis, "A particle filter tutorial for mobile robot localization," tech. rep., Centre for Intelligent Machines, McGill University, 2004.

[86] S. Thrun, "Particle filters in robotics," in *Proceedings Eighteenth Conf. on Uncertainty in A.I.*, (USA), pp. 511–518, 2002.

[87] S. Thrun, "Robotic mapping: A survey," *Exploring artificial intelligence in the new millennium*, vol. 1, pp. 1–35, 2003.

[88] R. Dearden and D. Clancy, "Particle filters for real-time fault detection in planetary rovers," in *Proceedings of the Thirteenth International Workshop on Principles of Diagnosis*, (Austria), pp. 1–6, 2002.

[89] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proceedings 2001 ICRA. IEEE International Conference*, (Seoul), pp. 1665–1670, 2001.

[90] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 271–281, 2003.

[91] K. Murphy, "Bayesian map learning in dynamic environments," *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, no. 1, pp. 1015–1021, 2000.

[92] T. Chen, T. Schon, H. Ohlsson, and L. Ljung, "Decentralized particle filter with arbitrary state decomposition," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 465–478, 2011.

[93] R. Steinman, "Gaze control under natural conditions," in *The Visual Neurosciences* (L. Chalupa and J. Werner, eds.), pp. 1339–1356, MIT Press, 2003.

[94] P. Lennie, "The cost of cortical computation," *Current Biology*, vol. 13, no. 6, pp. 493–497, 2003.

[95] G. Yang, L. Dempere-Marco, X. Hu, and A. Rowe, "Visual search: Psychophysical models and practical applications," *Image and vision computing*, vol. 20, no. 4, pp. 291–305, 2002.

[96] S. Martinez-Conde, S. Macknik, and D. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[97] S. Martinez-Conde, S. Macknik, X. Troncoso, and D. Hubel, "Microsaccades: A neurophysiological analysis," *Trends in neurosciences*, vol. 32, no. 9, pp. 463–475, 2009.

[98] A. Ramakrishnan, S. Chokhandre, and A. Murthy, "Voluntary control of multisaccade gaze shifts during movement preparation and execution," *Journal of neurophysiology*, vol. 103, no. 5, pp. 2400–2416, 2010.

[99] D. Finlay, "Motion perception in the peripheral visual field," *Perception*, vol. 11, no. 4, pp. 457–462, 1982.

[100] M. To, I. Gilchrist, T. Troscianko, J. Kho, and D. Tolhurst, "Perception of differences in natural-image stimuli: Why is peripheral viewing poorer than foveal?," *ACM Transactions on Applied Perception*, vol. 6, no. 4, pp. 1–9, 2009.

[101] H. Strasburger and I. R. M. Jüttner, "Peripheral vision and pattern recognition: A review," *Journal of Vision*, vol. 11, no. 5, pp. 1–82, 2011.

[102] J. Pelz, "Portable eyetracking: A study of natural eye movements," in *Human Vision and Electronic Imaging V*, (San Jose,CA), pp. 566–582, 2000.

[103] J. Babcock and J. Pelz, "Building a lightweight eyetracking headgear," in *Proc. Symposium on Eye tracking research & applications*, (San Antonio,TX), pp. 109–114, 2004.

[104] M. Castelhano, M. Mack, and J. Henderson, "Viewing task influences eye movement control during active scene perception," *Journal of Vision*, vol. 9, no. 3, pp. 1–15, 2009.

[105] B. Schnitzer and E. Kowler, "Eye movements during multiple readings of the same text," *Vision research*, vol. 46, no. 10, pp. 1611–1632, 2006.

[106] J. Droll and M. Hayhoe, "Trade-offs between gaze and working memory use," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 6, pp. 1352–1365, 2007.

[107] P. Gibbons, P. Culverhouse, and G. Bugmann, "Fabric manipulation: An eye tracking experiment," *Proceedings of Taros*, vol. 8, pp. 130–134, 2008.

[108] R. Foerster, E. Carbone, H. Koesling, and W. Schneider, "Saccadic eye movements in a high-speed bimanual stacking task: Changes of attentional control during learning and automatization," *Journal of Vision*, vol. 11, no. 7, pp. 1–16, 2011.

