Human studies have shown that gaze shifts are mostly driven by the current task demands. In manipulation tasks, gaze 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 work presents new computational models of gaze shifting, where the agent imagines ahead in time the informational effects of possible gaze fixations. Building on our previous work, the contributions of this paper are: a) the presentation of two new gaze control models; b) comparison of their performance to our previous model; c) results showing the fit of all these models to previously published human data; and d) integration of a visual search process. The first new model selects the gaze that most reduces positional uncertainty of landmarks (Unc), and the second maximises expected rewards by reducing positional uncertainty (RU). Our previous approach maximises the expected gain in cumulative reward by reducing positional uncertainty (RUG). In experiment b) the models are tested on a simulated humanoid robot performing a manipulation task, and each model’s performance is characterised by varying three environmental variables. This experiment provides evidence that the RUG model has the best overall performance. In experiment c) we compare the hand-eye coordination timings of the models in a robot simulation to those obtained from human data. This provides evidence that only the models that incorporate both uncertainty and reward (RU and RUG) match human data.

Categories and Subject Descriptors: I.2.10 [Vision and Scene Understanding] Perceptual reasoning
General Terms: Experimentation, Performance, Theory
Additional Key Words and Phrases: Decision Making, Gaze Control, Reinforcement Learning

ACM Reference Format:
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION
Most of our daily tasks require us to act under uncertain and incomplete information. Only by using sensing to fill in missing information and reduce task relevant uncertainty can these tasks be accomplished. One such sense is vision, which can actively explore the scene to gather information that would help us interact with our environment [Bajcsy 1988; Findlay and Gilchrist 2003]. Even though our work does not follow a purely biological approach, it is motivated by psychophysical human studies about the relation between gaze shifts and actions during the execution of tasks. Empirical evidence from these studies has led to three important findings: i) gaze shifts in the scene are, in general, driven...
by the ongoing task [Yarbus 1967; Land 2009]; ii) spatially, gaze is directed to task relevant objects and to the sites where actions are taking place [Johansson et al. 2001; Rothkopf et al. 2007]; iii) temporally, our actions are guided by the visual system [Ballard et al. 1992; Johansson et al. 2001].

One question is what mechanisms a rational decision maker could employ to select a gaze location given limited information and limited computation time. Another question is how humans select the next fixation location. Previous work has suggested that the answers to both questions might be similar: human eye movement behaviour is consistent with the use of decision making mechanisms for fixation selection, i) that are Bayes' rational [Najemnik and Geisler 2005], or ii) that try to maximise reward values [Navalpakkam et al. 2010; Schutz and Gegenfurtner 2010]. The aim of our work is to investigate these claims further, by examining in detail the formulation and behaviour of three Bayes' rational mechanisms for fixation selection during the performance of manipulation tasks. This paper makes four contributions. First we present two new models of fixation selection which are based on a model we presented in [Nunez-Varela et al. 2012b]. All three models are similar in that they calculate the benefit of a fixation by imagining its effect on the agent’s information state. In order to distinguish these models they are examined in two ways: relative performance in controlled conditions and goodness of fit to human data. Thus our second contribution is a study of the performance of each model in a simulation of a humanoid robot performing a manipulation task previously explored in [Nunez-Varela et al. 2012b; 2012a]. This paper plots the degradation of each strategy’s performance as i) grasp sensitivity increases, ii) observation reliability decreases and iii) the field of view narrows. Our third contribution is the integration of a visual search mechanism into the fixation selection process. In the fourth contribution the models are compared according to their fit to previously published human data on hand-eye coordination in a manipulation task [Johansson et al. 2001].

Next, we first define the gaze control problem and how it can be decomposed. Second, the benefit of robot simulation to study gaze control is discussed and the manipulation tasks are described. Finally, the structure of the rest of the paper is given.

1.1 Defining the Gaze Control Problem

One of the key characteristics of the human eye is the inhomogeneity of the retina, only a small region of the retina’s central part has high acuity (fovea). Due to this feature the eyes must actively move in order to obtain detailed views of different parts of the scene [Steinman 2003]. Although different eye movements exist, we are only interested in saccades, that shift gaze from one fixation location to another in rapid jump-like movements. After a saccade is completed the eyes are momentarily stationary and a fixation occurs, where visual information is captured and analysed [Findlay and Gilchrist 2003]. Based on this, and a small abusing of common terminology, we define gaze control as the problem of i) selecting fixation locations, ii) moving and centring the agent’s eyes on a particular fixation location, and iii) analysing the current viewpoint. This work is solely interested in i) the fixation selection process¹, where, given a number of task relevant landmarks (e.g. objects on a table), the agent decides what landmark to fixate next by imagining the informational effects of fixating each available landmark. Thus, the main problem faced by the agent 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

¹Because our work is solely concerned with i), it is simpler to think about ii) and iii) as separate processes (Section 2.1 explains how we deal with ii) and iii)). However, we do not advocate that particular sequential configuration. The design and implementation of these processes may execute simultaneously at different levels, especially if a biological approach is followed.
Models of Gaze Control for Manipulation Tasks

should be treated as a serial resource that must be first \textit{allocated} to a particular manipulation motor system, so that this motor system is able to select a particular fixational landmark. In other words, \textit{gaze allocation} is defined as a scheduling problem [Sprague et al. 2007] that assigns the choice about \textit{where to look} to a specific manipulation motor system. In summary, our models of gaze control first allocate gaze to a manipulation motor system, then this motor system is given the ability to select its preferred landmark location. Our gaze control models\footnote{Throughout the paper we might employ the general term of \textit{gaze control} to refer to the \textit{fixation selection process}.} provide an integrated account where both problems, \textit{gaze allocation} and \textit{where to look}, are dealt with in a single control mechanism.

