Robotics and Biomimetics
a SpringerOpen Journal

RESEARCH Open Access

Learning search polices from humans in a partially observable context

Guillaume de Chambrier* and Aude Billard

Abstract

Decision making and planning for which the state information is only partially available is a problem faced by all forms of intelligent entities they being either virtual, synthetic or biological. The standard approach to mathematically solve such a decisional problem is to formulate it as a partially observable decision process (POMDP) and apply the same optimisation techniques used in the Markov decision process (MDP). However, applying naively the same methodology to solve MDPs as with POMDPs makes the problem computationally intractable. To address this problem, we take a programming by demonstration approach to provide a solution to the POMDP in continuous state and action space. In this work, we model the decision making process followed by humans when searching blindly for an object on a table. We show that by representing the belief of the human's position in the environment by a particle filter (PF) and learning a mapping from this belief to their end effector velocities with a Gaussian mixture model (GMM), we can model the human's search process and reproduce it for any agent. We further categorize the type of behaviours demonstrated by humans as being either risk-prone or risk-averse and find that more than 70% of the human searches were considered to be risk-averse. We contrast the performance of this human-inspired search model with respect to greedy and coastal navigation search methods. Our evaluation metric is the distance taken to reach the goal and how each method minimises the uncertainty. We further analyse the control policy of the coastal navigation and GMM search models and argue that taking into account uncertainty is more efficient with respect to distance travelled to reach the goal.

Keywords: Belief space planning; Imitation learning; Partially observable environment; Search strategies in humans

Background

Acting under partial observability

Learning controllers or policies to act within a context where the state space is partially observable is of high relevance to all real robotic applications. Resulting from limited and inaccurate perceptual information, often only an approximation of the environment is available at any given time. If this inherent uncertainty is not taken into account during planning or control, there is a non-negligible risk of missing goals, getting lost and wasting valuable resources.

A common approach is to formulate the uncertainty present in both action and state as a partially observable Markov decision process (POMDP). POMDPs are an extensive area of research in the operational research, planning and decision theory community [1,2]. The emphasis is to be able to act optimally with respect to an objective criteria when the state information is only partially available due to perceptual limitations and actions that are non-deterministic (stochastic).

The first approach to solving a POMDP is to apply value iteration (VI) [3] over the belief space (space of all possible probability distributions over the state space) as if we were solving for a standard Markov decision process (MDP). If the states, actions and observations are all discrete and the cost (or reward) function which encodes the task is the expected reward, then the overall value function is a convex combination of linear functions. In this setting, an exact solution exists [4], p. 513; however, the time and space complexity of VI in this context grows exponentially.

A popular approach to find a tractable solution to a POMDP is to reduce the size of the belief space by approximating it as a set of discrete reachable beliefs and then perform VI in this reduced space. Such methods fall under the category of point-based value iteration (PBVI) [5]. Most research has focused on determining the best set of

*Correspondence: guillaume.dechambrier@epfl.ch
LASA, EPFL, Route Cantonale, 1015 Lausanne, Switzerland

belief points [6-8] to be evaluated in VI. These methods rely on exploratory/search heuristics to discover a sufficient set of probability densities or sample points to be able to construct a sufficiently accurate approximation of the belief space such that an optimal policy can be found (see [9] for a detailed review on PBVI algorithms).

Other approaches are based on compressing the belief to sufficient statistics (mean and entropy) as in [10] and thereafter to perform VI in this augmented state space. The drawback with these methods so far is that they cannot deal with both continuous state and action space (we do not consider macro/parametrised actions to be a true solution for the continuous domain). The noticeable exception is Monte Carlo POMDP [11] which represents the belief of the position of a robot by a particle filter. However, the value function is difficult to compute and requires storing belief instantiations for evaluating new unseen beliefs. The major drawback of all these approaches lies with the exploration problem which becomes infeasible as the number of states and actions increase.

Decision theoretic-based approaches have also been applied. Notable examples are [12,13] where a decision tree graph is constructed with nodes representing beliefs (different realizations of a probability density function over the state space) and edges being actions (discrete). The actions themselves are typically macro-actions comprised of predefined start and end conditions. A planner (A* search) is used to find the appropriate set of actions to take, which follows a heuristic to find a trade-off between reducing the uncertainty and achieving the goal. If a large discrepancy exists between the estimated state and actual state, a new policy has to be re-planned. The shortcomings of these methods lie with the computational cost of constructing the search tree with particle filters (PF) for the belief nodes and the design of macro-actions. The responsiveness of these systems are bound to the computational cost and frequency of the re-planning step.

Programming by demonstration and uncertainty

Programming by demonstration (PbD) is advantageous in this context since it removes the need to perform the time-consuming exploration of the state-action tree to discover an optimal policy and does not rely on any exploration heuristics to gather a sufficient set of belief points (as in point-based value iteration methods). We expect humans to perform an informed search. In contrast to stochastic sampling methods, humans utilise past experience to evaluate the costs of their actions in the future and to guide their search. This foresight and experience are implicitly encoded in the parameters of the model we learn from the demonstrated searches.

PbD has a long history in the autonomous navigation community. In [14], behaviour primitives of the PHOENIX robot control architecture are incrementally learned from demonstrations. Two types of behaviour namely *reactive* and *history-dependent* are learned and are encoded by radial basis functions. The uncertainty is implicitly handled by directly learning the mapping between stimulus and response. In [15], the parameters of a controller which performs obstacle avoidance are learned from human demonstrations. The uncertainty is inherently handled by learning directly the relation between sensor input and control output. In [16], the objective function of a path planner is learned from human demonstrations. The objective function is a weighted sum of features corresponding to raw sensor measurements. This is another example where the partial information of the state is taken into account at the perception-action level, with the difference that instead of a policy being learned the objective function from which it is generated is learned. In [17], the authors learn how to combine low-level pre-acquired action primitives to achieve more complex tasks from human demonstrations, but they do not consider the effect of uncertainty.

Much work has been undertaken in learning reactive behaviour, history-dependent behaviour and combining multiple behaviour primitives to achieve complex behaviour. However, very few have studied the effect of uncertainty in the decision process and do not consider it during the learning or assume that it is implicitly handled. A noticeable exception is [18], in which a human expert guides the exploration of a robot in an indoor environment. The high-level actions (*explore*, *loop closure*, *reach goal*) taken by the human are recorded along with three different features related to the uncertainty in the map. Using SVM classification, a model is learned which indicates which type of action to take given a particular set of features. The difference with our approach is that we perform the learning in continuous action space at trajectory level and multiple actions are possible given the same state, which cannot be handled by a classifier.

Human beliefs

A crucial aspect of our work is to be able to infer the human's location belief whilst he is searching. The work on modelling human beliefs and intentions [19,20] has been undertaken in cognitive science. Human mind attributes, such as beliefs, desires and intentions, are not directly observable. They have to be inferred from actions. In [21], the authors present a Bayesian framework for modelling the way humans reason about and predict actions of an intentional agent. The comparison between the model and humans' predictions yielded similar inference capabilities when asked to infer the intentions of an agent in a 2D world. This provided evidence supporting the hypothesis that humans integrate information using Bayes' rule. Further, in [19], a similar experiment was performed in which the inference capabilities of humans, with regard to both belief and desire of an agent, were comparable to those of their Bayesian model. Our work makes the similar hypothesis that humans integrate information in a Bayesian way, however in the continuous domain. We infer the belief humans have of their location in the world during a search task.