[109] M. Land, N. Mennie, and J. Rusted, "The roles of vision and eye movements in the control of activities of daily living," *Perception*, vol. 28, no. 11, pp. 1311–1328, 1999.

[110] J. Jovancevic, B. Sullivan, and M. Hayhoe, "Control of attention and gaze in complex environments," *Journal of Vision*, vol. 6, no. 12, pp. 1431–1450, 2006.

[111] J. Jovancevic-Misic and M. Hayhoe, "Adaptive gaze control in natural environments," *The Journal of Neuroscience*, vol. 29, no. 19, pp. 6234–6238, 2009.

[112] T. Foulsham, E. Walker, and A. Kingstone, "The where, what and when of gaze allocation in the lab and the natural environment," *Vision research*, vol. 51, no. 17, pp. 1920–1931, 2011.

[113] M. Hayhoe, A. Shrivastava, R. Mruczek, and J. Pelz, "Visual memory and motor planning in a natural task," *Journal of Vision*, vol. 3, no. 1, pp. 49–63, 2003.

[114] M. Land and P. McLeod, "From eye movements to actions: How batsmen hit the ball," *Nature neuroscience*, vol. 3, no. 12, pp. 1340–1345, 2000.

[115] M. Land and S. Furneaux, "The knowledge base of the oculomotor system," *Phil. Trans. of the Royal Society B: Biological Sciences*, vol. 352, no. 1358, pp. 1231–1239, 1997.

[116] M. Hayhoe, T. McKinney, K. Chajka, and J. Pelz, "Predictive eye movements in natural vision," *Experimental brain research*, vol. 3, no. 168, pp. 1–12, 2012.

[117] J. Pelz and R. Canosa, "Oculomotor behavior and perceptual strategies in complex tasks," *Vision research*, vol. 41, no. 25, pp. 3587–3596, 2001.

[118] F. Kandil, A. Rotter, and M. Lappe, "Driving is smoother and more stable when using the tangent point," *Journal of Vision*, vol. 9, no. 1, pp. 1–11, 2009.

[119] D. Ballard, M. Hayhoe, and J. Pelz, "Memory representations in natural tasks," *Journal of Cognitive Neurosciences*, vol. 7, no. 1, pp. 66–88, 1995.

[120] R. Rensink, "Internal vs. external information in visual perception," in *Proc. 2nd inter. symposium on Smart graphics*, (Hawthorne,NY), pp. 63–70, 2002.

[121] D. Levin, N. Momen, I. Sarah, I. Drivdahl, and D. Simons, "Change blindness blindness: The metacognitive error of overestimating change-detection ability," *Visual Cognition*, vol. 7, no. 1-3, pp. 397–412, 2000.

[122] B. Tatler, M. Hayhoe, M. Land, and D. Ballard, "Eye guidance in natural vision: Reinterpreting salience," *Journal of vision*, vol. 11, no. 5, pp. 1–23, 2011.

[123] J. Najemnik and W. Geisler, "Eye movement statistics in humans are consistent with an optimal search strategy," *Journal of Vision*, vol. 8, no. 3, pp. 1–14, 2008.

[124] P. Verghese, "Active search for multiple targets is inefficient," *Vision Research (in press)*, 2012.

[125] P. Redgrave, T. Prescott, and K. Gurney, "The basal ganglia: a vertebrate solution to the selection problem?," *Neuroscience*, vol. 89, no. 4, pp. 1009–1023, 1999.

[126] P. Dayan and M. Walton, "A step-by-step guide to dopamine," *Biol Psychiatry*, vol. 71, no. 10, pp. 842–843, 2012.

[127] W. Schultz, L. Tremblay, and J. Hollerman, "Changes in behavior-related neuronal activity in the striatum during learning," *Trends in neurosciences*, vol. 26, no. 6, pp. 321–328, 2003.

[128] P. Redgrave and K. Gurney, "The short-latency dopamine signal: A role in discovering novel actions?," *Nature Reviews Neuroscience*, vol. 7, no. 12, pp. 967–975, 2006.

[129] O. Hikosaka, K. Nakamura, and H. Nakahara, "Basal ganglia orient eyes to reward," *J. of Neurophysiology*, vol. 95, no. 2, pp. 567–584, 2006.