1.2 Using Robotic Simulations to Study Gaze Control

The implementation of computational models in robotic platforms allows testing under configurations that would be difficult in humans (e.g. the robot's field of view can easily be altered between trials). Robotic simulations give configuration flexibility not only for the robot but also the environment (e.g. objects can be created or modified in real time). They also ease the repeatability of experiments in order to characterise the behaviour of models statistically. For these reasons, our gaze control models are implemented using the iCub simulator [Metta et al. 2008]. The iCub is a humanoid robot with multiple motor systems (Figure 1A), whilst the iCub simulator is an open source platform that employs the same controllers and modules used to control the real robot (Figure 1B). In this work, we make use of both arms (i.e. \textit{two manipulation motor systems}), where each has a hand with five fingers. Also, we make use of what we call the \textit{perceptual motor system}, which includes the robot’s head, neck and eyes. In order to test our gaze control models, two manipulation tasks are simulated:

—\textbf{Pick \& place task:} This task is used to characterise the performance of our models of gaze control. It consists of picking up objects from a table top and then placing them inside one of two containers (Figure 1B). In this task the arms 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 in trials of five minutes each.

—\textbf{Johansson’s task:} This task is used to show the fit of our models to human behavioural data. We simulate the manipulation task devised by Johansson et al. [2001], where subjects have to reach for and grasp a bar located on the table, and move it in a vertical plane to touch a target switch whilst avoiding an obstacle located between the table and the switch (Figure 7A).

1.3 Road Map

Section 2 provides an intuitive explanation of our three models of gaze control, along with a summary of related work. Section 3 formulates the robot’s system and our gaze control models. Section 4 characterises the models’ performance using the pick \& place task by varying three environmental conditions. Section 5 presents the experimental analysis for Johansson’s task and demonstrates how well the models match human data. Finally, Section 6 presents conclusions and future work.

2. CANDIDATE MODELS OF GAZE CONTROL

This section provides an intuitive explanation of our three computational models of gaze control, whilst the next section gives a detailed account of the models. The key idea is that the agent/robot selects where to fixate next by imagining (or predicting) the informational effects of fixating the landmarks currently known to the robot one step into the future. This kind of prediction is known in the artificial intelligence literature as \textit{one-step look ahead planning}. Predicting the benefit of the possible fixational landmarks is based on two parameters:
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 1B. There are five landmarks (three objects and two containers), where their positional uncertainty is represented as bivariate Gaussian densities (shown as ellipsoids in the top view of the same scene (Figure 1C)). 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 [Sutton and Barto 1998] (Section 3.3), 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 1B, 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 attached to each object/container. In fact, the robot 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 1C. 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.

Depending on how the robot employs these two quantities, at least three gaze control models emerge. Before such models are described we provide a list of assumptions considered throughout this paper.

2.1 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 do not model the exact inhomogeneity of the human eye. However, we make use of an observation model that produces a smoothly varying spatial acuity (Section 3.4.1). 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°), which 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: The control of saccadic movements at the motor level is outside the scope of our work. Instead we rely on standard robotic control techniques [Shibata et al. 2001], particularly those available to the iCub humanoid robot (Section 1.2) 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 use the 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.
Models of Gaze Control for Manipulation Tasks

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 the reward function (Section 3.3).

Grasping: This is a complex problem which is also outside the scope of this paper. Once again we will take advantage of the simulator by using a magnet-like function (explained in Section 3.1).

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 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.

2.2 Gaze Control based on Uncertainty Reduction (Unc)

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 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. Figure 1D shows this, where the small ellipsoids represent the predicted residual uncertainty (the big ellipsoids are those from Figure 1C). In this example, looking at the right container will benefit the 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 the left container is chosen to be fixated next (marked with an X).
disregarding task rewards (i.e. the values of performing manipulation actions) completely, 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. Our results show that this model is not task efficient.

2.3 Gaze Control based on Rewards and Uncertainty (RU)

Our second gaze control model extends the previous one 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 1E shows this predicted value for each landmark (white vertical bars) contrasted with the grey bars that represent the current value (taken from Figure 1C). 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 confirm that, in general, this model's performance is sub-optimal.

2.4 Gaze Control based on Rewards, Uncertainty and Gain (RUG)

Our third gaze control model eliminates the tendency of fixating landmarks with high value that might not offer new information. The model predicts the expected value for each landmark, exactly as the RU 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 1F, 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 [Sprague et al. 2007]). 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. It is important to evaluate how much these differences affect task performance. We have already demonstrated that the RUG gaze strategy outperforms two common baselines: random and round robin schemes [Nunez-Varela et al. 2012b]; and we have also characterised this model's robustness in terms of variations of three environmental variables [Nunez-Varela et al. 2012a]. In this paper, our new results show how the RUG model is the most effective of the three strategies presented above, along with the baselines. Next we summarise and compare previous work with our proposed models.

2.5 Related Work

Previous work has suggested that human eye movement behaviour is consistent with the use of decision making mechanisms for fixation selection that are Bayes' rational. Najemnik and Geisler [2005] found that humans achieve near-optimal performance compared to an ideal Bayesian searcher for the problem of finding a target in a cluttered environment. Even though we do not deal with visual search tasks, their derivation of the ideal searcher is similar to our models (specially the Unc gaze scheme) in that the aim is to maximise the information gain about target location. Their analysis also shows that it is more important the efficient processing of information on each fixation (instead of across fixations), which we also follow. Renninger et al. [2010] used a shape-matching task to investigate
whether humans move their eyes to locations that maximise total or local information about a shape. They found that subjects fixate the most informative locations of these shapes, i.e. they try to reduce local uncertainty. This is similar to our models, since the robot reasons about the reduction of uncertainty of single landmarks, rather than trying to reduce uncertainty of several landmarks in one fixation. Of interest to our work are tasks that involve body actions. During a driving task in a virtual environment, Sullivan et al. [2012] manipulated the uncertainty about the speed of the car (shown in the car’s speedometer). Human subjects performed more fixations to the speedometer and fixation duration increased when uncertainty about the car’s speed was introduced. On the other hand, it has also been suggested that human eye movement behaviour might follow fixation selection mechanisms that maximise reward. Navalpakkam et al. [2010] compared human behavioural data against three computational models of fixation selection in a visual search task. They evaluate the impact of reward value and visual saliency (low-level perceptual features), separately and combined. They found that subjects behave as an ideal Bayesian observer that maximises expected reward. This is similar to our RU and RUG models, but instead of saliency we combine expected rewards and location uncertainty of landmarks. Similarly, Schutz and Gegenfurtner [2010] found that human subjects integrate saliency and rewards in another visual search task. They also show that saccade latency is higher when rewards are used in the decision process compared to saliency only.