As in our previous work [22], we learn a generative model of the human's search behaviour in the task of finding an object on a table. We compliment this work with four additional components, namely (1) an analysis of the different types of exhibited behaviour by the human demonstrators, a learned GMM model and two other search algorithms (greedy and coastal navigation), (2) a comparison between the human learned controller (GMM) and a coastal navigation search policy in addition to greedy and hybrid controllers which have already been discussed in our previous work, (3) an analysis of variance (ANOVA) to ensure that the search experiments were statistically different and a report on the distance taken to reach the goal and (4) a comparison of the policy generated by the GMM controller and the coastal navigation algorithm, with an emphasis of the role of the uncertainty.

Methods

Research design and methodology

In this work, we consider a task in which both a robot and a human must search for an object on a table whilst deprived of vision and hearing. The robot and the human have prior knowledge of the environmental setup making this a specific search problem with no required mapping of the environment, also known as active localisation. In Figure 1, a human has his sense of vision and hearing impeded, making the perception of the environment partially observable and only leaving the sense of touch available for solving the task. Before each demonstration, the human volunteer is disoriented. His transitional position is varied with respect to the table although his heading remains the same (facing the table) leaving no uncertainty on his orientation. The disorientation of the human subject is to ensure that his believed location is uniform. At the first time step, the human's state of mind can be considered observable. All proceeding beliefs can then be recursively estimated from the initial belief. The hearing sense is also impeded since it can facilitate localisation when no visual information is available, and the robot has no equivalent giving an unfair advantage to the human. By impeding hearing, we align the perception correspondence between the human and robot.

It is non-trivial to have a robot learn the behaviour exhibited by humans performing this task. As we cannot encapsulate the true complexity of human thinking, we take a simplistic approach and model the human's state through two variables, namely the human's uncertainty about his current location and the human's belief of his position. The various strategies adopted by humans are modelled by building a mapping from the state variables to actions, which are the motion of the human arm. Aside from the problem of correctly approximating the belief and its evolution over time, the model needs to take into consideration that people behave very differently given the same situation. As a result, it is not just a single strategy that will be transferred but rather a mixture of strategies. While this will provide the robot with a rich portfolio of search strategies, appropriate methods must be developed to encode, at times, these contradictory strategies. This

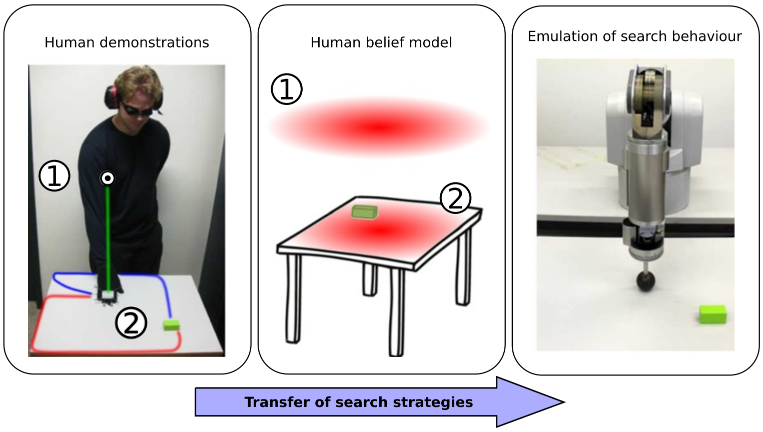

Figure 1 Experimental setup. *Left*: A human demonstrator searching for the green wooden block on the table given that both his hearing and vision senses have been impeded. He starts (hand) at the white spot near position (1). The red and blue trajectories are examples of possible searches. *Middle*: Inferred belief the human might have with respect to his position. If the human always starts at (1) and his belief is known, all following beliefs (2) can be inferred from Bayes' rule. *Right*: WAM Robot 7 DOF reproduces the search strategies demonstrated by humans to find the object.

leads to the main scientific question we seek to address in this work:

- Do humans exhibit particular search strategies, and if so, is it feasible to learn them?
- How well does a statistical controller learned from human demonstrations perform with respect to approaches which do not take into account the uncertainty directly?

Experimental setup

In the experimental setup, a group of 15 human volunteers were asked to search for a wooden green block located at a fixed position on a bare table (see Figure 2, *top left*). Each participant repeated the experiment ten times from each of four mean starting points with an associated small variance. The starting positions were given with respect to the location of the human's hand (all participants were right-handed). The humans were always facing the table with their right arm stretched out in front of them. The position of their hand was then either in front, to the left, to the right or in contact with the table itself.

As covered in the 'Background' section, previous work has taken a probabilistic Bayesian approach to model the beliefs and intent of humans. A key finding was that humans update their beliefs using Bayes' rule (shown so far in the discrete case). We make a similar assumption and represent the human's location belief (where he thinks he is) by a particle filter which is a point mass representation of a probability density function. There is no way of knowing the human's belief. We make the critical assumption that the belief is observable in the first time step of the search, and all following beliefs are assumed correct through applying Bayes integration. The belief is always initialized to be uniformly distributed on top of the table

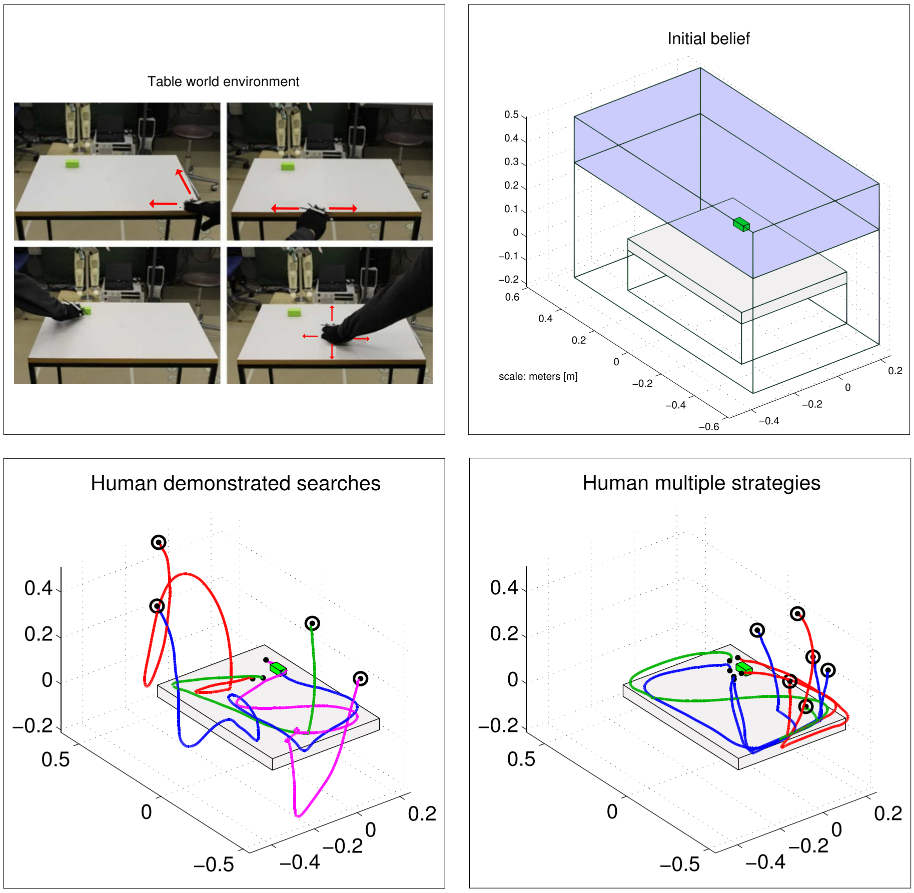

Figure 2 Experimental setup. *Top left*: A participant is trying to locate the green wooden block on the table given that both vision and hearing senses have been inhibited; the location of his hand is being tracked by the OptiTrack® system (NaturalPoint, Inc., Corvallis, OR, USA). *Top right:* Initial distribution of the uncertainty or belief we assume the human has with respect to his position. *Bottom right:* Set of recorded searches. The trajectories are with respect to the hand. *Bottom right:* Trajectories starting from the same area but have different search patterns. The red trajectories all navigate to the goal via the top right corner as opposed to the blue which go by the bottom left and right corners. Among these two groups, there are trajectories which seem to minimize the distance taken to reach the goal as opposed to some which seek to stay close to the edge and corners.