[130] B. Lau and P. Glimcher, "Action and outcome encoding in the primate caudate nucleus," *The Journal of Neuroscience*, vol. 27, no. 52, pp. 14502–14514, 2007.

[131] P. Montague, S. Hyman, and J. Cohen, "Computational roles for dopamine in behavioural control," *Nature*, vol. 431, no. 7010, pp. 760–767, 2004.

[132] K. Samejima and K. Doya, "Multiple representations of belief states and action values in corticobasal ganglia loops," *Annals of the New York Academy of Sciences*, vol. 1104, no. 1, pp. 213–228, 2007.

[133] P. Dayan, "The role of value systems in decision making," in *Better than conscious? Decision Making, the Human Mind, and Implications For Institutions* (C. Engel and W. Singer, eds.), p. 5170, MIT Press, 2008.

[134] K. Wunderlich, P. Dayan, and R. Dolan, "Mapping value based planning and extensively trained choice in the human brain," *Nature Neuroscience (in press)*, 2012.

[135] B. Seymour, N. Daw, J. Roiser, P. Dayan, and R. Dolan, "Serotonin selectively modulates reward value in human decision-making," *The Journal of Neuroscience*, vol. 32, no. 17, pp. 5833–5842, 2012.

[136] K. Doya, S. Ishii, A. Pouget, and R. Rao, *Bayesian brain: Probabilistic approaches to neural coding*. Cambridge, MA: MIT press, 2007.

[137] M. Carrasco, "Visual attention: The past 25 years," *Vision research*, vol. 51, no. 13, pp. 1484–1525, 2011.

[138] J. Henderson, "Visual attention and eye movement control during reading and picture viewing," in *Eye movements and visual cognition: Scene perception and reading* (K. Reyner, ed.), p. 260283, Springer-Verlag, 1992.

[139] H. Deubel and W. Schneider, "Saccade target selection and object recognition: Evidence for a common attentional mechanism," *Vision research*, vol. 36, no. 12, pp. 1827–1837, 1996.

[140] S. Yantis, "Goal-directed and stimulus-driven determinants of attentional control," *Attention and performance*, vol. 18, pp. 73–103, 2000.

[141] A. Nuthmann and J. Henderson, "Object-based attentional selection in scene viewing," *Journal of Vision*, vol. 10, no. 8, pp. 1–19, 2010.

[142] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.

[143] J. Wolfe and T. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?," *Nature Reviews Neuroscience*, vol. 5, no. 6, pp. 495–501, 2004.

[144] J. Wolfe, "Visual attention," *Seeing*, vol. 2, pp. 335–386, 2000.

[145] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[146] R. Klein, "Inhibition of return," *Trends in cognitive sciences*, vol. 4, no. 4, pp. 138–147, 2000.

[147] L. Elazary and L. Itti, "Interesting objects are visually salient," *Journal of Vision*, vol. 8, no. 3, pp. 1–15, 2008.

[148] S. Ban, I. Lee, and M. Lee, "Dynamic visual selective attention model," *Neurocomputing*, vol. 71, no. 4, pp. 853–856, 2008.

[149] K. Rapantzikos, Y. Avrithis, and S. Kolias, "Vision, attention control, and goals creation system," in *Perception-Action Cycle: Models, Architectures, and Hardware* (V. C. et al., ed.), pp. 363–386, Springer, 2011.

[150] D. Culibrk, S. Sladojevic, N. Riche, M. Mancas, and V. Crnojevic, "Data-driven approach to dynamic visual attention modelling," in *Proc. of SPIE vol 8436*, pp. 1–11, 2012.

[151] N. Butko, L. Zhang, G. Cottrell, and J. Movellan, "Visual saliency model for robot cameras," in *IEEE International Conference on Robotics and Automation, ICRA 2008*, (Pasadena,CA), pp. 2398–2403, 2008.

[152] M. Schlesinger, D. Amso, S. Johnson, N. Hantehzadeh, and L. Gupta, "Using the icub simulator to study perceptual development: A case study," *In press*.

[153] S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal, "Overt visual attention for a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. IROS*, (Maui,HI), pp. 2332–2337, 2001.

[154] N. Bruce and J. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of Vision*, vol. 9, no. 3, pp. 1–24, 2009.