Our work builds directly from the reward-based approach developed by Sprague et al. [2007], in particular the idea of using gain for gaze allocation (as in our RUG model). Our problem is that gaze is shared amongst multiple manipulation motor systems, whilst they deal with multiple tasks running simultaneously on a single motor system. They use a virtual humanoid that has to follow a sidewalk whilst avoiding obstacles and collecting litter. Gaze is allocated to the task that is likely to lose more reward based on the current state uncertainty. However, they do not explicitly consider the problem of where to look (in their work specific fixation locations are pre-defined), and assume all uncertainty disappears after a fixation is made. Another interesting computational model for hand-eye coordination was defined by Erez et al. [2011]. In a simulated reaching task, two “hands” must reach towards a common point whilst passing a pair of obstacles. They model the joint behaviour of eye and hands, which might result in high-dimensional state and action spaces. In contrast, our system splits the decision process into the selection of fixations and manipulation actions separately.

Having given a quick summary of relevant literature, the next section will describe the robot’s control architecture and provide a detailed formulation of our models of gaze control.

3. GAZE CONTROL FOR MANIPULATION

Tasks that involve object manipulation require the coordination of gaze and manipulation actions. Thus, the problem of gaze control should not be treated in isolation but rather as part of the robot’s control architecture, as shown in Figure 2A. This architecture encodes assumptions which apply across all the models studied here. The operation of this architecture is divided, temporally, into the learning phase and the subsequent execution phase. This section starts by modelling the pick & place task. Then, the common architecture is described. Finally, the different gaze control models are formulated.

3.1 Modelling the Pick & Place Task

This section describes how the pick & place task is modelled (Section 1.2), that consists of picking up objects from the table top and then placing them inside one of two containers (Figure 1B). This description is included for clarity and is based on that of [Nunez-Varela et al. 2012a]. As mentioned above, for this task two manipulation motor systems are considered: 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
Each sub-task is then modelled as a semi-Markov decision process (SMDP)\textsuperscript{3} [Puterman 1994] using a discrete state representation\textsuperscript{4}. Therefore, each manipulation motor system $ms \in MS$ (where $MS$ is the set of motor systems), is modelled as a tuple $(S_{ms}, A_{ms}, T_{ms}, R)$, where $S_{ms}$ is the set of discrete states, $A_{ms}$ is the set of manipulation actions, $T_{ms} : S_{ms} \times A_{ms} \times S_{ms} \times \mathbb{R} \rightarrow [0,1]$ is the transition probability density, where $\mathbb{R}$ is the set of real numbers representing the duration time of each action, and $R : S_{ms} \times A_{ms} \rightarrow \mathbb{R}$ is the reward 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 $MS = \{right\_arm, left\_arm\}$. A factorised discrete state space is defined for both arms ($S_{right\_arm}$ and $S_{left\_arm}$). Each arm has the same three state variables: $armPosition = \{onObject, onTable, onContainer, outsideTable\}$, $handStatus = \{grasping, empty\}$, and $tableStatus = \{objectsOnTable, empty\}$. 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. Thus, there is a mapping (explained in Section 3.4.2) between the continuous information, used by the manipulation actions, and the discrete state space. The set of manipulation actions for each arm ($A_{right\_arm}$ and $A_{left\_arm}$), consists of six actions and their respective completion times (indicated as Gaussian distributions with their mean and standard deviation ($\mu_o, \sigma_o$) specified in seconds): moveToObject $(2.65,0.56)$, moveToTable $(2.95,0.45)$, moveToContainer $(3.25,0.33)$, graspObject $(2.87,1.1)$, releaseObject $(1.0,0.2)$, and noAction $(0.01,0.0)$. The completion time distributions were obtained by executing all actions in sequence for 60 minutes. These actions are implemented as commands in the iCub libraries using PID-like Cartesian motor controllers [Pattacini et al. 2010]. Because the location of objects/containers is continuous, the completion time for each manipulation action varies. This variation results because actions do not start exactly from the same position at all times. The time it takes for an action to complete at the current time step is denoted by $k_a$. As mentioned in Section 2.1, grasping objects is simplified, so for action graspObject we use a magnet-like function implemented in the iCub simulator. However, we can control the sensitivity of grasping (and reaching) by checking the offset between the centre of the hand and the centre of the object. So grasping and reaching actions can fail (except for releaseObject) if the offset is greater than some threshold (e.g. 1.0 cm). The iCub controllers have a limited positional accuracy and thus the minimum effective value of that threshold is 0.5 cm.

\textsuperscript{3}SMDPs are an extension of the Markov decision processes (MDPs) [Puterman 1994], typically used to model decision making problems. Whilst MDPs assume that actions have the same duration, SMDPs allow variable duration actions.