(see Figure 2, *top right*), and the starting position of the human's hand is always in this area.

Before each trial, the participant was told that he/she would always be facing the same direction with respect to the table (so always facing the goal, like in the case of a door), but his/her transitional starting position would vary. For instance, the table might not be always directly in front of the person and his/her distance to the edge or corner could be varied. In Figure 2 (*bottom left*), we illustrate four representative recorded searches, whilst in the *bottom right*, we illustrate a set trajectories which all started from the same region. One interesting aspect is the diversity present, demonstrating clearly that humans behave differently given the same situation.

Formulation

In the standard PbD formulation of this problem, a parametrised function is learned, mapping from state, x_t, which denotes the current position of the demonstrator's hand to $\dot{x}_t$, the hand's displacement. In our case, since the environment is partially observable, we have a belief or probability density function, $p(x_t|z_{0:t})$, which is conditioned on all sensing information, z (the subscript, $0{:}t$, indicates the time slice which ranges from $t = 0$ to the current time $t = t$) over the state space at any given point in time. We seek to learn this mapping, $f : p(x_t|z_{0:t}) \mapsto \dot{x}$, from demonstrations. During each demonstration, we record a set of variables consisting of the following:

1. $\dot{x}_t \in \mathbb{R}^3$, velocity of the hand in Cartesian space, which is normalised.
2. $\hat{x}_t = \arg\max_{x_t} p(x_t|z_{0:t})$, the most likely position of the end effector or believed position.
3. $U \in \mathbb{R}$, the level of uncertainty which is the entropy of the belief: $H\left(p(x_t|z_{0:t})\right)$.

A statistical controller was learned from a data set of triples $\{(x, \hat{x}, U)\}$, and a desired direction (normalised velocity) was obtained from conditioning on the belief and uncertainty.

Having described the experiment, we proceed to give an in-depth description of the mathematical representation of the belief, sensing and motion models and the uncertainty.

Belief model

A human's belief of his location in an environment can be multi-modal or uni-modal, Gaussian or non-Gaussian and may change from one distribution to another. We chose a particle filter to be able to represent such a wide range of probability distributions. A particle filter is a Bayesian probabilistic method which recursively integrates dynamics and sensing to estimate a posterior from a prior probability density. The particle filter has two elements. The first estimates a distribution over the possible next state given dynamics, and the second corrects it through integrating sensing. Given a *motion model* $p(x_t|x_{t-1}, \dot{x}_t)$ and a *sensing model* $p(z_t|x_t)$, we recursively apply a prediction phase where we incorporate motion to update the state and an update phase where the sensing data is used to compute the state's posterior distribution. The two steps are depicted below:

$$p\left(x_t|z_{0:t-1}\right) = \int p\left(x_t|x_{t-1}, \dot{x}_t\right)\, p\left(x_{t-1}|z_{0:t-1}\right)\, dx_{t-1} \quad (1)$$

$$p\left(x_t|z_{0:t}\right) = \frac{p\left(z_t|x_t\right)\, p\left(x_t|z_{0:t-1}\right)}{p\left(z_t|z_{0:t-1}\right)} \quad (2)$$

The probability distribution over the state $p\left(x_t|z_{0:t}\right)$ is represented by a set of weighted particles which represent hypothetical locations of the end effector and their density which is proportional to the likelihood. The particular particle filter used was the *regularised sequential importance sampling* [23], p. 182. From the previous literature [19], it has been shown that there is a similarity between Bayes update rule and the way humans integrate information over time. Under this assumption, we hypothesise that if the initial belief of the human is known then the successive update steps of the particle filter should correspond to a good approximation of the next beliefs.

Sensing and motion model

Sensing model. The sensing model tells us the likelihood, $p\left(z_t|x_t\right)$, of a particular sensation z_t given a position $x_t \in \mathbb{R}^3$. In a human's case, the sensation of a curvature indicates the likelihood of being near an edge or a corner. However, the likelihood cannot be modelled through using the human's sensing information. Direct access to pressure, temperature and such salient information is not available. Real sensory information needs to be matched against virtual sensation at each hypothetical location x_t of a particle. Additionally, for the transfer of behaviour from human to robot to be successful, the robot should be able to perceive the same information as the human, given the same situation. An approximation of what a human or robot senses can be inferred, based on the end effector's distance to particular features in the environment. In our case, four main features are present, namely corners, edges, surfaces and an additional dummy feature defining no contact, air. The choice of these features is prior knowledge given to our system and not extracted through statistical analysis of recorded trajectories. The sensing vector is $z_t = \left[p_c, p_e, p_s, p_a\right]$, where p refers to probability and the subscript corresponds to the first letter of the feature it is associated with. In Equation 3, the sensing function, $h\left(x_t, x_c\right)$, returns the probability of sensing a corner, where $x_c \in \mathbb{R}^3$ is the Cartesian position of the corner which is the closest to x_t.

$$p_c = h\left(x_t, x_c; \beta\right) = \exp\left(-\left(\beta \cdot \|x_t - x_c\|\right)^2\right) \tag{3}$$

The exponential form of the function, h, allows the range of the sensor to be reduced. We set $\beta > 0$ such that any feature which is more than 1 cm away from the end effector or hand has a probability close to zero of being sensed. The same sensing function is repeated for all feature types.

The sensing model takes into account the inherent uncertainty of the sensing function (3) and gives the likelihood, $p\left(z_t|x_t\right)$, of a position. Since the range of sensing is extremely small and entries are probabilistic, we assume no noise in the sensor measurement. The likelihood of a hypothetical location, x_t, is related to Jensen-Shannon divergence (JSD), $p\left(z_t|x_t\right) = 1 - \text{JSD}\left(z_t||\hat{z}_t\right)$, between true sensing vector, z_t, obtained by the agent and that of the hypothetical sensation $\hat{z}_t$ generated at the location of a particle.

Motion model. The motion model is straightforward compared with the sensing model. In the robot's case, the Jacobian gives the next Cartesian position given the current joint angles and angular velocity of the robot's joints. From this, the motion model is given by $p\left(x_t|x_{t-1}, \dot{x}_t\right) = J(q)\dot{q} + \epsilon$ where q is the angular position of the robot's joints, $J(q)$ is the Jacobian and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is white noise. The robot's motion is very precise and its noise variance is very low. For humans, the motion model is the velocity of the hand movement provided by the tracking system.

Uncertainty

In a probability distribution framework, entropy is used to represent uncertainty. It is the expectation of a random variable's total amount of unpredictability. The higher the entropy, the more the uncertainty; likewise, the lower the entropy, the lesser the uncertainty. In our context, a set of weighted samples $\{w_i, x_i\}^{i=1 \ldots, N}$ replaces the true probability density function of the belief, $p_u\left(x_t|z_{0:t}\right)$. A reconstruction of the underlying probability density is achieved by fitting a Gaussian mixture model (GMM) (Equation 4) to the particles,

$$p_u\left(x_t|z_{0:t}\,; \{\pi, \mu, \Sigma\}\right) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}\left(x_t\,; \mu_k, \Sigma_k\right) \tag{4}$$

where K is the number of Gaussian components, the scalar π_k represents the weight associated to the the mixture component k (indicating the component's overall contribution to the distribution) and $\sum_{k=1}^{K} \pi_k = 1$. The parameters μ_k and Σ_k are the mean and covariance of the normal distribution k.