[155] A. Soltani and C. Koch, "Visual saliency computations: Mechanisms, constraints, and the effect of feedback," *The Journal of Neuroscience*, vol. 30, no. 38, pp. 12831–12843, 2010.

[156] P. Roelfsema, A. van Ooyen, and T. Watanabe, "Perceptual learning rules based on reinforcers and attention," *Trends in cognitive sciences*, vol. 14, no. 2, pp. 64–71, 2010.

[157] S. Frintrop, E. Rome, and H. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, pp. 1–39, 2010.

[158] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[159] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision research*, vol. 45, no. 2, pp. 205–231, 2005.

[160] G. Zelinsky, W. Zhang, B. Yu, X. Chen, and D. Samaras, "The role of top-down and bottom-up processes in guiding eye movements during visual search," in *Advances in neural information processing systems 18* (Y. Weiss, B. Scholkopf, and J. Platt, eds.), p. 15691576, MIT Press, 2006.

[161] F. Hamker, "Modeling feature-based attention as an active top-down inference process," *BioSystems*, vol. 86, no. 1, pp. 91–99, 2006.

[162] F. Hamker, "Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search," *Psychological review*, vol. 113, no. 4, pp. 766–789, 2006.

[163] F. Orabona, G. Metta, and G. Sandini, "Object-based visual attention: A model for a behaving robot," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Diego,CA), pp. 89–97, 2005.

[164] N. Hawes and J. Wyatt, "Towards context-sensitive visual attention," in *Proc. of the Second International Cognitive Vision Workshop (ICVW06)*, (Graz,Austria), 2006.

[165] J. Bohg, C. Barck-Holst, K. Huebner, M. Ralph, B. Rasolzadeh, D. Song, and D. Kragic, "Towards grasp-oriented visual perception for humanoid robots," *International Journal of Humanoid Robotics*, vol. 6, no. 3, pp. 387–434, 2009.

[166] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1054–1065, 2008.

[167] K. Gurney, T. Prescott, J. Wickens, and P. Redgrave, "Computational models of the basal ganglia: From robots to membranes," *Trends in neurosciences*, vol. 27, no. 8, pp. 453–459, 2004.

[168] R. Bogacz and K. Gurney, "The basal ganglia and cortex implement optimal decision making between alternative actions," *Neural Computation*, vol. 19, no. 2, pp. 442–477, 2007.

[169] R. Bogacz and T. Larsen, "Integration of reinforcement learning and optimal decision-making theories of the basal ganglia," *Neural computation*, vol. 23, no. 4, pp. 817–851, 2011.

[170] R. Bogacz, "Optimal decision making theories," in *Handbook of reward and decision making* (J. Dreher and L. Tremblay, eds.), p. 375397, Academic Press, 2009.

[171] R. P. Rao, "Decision making under uncertainty: a neural model based on partially observable Markov decision processes," *Frontiers in Computational Neuroscience*, vol. 4, no. 1, pp. 1–18, 2010.

[172] P. Trimmer, A. Houston, J. Marshall, R. Bogacz, E. Paul, M. Mendl, and J. McNamara, "Mammalian choices: combining fast-but-inaccurate and slow-but-accurate decision-making systems," *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, no. 1649, pp. 2353–2361, 2008.

[173] J. Marshall, R. Bogacz, and I. Gilchrist, "Consistent implementation of decisions in the brain," *PloS one*, vol. 7, no. 9, p. e43443, 2012.

[174] K. Gurney, T. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. ii. analysis and simulation of behaviour," *Biological cybernetics*, vol. 84, no. 6, pp. 411–423, 2001.

[175] G. Gancarz and S. Grossberg, "A neural model of the saccade generator in the reticular formation," *Neural Networks*, vol. 11, no. 7, pp. 1159–1174, 1998.

[176] G. Gancarz and S. Grossberg, "Modeling the role of basal ganglia in saccade generation: Is the indirect pathway the explorer?," *Neural Networks*, vol. 24, no. 8, pp. 801–813, 2011.

[177] D. Ognibene, C. Balkenius, and G. Baldassarre, "Integrating epistemic action (active vision) and pragmatic action (reaching): A neural architecture for camera-arm robots," in *From Animals to Animats 10, Proceedings of the Tenth International Conference on Simulation of Adaptive Behavior*, (Osaka,Japan), pp. 220–229, 2008.