\textsuperscript{4}Tasks can also be modelled following a hierarchical approach using options [Sutton et al. 1999], a generalisation of SMDPs, where each option is defined as a temporally extended action that can be composed of one or multiple single-step actions.
3.2 Visual Memory

The central component in the system is the visual memory ($\mathcal{VM}$) (Figure 2A), defined as the set of ordered pairs $\mathcal{VM} = \{(e_1, pos(e_1)), (e_2, pos(e_2)), \ldots, (e_n, pos(e_n))\}$, where $e_i$ is the $i^{th}$ landmark’s id (for the pick & place task landmarks include objects and containers), and $pos(e_i)$ is a probability density function for the location of landmark $e_i$. These pairs capture the continuous state information needed for low-level control, 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. $S_{right\_arm}$ and $S_{left\_arm}$). 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 [Thrun et al. 2008]. A particle filter contains a set of particles $g_i = g^1, g^2, \ldots, g^J$, where $J$ denotes the number of particles, and each particle $g^j$ represents a possible location $(x,y)$ for landmark $e_i$. Three basic operations are used to maintain the $\mathcal{VM}$: i) when a new landmark is inside the robot’s view point, it is added to the $\mathcal{VM}$ and its particle filter is initialised; ii) the information about objects that are put inside a container or that fall from the table is removed from the $\mathcal{VM}$; iii) when a landmark, already in the $\mathcal{VM}$, is seen again its particle filter is updated. The idea is that with every update the particles will converge to a specific location, this update is explained below and is illustrated in Figure 2B.

3.3 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 [Bradtke and Duff 1995]. Each manipulation motor system $ms$ learns a policy $\pi_{ms} : S_{ms} \rightarrow A_{ms}$, that defines a mapping from states to actions, and contains the estimated expected return for each state-action pair ($Q$-values). These values are the ones described in Section 2, that quantify each manipulation action. The robot learns how to perform the task under an assumption of complete observability, i.e. the $\mathcal{VM}$ contains the complete list of landmarks ($e_i$) and their true location (i.e. $pos(e_i)$ becomes a Dirac delta function with value 1 at the true location). We follow a minimal time to goal strategy, so 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. Note that the same policy can be used for both arms in the pick & place task. The learning rule is formulated as:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma k_a \max_{a' \in A_{ms}} Q(s', a') - Q(s, a) \right]$$

(1)

Where $0 \leq \alpha \leq 1$ is the learning rate, $r$ is the immediate reward received for executing an action $a$ being in the discrete state $s$ (where $s \in S_{ms}$), $0 \leq \gamma \leq 1$ is a discount factor to indicate the trade-off between short term and long term rewards, and $k_a$ is the duration of action $a$. In our task, $\alpha = 0.2$, $\gamma = 0.9$, and $k_a$ is the time to complete action $a$ as specified in Section 3.1.

3.4 Execution Phase

Once the policies for each manipulation motor system are learnt, they can be used to solve the task provided there is state certainty. However, the robot must act, in general, under uncertain information. Here we assume that this uncertainty comes from the location of landmarks. First, we explain how the $\mathcal{VM}$ is maintained using a kind of Bayes’ filter by means of an observation model learnt off-line (visual analysis in Figure 2A). Then we describe how the robot performs the manipulation action selection. Finally, we explain how the fixation selection process is done for each gaze control model in turn.
3.4.1 Visual Analysis. At the beginning, the VM is empty. When the robot performs a fixation, visual input is captured and the landmarks inside the robot’s current viewpoint are detected, added to the VM, and their 3D locations estimated. Using stereo vision in the simulator provided precise 3D locations. However, 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. The 2D image coordinates are also used for triangulation, but as mentioned in Section 2.1, we use the simulator directly for object detection.

Then, an observation model (OM) was learnt off-line that represents the noise from the triangulation process (Figure 3A). The OM was created by systematically moving the robot’s fixation point and an object in a discretised space. Each fixation point and object location is represented using spherical coordinates: \((\theta_g, \phi_g, \text{radius}_g)\) and \((\theta_o, \phi_o, \text{radius}_o)\) for the gaze and object respectively (Figure 3B and C). Whilst varying these parameters we recorded the offset between the triangulated 3D location (using the simulator’s triangulation command mentioned above) and the true object’s location. The OM is generalised by triangulating using the distance at which the fixation point is located (\(\text{radius}_g\)). Thus, each data point is represented by the rest of the coordinates except for \(\text{radius}_g\). The resulting data points were binned according to a quantisation of \((\theta_g, \phi_g, \theta_o, \phi_o, \text{radius}_o)\) and then fitted by bivariate Gaussian densities. This resulted in 405 densities that cover a big part of the working space. Figure 3D shows nine examples of these densities, representing the case when the robot is looking at the centre of the table \((\theta_g = 140^\circ, \phi_g = 0^\circ, \theta_o = 70^\circ)\). Notice how the noise accrues as the object horizontal eccentricity in the FoV increases, as \(\phi_o\) and \(\text{radius}_o\) vary. The estimated location is more accurate when the robot looks directly at the object (central column, \(\phi_o = 0^\circ\)).

During the execution phase the OM is used to estimate the location of the detected landmarks. These estimates are used in turn to maintain the visual memory (VM) after a fixation occurs illustrated in
Figure 2B). If a new landmark $e_i$ is detected (i.e. it appears in the robot’s viewpoint), its pair $(e_i, \text{pos}(e_i))$ is added into $\mathcal{V,M}$, where its particle filter $G_i$ is initialised by uniformly distributing the particles across the table. If a landmark is not seen for 1.5 seconds, Gaussian noise with zero mean and 1.0 cm of standard deviation is added to its current estimate. This models temporal decline (forgetting) in visual working memory. Finally, if a landmark $e_i$ already in $\mathcal{V,M}$ is detected its corresponding particle filter $G_i$ is updated by incorporating the new observation (i.e. the location just estimated), denoted by $\omega$. This update follows the resampling process of particle filters [Thrun et al. 2008], which makes the particles converge as more observations are made. This is done by first calculating the weight of each particle:

$$\text{weight}(g^j) = p(\omega|g^j),$$

that represents the probability of $\omega$ given the particle $g^j$. Once all weights 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 high weight are more likely to be chosen for the new set). As it is possible that some particles are selected more than once, some noise is added to each particle according to the $\mathcal{OM}$. In fact, the addition of this noise produces a smoothly varying spatial acuity (as mentioned in Section 2.1).