The main difficulty here is determining the number of parameters of the density function in a computationally efficient manner. We approach this problem by finding all the modes in the particle set via mean-shift hill climbing and set these as the means of the Gaussian functions. Their covariances are determined by maximising the likelihood of the density function via expectation-maximisation (EM).

Given the estimated density, we can compute the upper bound of the differential entropy [24], H, which is taken to be the uncertainty U,

$$\begin{aligned} &H\left(p_u\left(x_t\|z_{0:t}\,; \{\pi, \mu, \Sigma\}\right)\right) \\ &\quad = \sum_{k=1}^{K} \pi_k \left(-\log\left(\pi_k\right) + \frac{1}{2}\log\left((2\pi e)^D\left|\Sigma_k\right|\right)\right) \end{aligned} \tag{5}$$

where e is the base of the natural logarithm and D the dimension (being 3 in our case).

The reason for using the upper bound is that the exact differential entropy of a mixture of Gaussian functions has no analytical solution. When computing both the upper and lower bounds, it was found that the difference between the two was insignificant, making any bound a good approximation of the true entropy. The choice of the believed location of the robot/human end effector is taken to be the mean of the Gaussian function with the highest weighted π.

$$\hat{x}_t = \arg\max_{x_t} p_u\left(x_t|z_{0:t}\,; \{\pi, \mu, \Sigma\}\right) = \mu_{(k=\max(\pi))} \tag{6}$$

Figure 3 depicts different configurations of the modes (clusters) and believed position of the end effector (indicated by a yellow arrow).

Model of human search

During the experiments, the recorded trajectories show that different actions are present for the same belief and uncertainty making the data multi-modal (for a particular position and uncertainty, different velocities are present). That is, multiple actions are possible given a specific belief. This results in a one-to-many mapping which is not a valid function, eliminating any regression technique which directly learns a non-linear function. To accommodate this fact, we again made use of a GMM to model the human's demonstrated searches, $\{(x, \dot{x}, U)\}$. Using statistical models to encode control policies in robotics is quite common (see [25]).

By normalising the velocity, the amount of information to be learned was reduced. We also took into consideration that velocity is more specific to embodiment capabilities: the robot might not be able to reproduce safely some of the velocity profiles demonstrated.

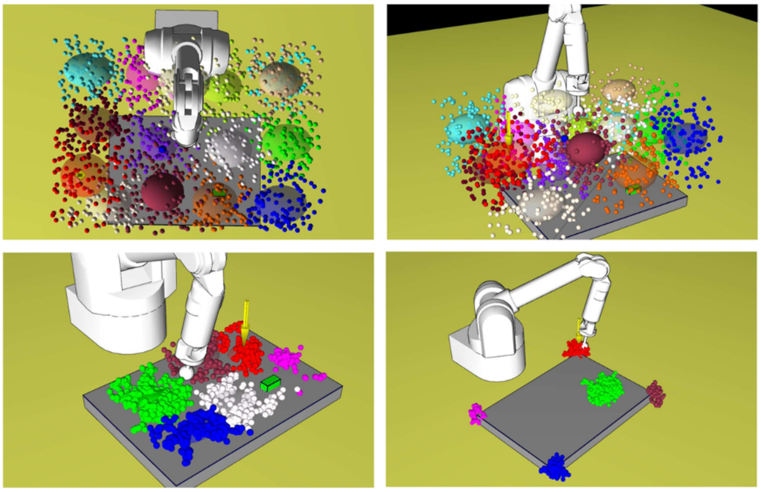

Figure 3 Representation of the estimated density function. *Top left and right*: Initial starting point. All Gaussian functions are uniformly distributed with uniform priors. The red cluster always has the highest likelihood which is taken to be the believed location of the robot's/human's end effector. *Bottom left*: Contact with the table has been established. The robot location differs from his belief. *Bottom right*: Contact has been made with a corner. The clusters reflect that the robot could be at any corner (note that weights are not depicted, only cluster assignment).

The training data set comprised a total of 20,000 triples $\left(\dot{x}, \hat{x}, U\right)$ from the 150 trajectories gathered from the demonstrators. The fitted GMM $p_s\left(\dot{x}, \hat{x}, U\right)$ had a total of seven dimensions, three for direction, three for position and one scalar for uncertainty. The definition of the GMM is presented in Equation 7:

$$p_s(\dot{x}, \hat{x}, U\,; \{\pi, \mu, \Sigma\}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\dot{x}, \hat{x}, U\,; \mu_k, \Sigma_k) \quad (7)$$

$$\mu_k = \begin{bmatrix} \mu_{\dot{x}} \\ \mu_{\hat{x}} \\ \mu_U \end{bmatrix} \Sigma_k = \begin{bmatrix} \Sigma_{\dot{x}\dot{x}} & \Sigma_{\dot{x}\hat{x}} & \Sigma_{\dot{x}U} \\ \Sigma_{\hat{x}\dot{x}} & \Sigma_{\hat{x}\hat{x}} & \Sigma_{\hat{x}U} \\ \Sigma_{U\dot{x}} & \Sigma_{U\hat{x}} & \Sigma_{UU} \end{bmatrix}$$

Given this generative representation of the humans' demonstrated searches, we proceeded to select the necessary parameters to correctly represent the data. This step is know as model selection, and we used Bayesian information criterion (BIC) to evaluate each set of parameters which were optimised via EM.

A total of 83 Gaussian functions were used in the final model, 67 for trajectories on the table and 15 for those in the air. In Figure 4 (*left*), we illustrate the model learned from human demonstrations where we plot the three-dimensional slice (the position) of the seven-dimensional GMM to give a sense of the size of the model.

Coastal navigation

Coastal navigation [26] is a path planning method in which the objective function (Equation 8) is composed of two terms.

$$f\left(x_{0:T}\right) = \sum_{t=0}^{T} \lambda_1 \cdot c\left(x_t\right) + \lambda_2 \cdot I\left(x_t\right) \quad (8)$$

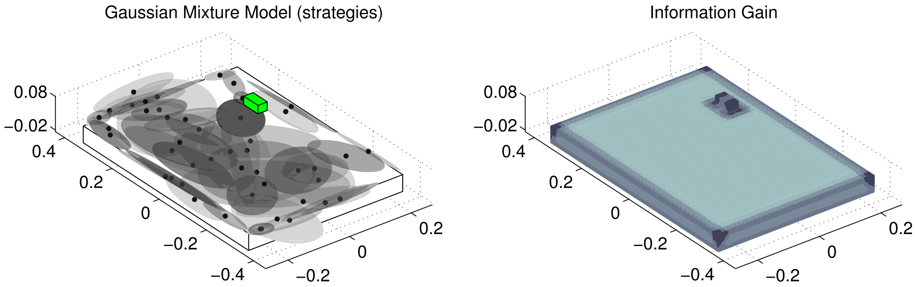

Figure 4 Resulting search GMM and information gain map. *Left*: Resulting search GMM. A total of 67 Gaussian mixture components are present. We note the many overlapping Gaussians: this results from the level of uncertainty over the different choices taken. For example, humans follow along the edge of the table in different directions and might leave the edge once they are confident with respect to their location. *Right*: Information gain map of the table environment. Dark regions indicate high information gain as oppose to lighter ones. Not surprisingly, the highest are the corners, followed by the edges.

The first term, $c\left(x_t\right)$, is the traditional 'cost to go' which penalizes every step taken so as to ensure that the optimal path is the shortest. The value was simply set to 1 for all discrete states in our case. The second term, $I\left(x_t\right)$, is the information gain of a state. The information gain, I, of a particular state is related to how much the entropy of a probability density function (pdf), being the location's uncertainty in our case, can be reduced. The two λ's are scalars which weigh the influence of each term.