[178] G. Aragon-Camarasa, H. Fattah, and J. Siebert, "Towards a unified visual framework in a binocular active robot vision system," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 276–286, 2010.

[179] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[180] P. Forssén, "Learning saccadic gaze control via motion prediciton," in *Fourth Canadian Conference on Computer and Robot Vision, CRV'07*, (Montreal,Canada), pp. 44–54, 2007.

[181] L. Renninger, J. Coughlan, P. Verghese, and J. Malik, "An information maximization model of eye movements," *Advances in neural information processing systems*, vol. 17, pp. 1121–1128, 2005.

[182] T. Arbel and F. Ferrie, "Entropy-based gaze planning," *Image and vision computing*, vol. 19, no. 11, pp. 779–786, 2001.

[183] S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Meissner, G. Bradski, P. Baumstarck, S. Chung, and A. Ng, "Peripheral-foveal vision for real-time object recognition and tracking in video," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, (Hyderabad,India), 2007.

[184] N. Butko and J. Movellan, "I-POMDP: An infomax model of eye movement," in *7th IEEE Inter. Conf. on Development and Learning*, (Monterey,CA), pp. 139–144, 2008.

[185] N. Butko and J. Movellan, "Infomax control of eye movements," *IEEE Tran. on Aut. Mental Development*, vol. 2, no. 2, pp. 91–107, 2010.

[186] N. Butko and J. Movellan, "Learning to look," in *Proceedings of the 2010 IEEE International Conference on Development and Learning (ICDL)*, (Ann Arbor,MI), pp. 70–75, 2010.

[187] J. Vogel and N. de Freitas, "Target-directed attention: Sequential decision-making for gaze planning," in *IEEE Inter. Conf. on Robotics and Automation, 2008. ICRA 2008*, (Pasadena,CA), pp. 2372–2379, May 2008.

[188] S. Minut and S. Mahadevan, "A reinforcement learning model of selective visual attention," in *Proc. 5th Inter. Conf. on Autonomous agents*, (Montreal,Canada), pp. 457–464, 2001.

[189] J. Stober, L. Fishgold, and B. Kuipers, "Learning the sensorimotor structure of the foveated retina," in *Proceedings of the Ninth International Conference on Epigenetic Robotics*, 2009.

[190] D. Ognibene, C. Balkenius, and G. Baldassarre, "E-HiPPo: Extensions to hierarchical POMDP-based visual planning on a robot," in *Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG '08)*, (Edinburgh,UK), pp. 33–38, 2008.

[191] D. Marocco and D. Floreano, "Active vision and feature selection in evolutionary behavioral systems," in *Proceedings of Simulated of Adaptive Behaviour SAB2002*, (Cambridge,MA), pp. 247–255, 2002.

[192] M. Hoffman, D. Grimes, A. Shon, and R. Rao, "Active vision and receptive field development in evolutionary robots," *Evolutionary Computation*, vol. 13, no. 4, pp. 527–544, 2005.

[193] A. Borji, D. Sihite, and L. Itti, "What/where to look next? Modeling top-down visual attention in complex interactive environments," *IEEE Transactions on Systems, Man, and Cybernetics, PART A-SYSTEMS AND HUMANS (In press)*.

[194] A. Seekircher, T. Laue, and T. Röfer, "Entropy-based active vision for a humanoid soccer robot," in *RoboCup 2010: Robot Soccer World Cup XIV*, pp. 1–12, 2011.

[195] S. Kohlbrecher, A. Stumpf, and O. von Stryk, "Grid-based occupancy mapping and automatic gaze control for soccer playing humanoid robots," in *Proc. 6th Workshop on Humanoid Soccer Robots*, (Slovenia), pp. 1–6, 2011.

[196] S. Yeo, M. Lesmana, D. Neog, and D. Pai, "Grasping of extrafoveal targets: A robotic model," *New Ideas in Psychology*, vol. 29, no. 3, pp. 235–259, 2011.

[197] M. Hülse, S. McBride, and M. Lee, "Fast learning mapping schemes for robotic hand-eye coordination," *Cognitive Computation*, vol. 2, no. 1, pp. 1–16, 2010.