By accessing the $\mathcal{V,M}$, the robot employs the estimates about the location of landmarks in order to decide what manipulation actions to perform (Figure 2A).

3.4.2 Manipulation Action Selection. The robot selects 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, their actions are selected in parallel using the Q-MDP algorithm [Cassandra 1998]:

$$a_{ms} = \underset{a \in \mathcal{A}_{ms}}{\arg\max} \frac{1}{J} \sum_{g^j \in \mathcal{G}_i} \text{value}(g^j, a)$$

(2)

Where $g^j$ is a particle from the set $G_i$ of the landmark $e_i$, and $J$ is the number of particles. Then, $\text{value}(g^j, a)$ is the value of performing action $a$ on landmark $e_i$, assuming $g^j$ is the landmark’s location. An action $a$ is determined to: i) 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); or ii) fail, if the offset is greater than the threshold. If an action succeeds, $\text{value}(g^j, a)$ takes the Q-value of $Q_{ms}(s, a)$ (from the learnt policy $\pi_{ms}$), if it fails, it takes the Q-value of the second best action 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. For instance, Figure 4A shows the probability of success/failure for action $\text{Grasp}$ (the same applies for reaching). The area where grasping succeeds is determined by a threshold (red circle). If the number of particles inside this area is big then each particle will get the maximum value ($Q_{ms}(s, a)$) and the probability of success will be high. The number of particles lying inside the grasping area is determined by the spread of the particles (pos($e_i$)) (green ellipsoid). Note that for a manipulation action to succeed, the uncertainty should be minimised. This is achieved by looking at the landmarks, as determined by the gaze control models which we now describe in turn.

3.5 Models of Gaze Control

The selection and 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 1.1 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 (summarised in Section A:11).
2) that aim to select fixation locations, where a fixation location corresponds to a specific landmark listed in \( VM \). Because our models predict the benefit of looking at some landmark, which includes: i) the selection, ii) saccade, and iii) analysis of that fixation location; then we will use the term *perceptual actions*. Thus, the aim is to select a perceptual action \( p_i \in P \), associated to landmark \( e_i \).

### 3.5.1 Gaze Control based on Uncertainty Reduction (Unc)

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 \( VM \), 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 (denoted as \( p_{ms} \)):

\[
p_{ms} = \arg \max_{p_i \in P} \left\{ \frac{1}{O} \sum_{o \in \Omega_i} \text{dist}(G_i) - \text{dist}(G_i^{p_i}, \omega_o) \right\}
\]  \hspace{1cm} (3)

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 (\( \text{mean}(G_i) \)), where \( G_i \) is the particle set representing the location uncertainty of that landmark. For each perceptual action \( p_i \), imaginary observations \( \omega_o \) (i.e. imaginary landmark locations) are sampled using the observation model \( OM \), where \( \Omega_i \) is the set of observations produced by \( p_i \), and \( O \) is the number of observations. Thus, we first calculate the current spread in uncertainty using function \( \text{dist}(G_i) \). The spread is the average Euclidean distance between each particle \( g^\ell \in G_i \) and the mean of the cloud (\( \text{mean}(G_i) \)). Then, each observation \( \omega_o \) is used to perform a particle filter update (see Section 3.4.1) and the predicted spread in uncertainty is calculated by \( \text{dist}(G_i^{p_i}, \omega_o) \), assuming \( p_i \) is taken. 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 motor system that maximises the reduction in uncertainty. Then this motor system executes its selected perceptual action \( p_{ms} \). As mentioned in Section 2.2, the main drawback of this model is that it favours perceptual actions (i.e. fixations) that obtain new information that might not be relevant to the current needs of the task, because task rewards are not considered.

### 3.5.2 Gaze Control based on Rewards and Uncertainty (RU)

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 that would be obtained if each landmark, for each manipulation motor system, is fixated the next time step. Thus, for each motor system \( ms \), we calculate the predicted value of each manipulation action (denoted as \( V_{ms}^{p_i} \)) assuming perceptual action \( p_i \) is taken:

\[
V_{ms}^{p_i} = \frac{1}{O} \sum_{o \in \Omega_i} \max_{a \in A_{ms}} \left( \frac{1}{J} \sum_{g^j \omega_o \in G_i^{a_o}} \text{value}(g^j \omega_o, a) \right)
\]  \hspace{1cm} (4)

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 (\( \text{mean}(G_i) \)) of landmark \( e_i \). For a perceptual action \( p_i \), imaginary observations \( \omega_o \) (i.e. imaginary landmark locations) are sampled using the \( OM \), where \( \Omega_i \) is the set of observations produced by perceptual action \( p_i \), and \( O \) is the number of observations. Basically, for each observation \( \omega_o \) we verify which manipulation action \( a \) produces the maximum value the next time step. This is similar to Eq. 2 but here we need to update the particle filter with the imaginary observation \( \omega_o \). Thus, \( g^j \omega_o \) is a particle from the set \( G_i^{o_o} \), and \( J \) is the number of particles. \( \text{value}(g^j \omega_o, a) \) is the value of performing action \( a \) on landmark \( e_j \), assuming \( g^j \omega_o \) is the predicted landmark’s location.
As explained in Section 3.4.2, if action $a$ succeeds (if $g^j_{\omega}$ lies within the threshold area) then it takes the Q-value of $Q_{ms}(s,a)$. If the action 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 $J$. Now, for each manipulation motor system $ms$, we calculate the values $V^p_{ms}$ of each perceptual action associated to that motor system, and select the one with the maximum value (denoted as $M^p_{ms}$):

$$M^p_{ms} = \max_{p \in P} \{ V^p_{ms} \}$$

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

$$p_{ms} = \arg \max_{p \in P} \left\{ \max_{ms \in MS} \{ M^p_{ms} \} \right\}$$

The main disadvantage of this model is its tendency to look at landmarks with low location uncertainty and high predicted value, which will not typically provide much new task information.