In our table environment, we discretised the state space, $\mathbb{R}^3$, into bins so as to have a resolution of approximately, 1 cm^3, giving us a total of a 125,000 states. The action space was discretised to six actions, two for each dimension meaning that all motion is parallel to the axis. For each state, x_t, an $I\left(x_t\right)$ value is computed by evaluating Equation 9:

$$I\left(x_t\right) = \mathbb{E}_{p(z_t|x_t)}\left\{H(p_u(x_t|z_{0:t})\right\} - H\left(p_u\left(x_t|z_{0:t-1}\right)\right), \tag{9}$$

which is essentially the difference between the entropy of a prior pdf to that of a posterior pdf. We set our initial pdf to be uniformly distributed, and we computed the maximum likelihood sensation for each discrete state x_t which is akin to the expected sensation or assuming that there is no uncertainty in sensor measurement (an assumption often made throughout the literature to avoid carrying out the integral of the expectation in Equation 9). The result is the difference between the posterior pdf, given that the sensation occurred in x_t, and the prior pdf. The resulting cost map is illustrated in Figure 4. As expected, corners have the highest information gain followed by edges and surfaces. We do not show the values of the table since they provided much less information gain.

The optimization of the objective function is accomplished by running Dijkstra's algorithm. This algorithm, given a cost map, computes the shortest path to a specific target from all the states. This results in a policy.

Control

The standard approach to control with a GMM is to condition on the state $\hat{x}_t$ and U_t in our case and perform inference on resulting conditional GMM (Equation 10) which is a distribution over velocities or directions.

$$p_s\left(\dot{x}|\hat{x},U\right) = \sum_{k=1}^{K} \pi^k_{\dot{x}|\hat{x},U} \cdot \mathcal{N}\left(\dot{x}\,;\mu^k_{\dot{x}|\hat{x},U},\Sigma^k_{\dot{x}|\hat{x},U}\right) \tag{10}$$

The new distribution is of the dimension of the output variable, the velocity (dimension 3). The variable $\dot{x}$ in $\dot{x}|\hat{x},U$ indicates the predictor variable, and the variables $\hat{x},U$ have been conditioned. A common approach in statistical PbD methods using GMMs is to take the expectation of the conditional (known as Gaussian mixture regression) (Equation 11):

$$\dot{x} = \mathbb{E}\{p_s\left(\dot{x}|\hat{x},U\right)\} = \sum_{k=1}^{K} \pi^k_{\dot{x}|\hat{x},U} \cdot \mu^k_{\dot{x}|\hat{x},U} \tag{11}$$

The problem with this expectation approach is that it averages out opposing directions or strategies and may leave a net velocity of zero. One possibility would be to sample from the conditional; however, this can lead to non-smooth behaviour and flipping back and forth between modes resulting in no displacement. To maintain consistency between the choices and avoid random switching, we perform a weighted expectation on the means so that directions (modes) similar to the current direction of the end effector receive a higher weight than opposing directions. For every mixture component k, a weight α_k is computed based on the distance between the current direction and itself. If the current direction agrees with the mode, then the weight remains unchanged, but if it is in disagreement, a lower weight is calculated according to the equation below:

$$\alpha_k(\dot{x}) = \pi^k_{\dot{x}|\hat{x},U} \cdot \exp\left(-\cos^{-1}\left(<\dot{x},\mu^k_{\dot{x}|\hat{x},U}>\right)\right) \tag{12}$$

Gaussian mixture regression is then performed with the normalised weights α instead of π (the initial weight obtained when conditioning).

$$\dot{x} = \mathbb{E}_{\alpha}\left\{p_s\left(\dot{x}|\hat{x},U\right)\right\} = \sum_{k=1}^{K} \alpha_k(\dot{x})\,\mu^k_{\dot{x}|\hat{x},u} \tag{13}$$

The final output of Equation 13 gives the desired direction ($\dot{x}$ is re-normalised). In the case when the mode suddenly disappears (because of sudden change of the level of uncertainty caused by the appearance or disappearance of a feature), another present mode is selected at random. For example, when the robot has reached a corner, the level of uncertainty for this feature drops to zero. A new mode, and hence new direction of motion, will then be computed. However, this is not enough to be able to safely control the robot. One needs to control the amplitude of the velocity and ensure compliant control of the end effector when in contact with the table. This behaviour is not learned here, as this is specific to the embodiment of the robot and unrelated to the search strategy. The amplitude of the velocity is computed by a proportional controller based on the believed distance to the goal,

$$\nu = \max(\min(\beta_1, K_p(x_{\mathrm{g}} - \hat{x}), \beta_2) \tag{14}$$

where the β's are lower and upper amplitude limits, x_g is the position of the goal and K_p is the proportional gain which was tuned through trials.

As mentioned previously, compliance is the other important aspect when having the robot duplicate the search strategies. Collisions with the environment occur as a result of the uncertainty. To avoid risks of breaking the table or the robot sensors we have an impedance controller at the lowest level which outputs appropriate joint torques τ. The overall control loop is depicted in Figure 5.

Results and discussion

We analysed the types of behaviour present in the human demonstration as well as in four different search algorithms, namely greedy, GMM, hybrid and coastal. A qualitative analysis of the GMM search policy (namely the different modes/decisions present) is contrasted with the coastal navigation policy. Finally, we evaluated the performance of the searches, with respect to the distance taken to reach the goal and the uncertainty profiles towards the end of the searches in five different experiments (different types of initializations).

Search and behaviour analysis

For each method (greedy, GMM, hybrid, coastal), 70 searches were performed with all starting positions drawn from the uniform distribution depicted in Figure 2 (*top right*). Figure 6 gives the expected sensation $\mathbb{E}\{z\}$ and variance $\mathrm{Var}\{z\}$ for each trajectory with respect to the edge and corner of the table.

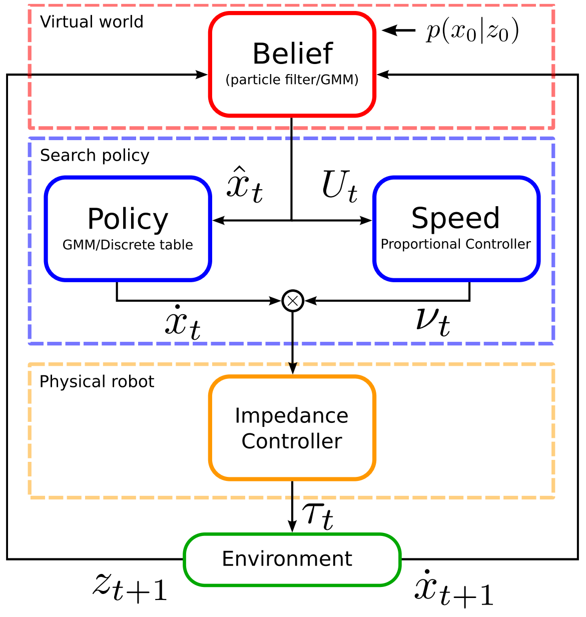

Figure 5 Overview of the decision loop. At the top, a strategy is chosen given an initial belief $p(x_0|z_0)$ of the location of the end effector (initially through sampling the conditional). A speed is applied to the given direction based on the believed distance to the goal. This velocity is passed onwards to a low-level impedance controller which sends out the required torques. The resulting sensation, encoded through the multinomial distribution over the environment features, and actual displacement are sent back to update the belief.

The selection of edges and corners as features as a means of classifying the type of behaviours present is not solely restricted to our search task. Salient landmarks will result in a high level of information gain, which is the case for the edge and corner (see Figure 4, *right*). Other tasks can use such features or variants in which the curvature is considered for representing the task space. These features are present in most settings, and high-level features can use these easily as their building blocks.