[198] D. Marr, *Vision*. San Francisco: W. H. Freeman and Co., 1982.

[199] S. Whitehead and D. Ballard, "Learning to perceive and act by trial and error," *Machine Learning*, vol. 7, no. 1, pp. 45–83, 1991.

[200] G. Salgian and D. Ballard, "Visual routines for autonomous driving," in *Sixth International Conference on Computer Vision*, pp. 876–882, 2008.

[201] D. Hernandez, J. Cabrera, A. Naranjo, A. Dominguez, and J. Isern, "Gaze control in a multiple-task active-vision system," in *Proc. of the 5th Intl. Conf. on Computer Vision Systems (ICVS07)*, (Bielefeld,Germany), 2007.

[202] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[203] D. Kragic and H. Christensen, "Survey on visual servoing for manipulation," tech. rep., KTH, School of Computer Science and Communication, 2002.

[204] R. Cupec, O. Lorch, and G. Schmidt, "Vision-guided humanoid walking - Concepts and experiments," in *Proc. of RAAD03, 12th Inter. Workshop on Robotics in Alpe-Adria-Danube Region*, (Cassino,Italy), pp. 1–6, 2003.

[205] M. Argyle and J. Dean, "Eye-contact, distance and affiliation," *Sociometry*, vol. 28, no. 3, pp. 289–304, 1965.

[206] C. Sidner, C. Kidd, C. Lee, and N. Lesh, "Where to look: A study of human-robot engagement," in *Proc. of the 9th international conference on Intelligent user interfaces*, (Madeira,Portugal), pp. 78–84, 2004.

[207] M. Hoffman, D. Grimes, A. Shon, and R. Rao, "A probabilistic model of gaze imitation and shared attention," *Neural Networks*, vol. 19, no. 3, pp. 299–310, 2006.

[208] C. Rich, B. Ponsler, A. Holroyd, and C. Sidner, "Recognizing engagement in human-robot interaction," in *5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2010*, (Osaka), pp. 375–382, 2010.

[209] M. Hashimoto, H. Kondo, and Y. Tamatsu, "Gaze guidance using a facial expression robot," *Advanced robotics*, vol. 23, no. 14, pp. 1831–1848, 2009.

[210] S. Andrist, T. Pejsa, B. Mutlu, and M. Gleicher, "Designing effective gaze mechanisms for virtual agents," in *Proc. of the 2012 ACM annual conference on Human Factors in Computing Systems*, (Austin,TX), pp. 705–714, 2012.

[211] T. Halverson and A. Hornof, "A computational model of active vision for visual search in human-computer interaction," *Human-Computer Interaction*, vol. 26, no. 4, pp. 285–314, 2011.

[212] A. Ude, T. Shibata, and C. Atkeson, "Real-time visual system for interaction with a humanoid robot," *Robotics and Autonomous Systems*, vol. 37, no. 2, pp. 115–125, 2001.

[213] M. Lopes, A. Bernardino, J. Santos-Victor, K. Rosander, and C. von Hofsten, "Biomimetic eye-neck coordination," in *IEEE 8th International Conference on Development and Learning, ICDL 2009*, (Shanghai), pp. 1–8, 2009.

[214] M. Lesmana and D. Pai, "A biologically inspired controller for fast eye movements," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*, (Shanghai), pp. 3670–3675, 2011.

[215] A. Haith and S. Vijayakumar, "Robustness of VOR and OKR adaptation under kinematics and dynamics transformations," in *IEEE International Conference on Development and Learning, ICDL 2007*, (London), pp. 37–42, 2007.

[216] F. Chao, M. Lee, and J. Lee, "A developmental algorithm for ocular-motor coordination," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 239–248, 2010.

[217] J. Wolfe, "Visual search," in *Attention* (H. Pashler, ed.), p. 1373, Psychology Press, 1998.

[218] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[219] J. Wolfe, "Guided search 2.0 a revised model of visual search," *Psychonomic bulletin and review*, vol. 1, no. 2, pp. 202–238, 1994.

[220] J. Findlay, "Saccade target selection during visual search," *Vision Research*, vol. 37, no. 5, pp. 617–631, 1997.