### 3.5.3 Gaze Control based on Rewards, Uncertainty and Gain (RUG)

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 $V^a_{ms}$ of each perceptual action $a$, using Eq. 4. Second, Eq. 5 selects the value $M^p_{ms}$ for each motor system $ms$. However, instead of using $M^p_{ms}$ for gaze allocation directly as the RU model, here we calculate the gain that each motor system $ms$ is expected to obtain if it is given access to perception:

$$gain^p_{ms} = M^p_{ms} - \max_{a \in A_{ms}} \{ V^a_{ms} \}$$

$M^p_{ms}$ is the maximum value of performing a particular action, for motor system $ms$, if perceptual action $p_i$ is taken (i.e. the predicted value one step ahead in time). $V^a_{ms}$ is the value of executing action $a$ in the current time step. The second term determines the current maximum value of performing some action, which essentially is calculated during the manipulation action selection using Eq. 2. 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 with the highest gain. Then this motor system executes its associated perceptual action, denoted as $p_{ms}$:

$$p_{ms} = \arg \max_{p \in P} \left\{ \max_{ms \in MS} \{ gain^p_{ms} \} \right\}$$

This model tries to overcome the weakness of the preceding model by fixating landmarks that provide high predicted value and relevant task information. The next section presents a series of experiments that characterise and compare all of our models of gaze control.

### 4. EXPERIMENTAL ANALYSIS FOR THE PICK & PLACE TASK

In this section we characterise the three gaze control models, described above, in terms of task performance by varying three environmental variables: i) reach/grasp sensitivity in the manipulation actions, ii) the level of observation noise and, iii) the camera’s field of view (FoV). This analysis makes use of the pick & place task (Section 1.2), that consists of picking up objects from the table top and then placing them inside one of two containers. Even though the aim is to make comparisons between
our three gaze control models, two more gaze schemes are presented that serve as a common baseline and provide a more general analysis [Nunez-Varela et al. 2012a]:

Random gaze control. Fixates randomly on landmarks from those available in visual memory ($\mathcal{V}_M$).

Round Robin gaze control. Loops through the list of landmarks in $\mathcal{V}_M$ fixating one by one.

Next, the results and analysis for each environmental variable are presented, where a total of 15 trials of 5 minutes each were performed for each gaze strategy. The error bars in all the graphs below represent the 95% confidence intervals.

4.1 Reach/Grasp Sensitivity

First we analyse the sensitivity in the manipulation actions. As explained in Section 3.4.2, the success of actions is determined by how many particles lie within the threshold area. This threshold defines the accuracy at which the robot should reach for and grasp objects. Figure 4A shows the probability of success for action Grasp. The area where grasping succeeds is determined by the threshold (red circle). If the proportion of particles inside this area is big then the probability of success is high. Notice that as the threshold area gets bigger, more particles will lie inside it, so even if the uncertainty about a landmark's location is high, it is likely that the robot will still succeed in performing the action. As the threshold gets smaller sensitivity to positional uncertainty rises and task performance decreases.

4.1.1 Results. For this experiment six threshold values were defined: 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0 cm. Figure 4B shows the average number of objects correctly placed in the containers for all five gaze strategies. The observation noise is set to 1.0, meaning that we use an unmodified version of the observation model, and the FoV is set to 60°x40°, the default FoV in the simulator. As expected, the task performance of all gaze strategies drops as the sensitivity increases (i.e. as the threshold gets smaller). The RUG gaze strategy outperforms all other schemes, except when sensitivity is 2.5 and 3.0 cm, where the Unc strategy performs better. The RU scheme performs better than the Unc for the values of 0.5 and 1.0 cm, but in general it is no better than Random and Round Robin. The lower graph of Figure 4B 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. In this case, RUG is more robust to those changes. 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 $p < .0001$. The two-tailed unpaired t-test was then used to compare the results of the RUG scheme against each strategy. 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, RUG is not statistically significant against the Unc scheme. Finally, for the values of 2.5 and 3.0 cm only against the RU scheme the differences are statistically significant at $p < .002$.

4.1.2 Analysis. First, recall that we have a smoothly varying spatial acuity, and all landmarks' locations inside the current's view point are updated during a fixation (Section 3.4.1). As the sensitivity increases the RUG strategy becomes relatively more preferable to the other schemes. For example, when the threshold value is small (0.5 or 1.0 cm), landmarks need to fall close to the camera’s centre to get better observations, in order to better reduce location uncertainty. However, as the threshold value increases more particles will lie inside the threshold area. This makes it more likely for manipulation actions to succeed even if the landmarks’ location uncertainty is high, and even if they are far from the camera’s centre. This is why the performance of Random and Round Robin is high for the threshold values of 2.5 and 3.0 cm. The Unc scheme gets more benefit from this situation, which suggests that only reducing positional uncertainty is enough when the requirements of the task are simplified. For the RU strategy, recall that it might prefer to fixate landmarks that do not provide new task information but provide high value, so it will tend to fixate a landmark more than needed.
Models of Gaze Control for Manipulation Tasks

A:15

Fig. 4. A. Representation of the probability of success for action *Grasp*. B. Results for reach/grasp sensitivity analysis, with observation noise = 1.0 and 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.

This behaviour is good for small threshold values, as more accuracy is required; but it is bad when the threshold increases, as it wastes time fixating landmarks from which no more relevant information can be extracted. In fact, for RU, the workload between arms fluctuates during the trial. The robot ends up working with only one arm for some periods of time. If the landmarks’ location uncertainty associated with an arm increases, then the expected value attached to those landmarks gets smaller. As the RU scheme fixates landmarks with high expected value, it will sometimes favour only one arm.