We note that the greedy search approach seeks to go directly to the goal without taking into account the uncertainty. The GMM models human search strategies. The hybrid is a combination of both the greedy and GMM method where once the uncertainty has been sufficiently minimised switches (threshold) to the greedy method for the rest of the search. The coastal navigation algorithm finds the optimal path to the goal based on an objective function which consists of a trade-off between the time taken to reach the goal and the minimisation of the uncertainty.

It can be seen that the human demonstrations have a much wider spread than those of the search algorithms. We suggest that this is due to human behaviours being optimal with respect to their own criteria as opposed to the algorithms which usually tend to only maximise a single objective function. The trajectories of the greedy and GMM methods represented by their expected features demonstrate two distinctive behaviours (in terms of expected sensation), risk-prone for the greedy and risk-averse for the GMM.

We take the assumption that greedy trajectories are risk-prone by nature; we performed a SVM classification on the greedy-GMM expected features (Figure 2, *left*) and used the result to construct a decision boundary as a means of classifying a trajectory as being either risk-prone or risk-averse. Table 1 (*first row*) shows that the GMM and human search trajectories are mostly risk-averse, and more surprisingly, the GMM seems to be more risk-averse than the GMM which seems counterintuitive. This is due to the choice of feature-based metric which is sensitive to the decision boundary. We use a second metric based on the information gain, which we call the risk factor, to classify trajectories as being either risk-prone or risk-averse.

The risk factor of each individual trajectory is inversely proportional to its accumulated information gain. Figure 7 (*left*), shows the kernel density estimation distribution of the risk for each search method. Two trajectories per search type corresponding to a supposed risk-prone and risk-averse search are plotted in the expected feature space in Figure 7 (*right*). As expected, risk-prone strategies for which the risk tends to 1 have a low expectation of sensing edges and corners and produce trajectories with a low information gain, whilst those with a high expectation of

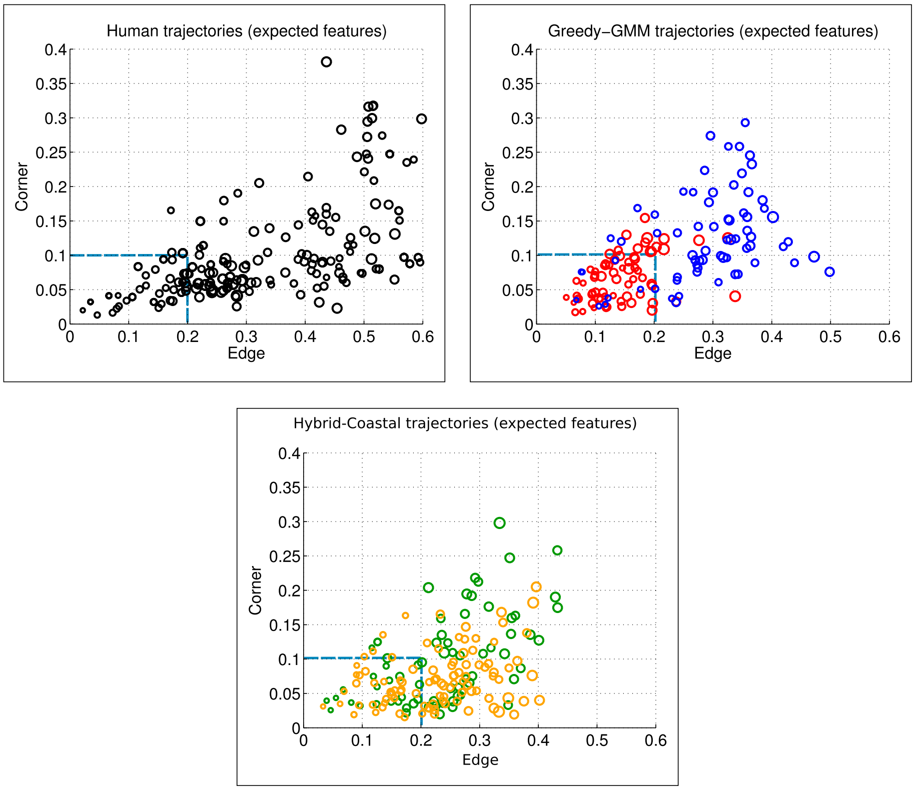

Figure 6 Expected sensation. Plots of the expected sensation of the edge and corner feature for all trajectories. The axes are associated with the sensor measurements; 0 means that the corresponding feature is not sensed and 1 the feature is fully sensed. A point in the plot summarise a whole trajectory by the mean and variance of the probability of sensing a corner or edge. The radius of the circles are proportional to the variance. The dotted blue rectangle represents the decision boundary for classifying a trajectory as being either risk-prone or risk-averse. A point which lies inside the rectangle is risk-prone. *Left*: Human trajectories demonstrate a wide variety of behaviours ranging from those remaining close to features to those preferring more risk. *Right*: Red points show greedy and blue points the GMM model. *Bottom*: Green circles are associated with the hybrid method, whilst orange are those of the coastal navigation method. The hybrid method is a skewed version of the GMM which tends towards risky behaviour and exhibits the same kind of behaviour as the coastal algorithm.

sensing features have a high information gain. Since the metric lies exclusively in the range [0,1], we set that every trajectory which has a risk factor lower than than 0.5 will be considered risk-averse whilst does above are risk-prone. Table 1 (*second row*) illustrates the riskiness of each search method. It is evident that humans are risk-averse in general followed by GMM which is a smoothing of the human data, then hybrid which as expected should be more risk-prone since it is a linear interpolation between the GMM and greedy search policies and finally coastal and greedy.

Table 1 Percentage of risk-prone trajectories based on two decision criteria

	Greedy	GMM	Hybrid	Coastal	Human
Risk-prone (f)	77%	11%	30%	46%	26%
Risk-prone (r)	78%	12%	24%	45%	7%

Two decision criteria: the feature (f) and the risk (r) (information gain) metrics.

Figure 8 (*top left and right*) shows risk-prone (red) and risk-averse (green) trajectories produced by human demonstrations and by the greedy search. Both these extremes correspond to our intuition that risk-averse trajectories tend to remain closer to features or areas of high information gain as oppose to risk-prone searches. However, to stress the case that humans have multiple search strategies present, we performed 40 GMM searches (model of the human behaviour) which all started under the same initial conditions (same belief distribution, true position and believed position). Figure 8 shows the resulting trajectories and expected features for each trajectory. It is clear that multiple searches occur which is reflected in the plot of the expected features. All of the search strategies generated by the GMM for this initial condition produced risk-averse trajectories.

We conclude that there is a strong inclination towards inferring that indeed multiple search strategies do arise

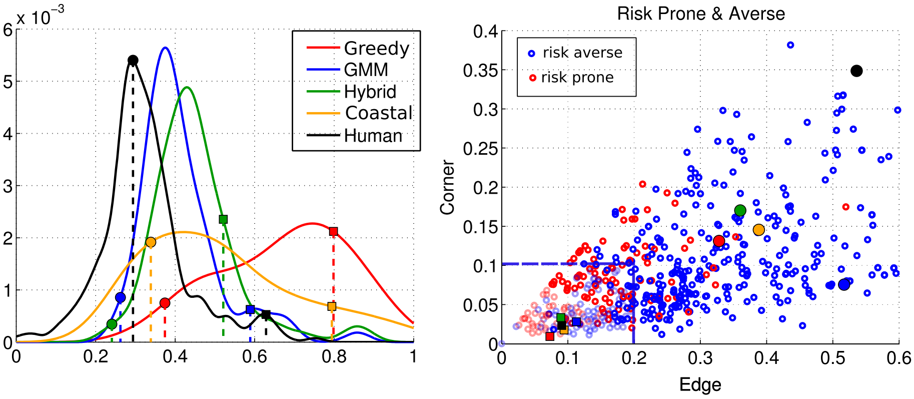

Figure 7 Risk of searches. Illustration of risk-prone and risk-averse searches in terms of a risk factor (*left*) and expected sensation (*right*). *Left*: Each trajectory was reduced to a single scalar, which we call the risk factor, quantizing the risk of a trajectory. The risk factor is inversely proportional to the sum of the information gain of a particular trajectory. The colour paired dots (risk averse) and squares (risk prone) represent trajectories which are plotted in Figure 8, to illustrate that these correspond to risk-averse and risk-prone searches. *Right*: Corresponding trajectories chosen in the risk factor space but represented in the feature space. As expected, trajectories with a high risk map to regions of low expected feature. However, the transition from the risk space to feature space is non-linear and will result in a different risk-level classification than the feature metric previously discussed.