[221] B. Motter and J. Holsapple, "Saccades and covert shifts of attention during active visual search: Spatial distributions, memory, and items per fixation," *Vision Research*, vol. 47, pp. 1261–1281, 2007.

[222] G. Zelinsky, "A theory of eye movements during target acquisition," *Psychological review*, vol. 115, no. 4, pp. 787–835, 2008.

[223] X. Chen and G. Zelinsky, "Real-world visual search is dominated by top-down guidance," *Vision research*, vol. 46, no. 24, pp. 4118–4133, 2006.

[224] J. Henderson, J. Brockmole, M. Castelhano, and M. Mack, "Visual saliency does not account for eye movements during visual search in real-world scenes," in *Eye movements: A window on mind and brain* (R. van Gompel, M. Fischer, W. Murray, and R. Hill, eds.), pp. 537–562, Elsevier, Oxford, 2007.

[225] J. Wolfe, G. Alvarez, R. Rosenholtz, Y. Kuzmova, and A. Sherman, "Visual search for arbitrary objects in real scenes," *Attention, Perception, and Psychophysics*, vol. 73, no. 6, pp. 1650–1671, 2011.

[226] G. Malcolm and J. Henderson, "Combining top-down processes to guide eye movements during real-world scene search," *Journal of Vision*, vol. 10, no. 2, pp. 1–11, 2010.

[227] B. H.-S. A. Oliva, "Person, place, and past influence eye movements during visual search," in *Proc. of the 32nd Annual Meeting of the Cognitive Science Society*, (Austin,TX), pp. 820–825, 2010.

[228] V. Navalpakkam, C. Koch, and P. Perona, "Homo economicus in visual search," *Journal of Vision*, vol. 9, no. 1, pp. 1–16, 2009.

[229] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[230] R. Rao, G. Zelinsky, M. Hayhoe, and D. Ballard, "Eye movements in iconic visual search," *Vision research*, vol. 42, no. 11, pp. 1447–1464, 2002.

[231] K. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva, "Modelling search for people in 900 scenes: A combined source model of eye guidance," *Visual Cognition*, vol. 17, no. 6-7, pp. 945–978, 2009.

[232] Y. Ye and J. Tsotsos, "Sensor planning for 3d object search," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 145–168, 1999.

[233] J. Tsotsos and K. Shubina, "Attention and visual search: Active robotic vision systems that search," in *5th International Conference on Computer Vision Systems*, (Bielefeld,Germany), pp. 21–32, 2007.

[234] K. Shubina and J. Tsotsos, "Visual search for an object in a 3d environment using a mobile robot," tech. rep., York University, 2008.

[235] F. Saidi, O. Stasse, and K. Yokoi, "Active visual search by a humanoid robot," *Recent Progress in Robotics: Viable Robotic Service to Human*, vol. 16, pp. 171–184, 2008.

[236] A. Aydemir, K. Sjoo, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *IEEE Inter. Conf. on Robotics and Automation (ICRA) 2011*, (Shangai,China), pp. 2818–2824, 2011.

[237] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *Proc. of the 22th International joint conference on Artificial Intelligence-Vol3*, (Barcelona,Spain), pp. 2442–2449, 2011.

[238] D. Lopez, K. Sjoo, C. Paul, and P. Jensfelt, "Hybrid laser and vision based object search and localization," in *IEEE Inter. Conf. on Robotics and Automation (ICRA) 2008*, (Pasadena,CA), pp. 2636–2643, 2008.

[239] P. Forssen, D. Meger, K. Lai, S. Helmer, J. Little, and D. Lowe, "Informed visual search: Combining attention and object recognition," in *IEEE Inter. Conf. on Robotics and Automation (ICRA) 2008*, (Pasadena,CA), pp. 935–942, 2008.

[240] S. Zhang and M. Sridharan, "Active visual sensing and collaboration on mobile robots using hierarchical POMDPs," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 181–188, 2012.

[241] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. Dimarogonas, and D. Kragic, "Dual arm manipulation A survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, p. 13401353, 2012.

[242] M. Toussaint, M. Gienger, and C. Goerick, "Compliance control with dual-arm humanoid robots: Design, planning and programming," in *10th IEEE-RAS Inter. Conference on Humanoid Robots*, (Nashville, TN), pp. 275–281, 2010.