4.2 Observation Noise

This experiment refers to the observation model (OM) used to estimate the location of detected landmarks (Section 3.4.1). By scaling the observation noise the location uncertainty about landmarks also increases, making it harder for the robot to successfully manipulate them. As an example, Figure 5A shows one of the bivariate Gaussian densities from the OM scaled by different levels of noise. As the observation noise accrues it is expected that the task performance will drop.

4.2.1 Results. The OM is scaled by the noise factors: 1.0, 1.5, 2.0 and 2.5. Figure 5B 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 FoV is 60°x40°. As expected, the task performance of all gaze schemes decreases as observation noise accrues. In terms of task performance the RUG gaze strategy outperforms all other schemes, whilst the second best is the RU strategy. The lower graph of Figure 5B shows that, this time, the RU gaze strategy is the most robust to changes in observation noise, followed by the RUG gaze scheme. 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 compare the results of the RUG scheme against each strategy. All the differences are statistically significant at p < .0001.

4.2.2 Analysis. In terms of task performance the RUG gaze strategy is the best option. Interestingly, the RU scheme is the second best option. As pointed out before, this strategy tends to fixate landmarks more often than needed, which is good when the observation noise is large. This is why the RU strategy is more robust to observation noise than the other schemes. In contrast, the Unc strategy
Fig. 5. A. A distribution from the observation model scaled by noise factors. B. Results for observation noise analysis with sensitivity = 1.0 cm and 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.

has a similar task performance to Random and Round Robin. As the observation noise accrues, landmarks need to be fixated several times for a manipulation action to succeed. However, the Unc scheme is likely to select different fixation locations every time step.

4.3 Field of View

The field of view (FoV) refers to the camera’s angles that determine the size of the image obtained during a fixation. If the FoV narrows we would expect to see a reduction in task performance since there is less information in the view point. 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 6A shows an image captured by the right camera and how some landmarks (objects/containers) no longer appear in the image as the FoV decreases. In fact, as the FoV gets smaller it is less likely to find new objects in the current view point. Thus, the ability to find new objects is crucial for good performance, particularly when the FoV is (35°x25°) and (25°x20°). To this end, we have incorporated a visual search mechanism that aims to find new objects, interleaved with the fixation selection process. Thus, the gain of performing a visual search perceptual action is calculated for each manipulation motor system $ms$:

$$gain_{vs}^{ms} = P(obj_{new}|vm_{ms})P(obj_{seeing}|p_{vs}, obj_{new}) \left[ \max_{a \in A_{ms}} Q(ms,s',a) - \max_{a \in A_{ms}} Q(ms,s,a) \right]$$

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 ($VM$) reachable to motor system $ms$. $P(obj_{seeing}|p_{vs}, obj_{new})$ is the probability of seeing an object given that the visual search perceptual action $p_{vs}$ is executed and there is a new object on the table. Both probabilities were learnt off-line. The subtraction is between the value of performing a manipulation action in the state $s'$ that results after performing visual search, and the value in the current state $s$.

4.3.1 Results. Figure 6B presents the average number of objects correctly placed in the containers for all gaze strategies, where the reach/grasp sensitivity is 1.0 cm and the observation noise is 1.0. Task performance drops, as expected, for all gaze schemes as the FoV narrows. The RUG strategy
Models of Gaze Control for Manipulation Tasks

4.3.2 Analysis. The RUG strategy is again the best option in terms of task performance. However, the Round Robin scheme is slightly more robust to changes in the FoV. A reason for this is simply because the performance that the RUG 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. The RU and Unc schemes have a similar performance to Random and Round Robin.

5. EXPERIMENTAL ANALYSIS FOR JOHANSSON’S TASK

The previous section characterised and analysed all the gaze control strategies in terms of three environmental variables. This section aims to determine how well our models of gaze control fit behavioural human data. To do so we simulate the psychophysical task employed by Johansson et al. [2001] and compare with their results. Figure 7A shows the schematic representation of the task as presented in [Johansson et al. 2001]. Subjects have to reach for and grasp the bar, then 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. This task was designed to study and measure the spatio-temporal relation between gaze and actions. One of the main findings was that gaze fixations specify landmark positions to which the manipulation actions are subsequently directed. The task is divided into the action phases of: pre-reach, reach, load & lift, target switch, replace & unload, and reset (shown in Figure 7A). The fixational landmarks were defined by looking at the places where subjects fixated the most during the task. These landmarks are: the bar, the tip of the bar, the obstacle, the target and the support surface (the coloured areas in Figure 7A). For our simulation, the same action phases and landmarks are considered.
5.1 Simulating the Task

Following the same procedure as in Section 3, the task is modelled using a factorised discrete state representation. Notice that for this task only one arm is required, thus there are three state variables for \( S_{right\_arm} \): \( \text{armPosition} = \{ \text{onTable}, \text{onBar}, \text{onTarget}, \text{onPos1}, \text{onPos2}, \text{onPos3}, \text{onPos4}, \text{onPos5}, \text{onPos6}, \text{onSupport} \} \), \( \text{handStatus} = \{ \text{graspingBar, handEmpty} \} \), and \( \text{targetStatus} = \{ \text{targetTouched, targetUntouched} \} \) (Figure 7B shows the work space for the state variable armPosition). There are eleven actions for the right arm (\( A_{right\_arm} \)): \( \text{moveToBar}, \text{moveToTarget}, \text{moveToPos1}, \text{moveToPos2}, \text{moveToPos3}, \text{moveToPos4}, \text{moveToPos5}, \text{graspBar}, \text{releaseBar} \), and \( \text{noAction} \). The times used for each action are those defined in Section 3.1, where all \( \text{moveToX} \) actions have a mean and standard deviation of \((2.65,0.56)\) in seconds respectively. Then, learning occurs as described in Section 3.3 for one arm, assuming the locations of all landmarks are known to the robot.