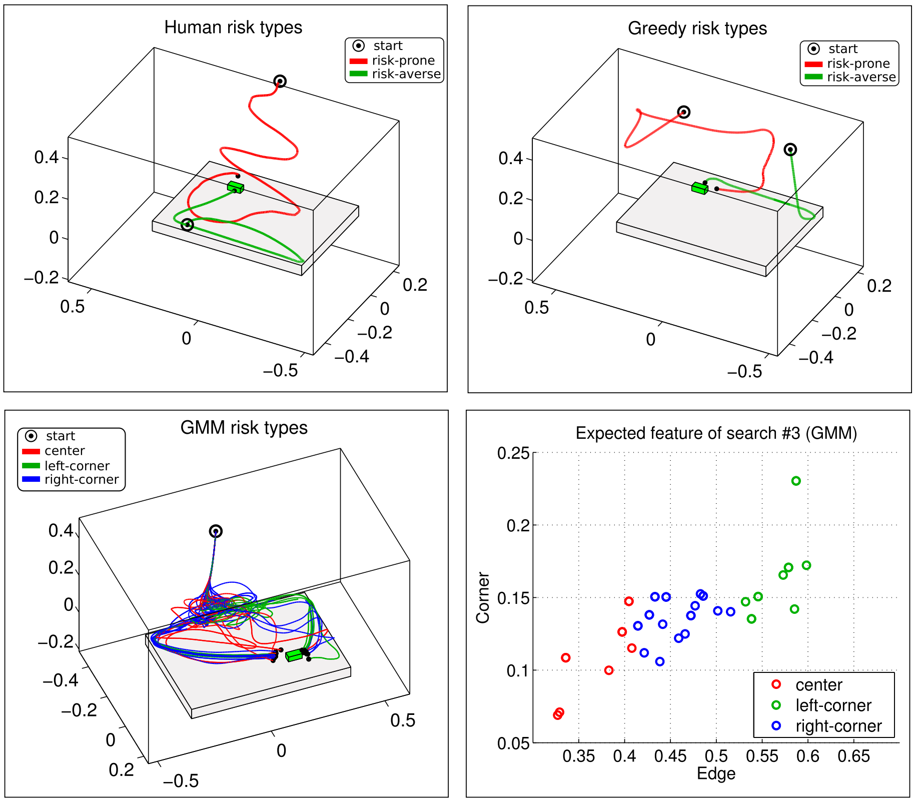

Figure 8 Risk-prone and risk-averse searches (red and green trajectories). *Top left*: Two human trajectories taken from the data shown in Figure 7. *Top right*: Two greedy trajectories. *Bottom left*: GMM trajectories, all starting from the same location; the colour coding is to illustrate the different policies which were encoded and emerge given the same initial conditions. *Bottom right*: Corresponding expected features of each trajectory. The colour coding matches the trajectories to the ‘GMM risk types’ sub-figure. All the searches which were generated by the GMM for this initialisation produced risk-averse searches (based on the feature metric discussed previously).

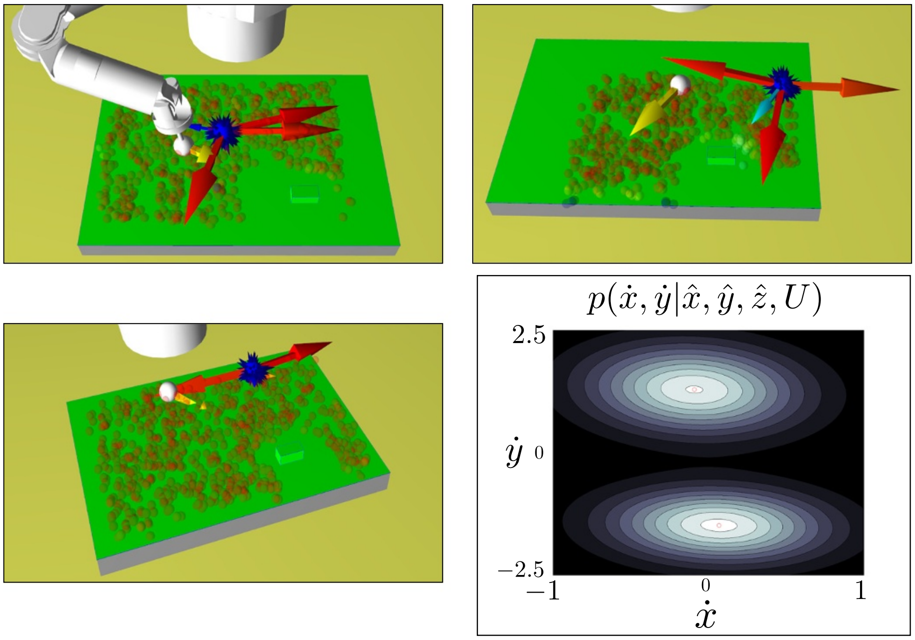

Figure 9 Illustration of three different types of modes present during the execution of the task. The robot is being controlled by the learned GMM model. The white ball represents the actual position of the robot's end effector. The blue ball represents the believed position of the robot's end effector and the robot is acting according to it. The blue ball arrows represent modes. Colours encode the mode's weights given by the priors π_k after conditioning (but not re-weighted as previously described). The spectrum ranges from red (high weight) to blue (low weight). *Top left*: Three modes are present, but two agree with each other. *Top right*: Three modes are again present indicating appropriate ways to reduce the uncertainty. *Lower left*: Two modes are in opposing directions. No flipping behaviour between modes occurs since preference is given to the modes pointing in the same direction as the robot's current trajectory. *Lower right*: GMM modes when conditioned on the state represented in the lower left figure. The two modes represent the possible directions (un-normalised).

in the human searches since they were extracted and encoded in the GMM model. From the risk distribution, humans have a tendency to be risk-averse.

GMM and coastal navigation policy analysis

We next illustrate some of the modes (action choices) present during simulation and evaluate their plausibility. Figure 9 shows that multiple decision points have been correctly embedded in the GMM model. All arrows (red) indicate directions that reduce the level of uncertainty.

Figure 10 depicts the vector fields of both coastal and GMM models, where as expected the coastal navigation trajectories tend to stay close to edges and corners until they are sufficiently close to the goal. This is achieved by weighting the information gain term $I(x_t)$ in the objective function sufficiently (λ_2). If $\lambda_2 = 0$, the coastal policy is the same greedy algorithm.

It can be further seen that when the uncertainty tends towards its maximum value ($U \rightarrow 1$), all behaviour tends to go towards the edges and corners. As the uncertainty reduces ($U \rightarrow 0$), the vector field tends directly towards the goal. However, even at a low level of uncertainty, the behaviour at the edges and corners remains multi-modal and tends to favour remaining close to the edges and corners. This is an advantage of the GMM model. If the uncertainty has been sufficiently reduced and the true position of the end effector or hand is not near an edge, the policy dictates to go straight to the goal. This is not the case for the coastal algorithm which ignores the uncertainty and strives to remain in the proximity of corners and edges until sufficiently close. This approach could potentially lead to unnecessary travel cost which could otherwise have been avoided.