[243] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[244] D. Kragic and H. Christensen, "Robust visual servoing," *The international journal of robotics research*, vol. 22, no. 10-11, p. 923939, 2003.

[245] S. Yeo, M. Lesmana, D. Neog, and D. Pai, "Eyecatch: Simulating visuomotor coordination for object interception," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 42–52, 2012.

[246] D. Srinivasan, *Visuomotor coordination in symmetric and asymmetric bimanual reaching tasks*. PhD thesis, The University of Michigan, Michigan, 2010.

[247] M. Beetz, U. Klank, A. Maldonado, D. Pangercic, and T. Rühr, "Robotic roommates making pancakes-look into perception-manipulation loop," in *IEEE International Conference on Robotics and Automation (ICRA), Workshop on Mobile Manipulation: Integrating Perception and Manipulation*, pp. 9–13, 2011.

[248] A. Quaid and R. Hollis, "Cooperative 2-dof robots for precision assembly," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2188–2193, 1996.

[249] E. Ostergaard, G. Sukhatme, and M. Matari, "Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks," in *Proceedings of the fifth international conference on Autonomous agents*, (Montreal,Canada), pp. 29–35, 2001.

[250] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact behaviour generation," in *7th IEEE-RAS Inter. Conference on Humanoid Robots*, (Pittsburgh, PA), pp. 122–129, 2007.

[251] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS*, (St. Louis,MO), pp. 2464–2470, 2009.

[252] K. Rohanimanesh, *Concurrent Decision Making in Markov Decision Processes*. PhD thesis, Univ. of Massachusetts Amherst, Massachusetts, Feb. 2006.

[253] S. Abdallah and V. Lesser, "Modeling task allocation using a decision theoretic model," in *Proc. 4th Inter. joint conference on autonomous agents and multiagent systems*, (Netherlands), pp. 719–726, 2005.

[254] M. Ghavamzadeh, S. Mahadevan, and R. Makar, "Hierarchical multi-agent reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, p. 197229, 2006.

[255] Mausam, *Stochastic Planning with Concurrent, Durative Actions*. PhD thesis, Univ. of Washington, Seattle, Jan. 2007.

[256] H. L. S. Younes, *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon Univ., Pittsburgh, Jan. 2005.

[257] M. Roth, R. Simmons, and M. Veloso, "Reasoning about joint beliefs for execution-time communication decisions," in *Proc. 4th Inter. joint conference on autonomous agents and multiagent systems*, (Netherlands), pp. 786–793, 2005.

[258] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, p. 156172, 2008.

[259] C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in *Proc. ICML-2002 19th Inter. Conf. on Machine Learning*, (Sydney,Australia), pp. 227–234, 2002.

[260] B. Banerjee, S. Sen, and J. Peng, "On-policy concurrent reinforcement learning," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 16, no. 4, p. 245260, 2004.

[261] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

[262] C. Rothkopf, *Modular models of task based visually guided behaviour*. PhD thesis, University of Rochester, New York, May 2008.

[263] W. Garage, "Robot operating system." `http://www.willowgarage.com/pages/software/ros-platform`, 2012.

[264] A. Robotics, "Nao." `http://www.aldebaran-robotics.com`, 2012.

[265] C. D. Granville, D. Wang, J. Southerland, A. Fagg, and R. Platt, "Grasping affordances: Learning to connet vission to hand action," *The Path to Autonomous Robots*, pp. 1–22, 2009.

[266] C. Gielen, T. Dijkstra, I. Roozen, and J. Welten, "Coordination of gaze and hand movements for tracking and tracing in 3d," *Cortex*, vol. 45, no. 3, pp. 340–355, 2009.

[267] A. Leonardis and Fidler, "Learning hierarchical representations of object categories for robot vision," *Robotics Research*, vol. 66, pp. 99–110, 2011.

[268] F. Kaplan and P. Oudeyer, "Intrinsically motivated machines," *LNCS: 50 years of artificial intelligence*, vol. 4850, pp. 303–314, 2007.

[269] M. Ernst and M. D. Luca, "Multisensory perception: from integration to remapping," in *Sensory cue integration* (J. Trommershauser, K. Kording, and M. Landy, eds.), pp. 224–250, Oxford Press, 2011.