5.2 Results

Figure 8A shows the results presented by Johansson et al. [2001]. The data was collected from 10 human subjects, each performing 4 trials. The graph presents the probability of fixating a given landmark during each action phase (the colours correspond to the areas shown in Figure 7A). 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 duration of the task was 7.8 seconds. We defined the duration of the action phases following the same procedure as Johansson et al. A total of 20 trials were performed for each of our gaze control schemes, where the reach/grasp sensitivity was set to 1.0 cm, the observation noise to 1.0, and the FoV to 60°x40°. Figure 8 compares the results of all the gaze strategies. Note how the RUG (B) and the RU (C) schemes produce the same relative ordering of gaze and actions as the human data (A); with a median of 8 and 8.5 fixations per trial respectively, and a median duration of the task of 42.7 and 42.5 seconds respectively. The rest of the gaze schemes do not reproduce the human data (D-F). A median of 22, 24 and 23.5 fixations per trial were made by the Unc (D), Random (E), and the Round Robin (F) gaze strategies respectively; whilst the corresponding median durations of the task were 59.5, 59.1 and 50 seconds respectively. To quantitatively compare our gaze schemes with the human data we employ the Levenshtein (or edit) distance [Levenshtein 1966], which measures how different two string sequences are by determining the number of edit operations to transform one string into the other. According to the human data (Figure 8A), the most likely fixation sequence is: \text{grasp site, bar tip, obstacle, target, obstacle, support surface}, resulting in the string “gbotos”. Based on this string, the Levenshtein distance of each trial is calculated, and the results for each gaze strategy are averaged. Values close to zero indicate similarity.

\(^5\)This graph shows all the results in a single plot and is taken from [Johansson and Flanagan 2009] where the task was revisited.
Models of Gaze Control for Manipulation Tasks

5.3 Analysis

First, the RUG and the RU gaze strategies (Figure 8B,C) produce the same gaze pattern because for this task, as only one arm is used, the gaze allocation problem does not exist. Thus, the behaviour of both gaze schemes is necessarily the same. The RUG (or RU) gaze scheme 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. However, the robot is slower in terms of saccade execution and processing time. The robot takes an average of 2 seconds to saccade, 0.65 seconds to decide where to the original sequence, which is the case for RUG and RU gaze strategies as shown in Figure 8G. The differences are statistically significant at p < .0001, using a one-way ANOVA test.
to look, and 0.01 to 0.07 seconds to update the particle filters (depending on the number of objects detected). The robot performed a median of 8 fixations per trial, compared to 16 made by humans; and the median duration of the task was 42.6 seconds for the robot, compared to 7.8 seconds in humans. The shape of the probabilities is also different in the human data and the RUG graphs (Figure 8A,B).

For 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 them. In our case, we know exactly where the robot has decided to look. This is why our graphs present a high fixation probability for some landmarks. Another difference is that the probabilities from the human data are smooth compared to our graphs. This is because human subjects performed more fixations per trial than the robot, thus in the model fewer data points are available. The action phases in the human data are divided in bins of 100 msec, compared to bins of 2 to 3 seconds in our graphs. The Unc gaze scheme fixates landmarks located far from each other. This can be seen during the Reach action phase (Figure 8D), where the target zone is fixated and then gaze moves to the tip of the bar, which is located on the other side of the workspace. On the other hand, the probability of fixating the obstacle is small, because it is in the centre of the workspace constantly appearing in the FoV and updated in almost every fixation. In the case of the Random gaze scheme (E), the likelihood of fixating some landmark has almost uniform probability. Finally, Round Robin (F) 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.

6. CONCLUSIONS

This work addressed two questions: i) what mechanisms a rational decision maker could employ to select a gaze location given limited information and limited computation time, ii) how humans select the next fixation location. Previous work has suggested that human eye movement behaviour is consistent with decision making mechanisms for fixation selection that are Bayes’ rational [Najemnik and Geisler 2005], or that try to maximise reward [Navalpakkam et al. 2010]. The aim of our work is to investigate these claims further, by examining in detail the formulation and behaviour of three one-step look ahead models of gaze control that deal with the problem of fixation selection, during the performance of manipulation tasks. Our first model chooses the fixation location that maximises the reduction of location uncertainty (Unc). Our second model incorporates task rewards and selects the fixation that maximises the value of performing an action by reducing location uncertainty (RU). Our third model is similar to RU but it maximises the gain that results when a motor system is given access to perception (RUG). A pick & place task was used to characterise our models in terms of task performance by varying three environmental variables: reach/grasp sensitivity, observation noise, and field of view. The RUG gaze scheme is, in general, the best option in terms of task performance and robustness to changes for all environmental variables. A second task, based on a psychophysical experiment devised by Johansson et al. [2001], showed the goodness of fit of our gaze control models to existing human data. Only the RUG and RU schemes reproduced the same relative ordering of gaze and actions as the human subjects. Because the behaviour of the RUG and RU schemes is necessarily the same for this task, we cannot decide which of these two models best fits the human data, so further experiments are required to differentiate between them. Possible experiments to distinguish RU and RUG rely on presenting targets where target rewards vary, and high reward targets have low positional uncertainty (HRLU) and low reward targets high uncertainty (LRHU). In this case RU will favour the HRLU targets and RUG the LRHU targets. Finally, the results demonstrate that reasoning about task rewards is critical for the control of gaze, since the Unc scheme behaved, in general, as Random or Round Robin. Still, further experiments (for humans and machines) should be devised to analyse the precise role of rewards and information uncertainty during the performance of tasks.

ACKNOWLEDGMENTS
We are grateful for the support of CONACYT-Mexico (Reg.179604), UKIERI (SA06-0031), CogX (FP7-ICT-215181), GeRT (FP7-ICT-248273), PaCMan (FP7-ICT-600918), and the invaluable help of Dr. B. Ravindran and the reviewers.

REFERENCES

Received Month YYYY; revised Month YYYY; accepted Month YYYY.