Time efficiency and uncertainty

We seek to distinguish the most efficient method in terms of two metrics, the distance taken to reach the goal and the level of uncertainty upon arriving at the goal. We report results on five different search experiments in which we compare the greedy, GMM and coastal navigation algorithms. The hybrid was not fully considered since it is a heuristic combination of the greedy and GMM methods.

In the first experiment, the true and believed locations of the end effector were drawn uniformly from the original start distribution (Figure 2, *top right*) reflecting the

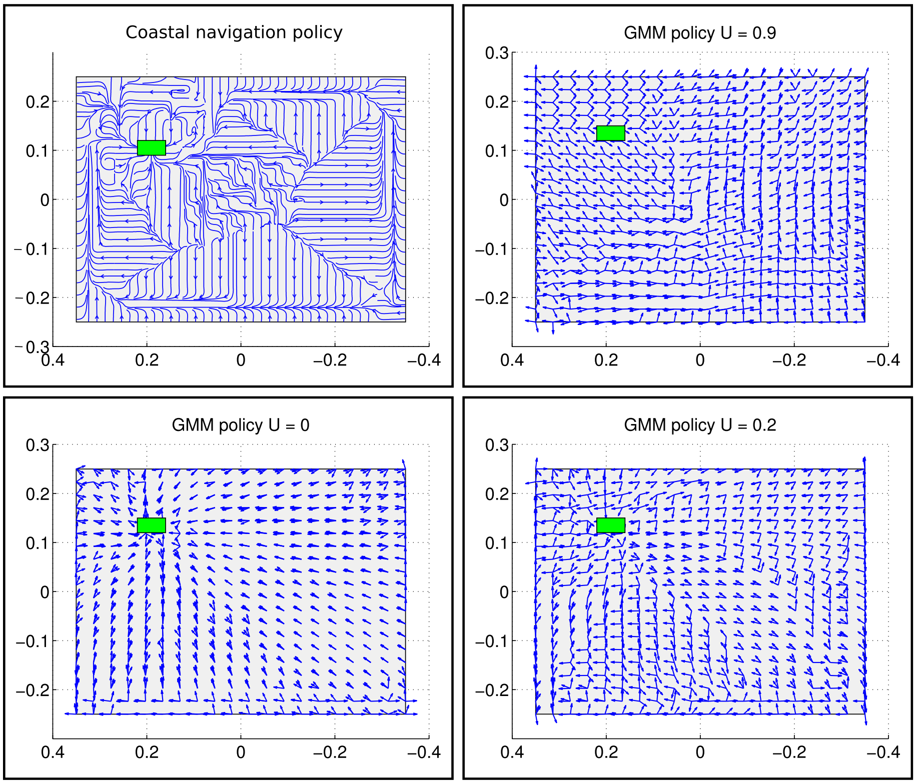

Figure 10 Illustration of the vector field for the coastal and GMM policy. *Top left*: Coastal policy. There is only one possible direction for every state at any time. The values of λ_2 in the cost function were set experimentally. *Others*: The GMM policy for three different levels of uncertainty. For each point, multiple possible actions are possible which are reflected by the number of arrows (only the first three most likely actions). As the uncertainty decreases, the policy becomes less multi-model, but still is around the edges and corners. Note that once being certain if one is close to the edge there is a possibility to go either straight to the goal or stay close to the edge and corners.

default setting. The initializations (both real and believed end effector locations) for the remaining four experiments were chosen in order to reflect particular situations which highlight the differences and drawbacks between each respective search method. Figure 11 depicts the starting points for the four searches. One hundred trials were carried out in the search experiment for which the end effector position and belief were initialized uniformly (uniform search experiment). As for the other 4 search experiments, 40 separate runs were carried for each of the 3 algorithms.

Table 2 reports the mean and variance of the distance taken to reach the goal for each search method for all five experiments. We report on ANOVA to test that all experiments were significantly different from one another as were the searches. We test the null hypothesis, H_o, that there is no statistical difference between the five search experiments. Before performing the ANOVA, we verified that our dependent variable, distance taken to reach the goal, follows a normal distribution for all methods and all experiments (a total of $5 \times 3 = 15$ tests), an assumption which is required by an ANOVA analysis. A Kolmogorov-Smirnov test was performed on each experiment and associated search method. A total of 11/15 searches rejected the null hypothesis with a significance level of less than 5% (p value <0.05).

In Table 3, we report the p values and F statistics for an ANOVA on the five different experiments, where our null hypothesis is that all experiments produce statistically the same type of search. For all experiment types, the p value is extremely small, below a significance value of 1% (p value <0.01) which indicates that we can safely reject the null hypothesis and accept that all experiments produced very different searches, which is important for a comparative study.

As the first ANOVA only indicated that the experiments produced different searches, we also performed a second ANOVA test between the paired search methods, to confirm that the methods themselves are statistically different. Table 4 illustrates the difference between the individual search methods for each experiment. It was found that most search algorithms produced significantly different searches (p value <0.01) with the exception of the GMM and coastal algorithm for the uniform and

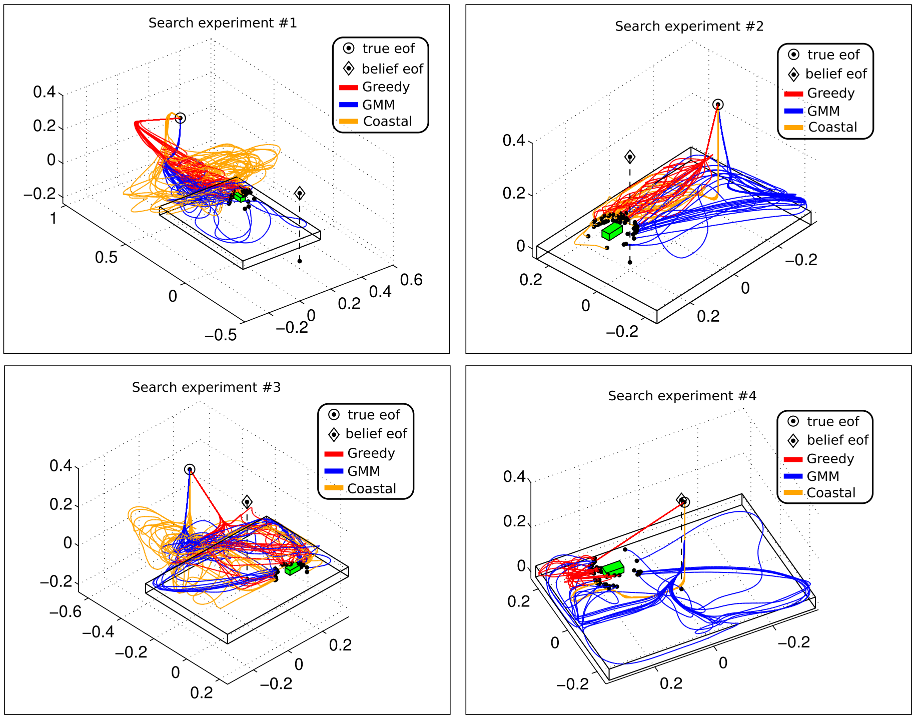

Figure 11 Four search initializations. From *top left* to *bottom right*, we refer to them as #1 to #4, respectively. The circle with a black dot at its centre indicates the true starting point of the end effector (eof), whilst the triangle with the black dot is the initial believed location of the eof. The initialisation in #1 was chosen such that the true and believed eof location were at opposite sides of the table. This setting was selected to highlight the drawback in methods which do not take into account uncertainty. The second initialisation, #2, reflects the situation where once again their is a large distance between true and believed location of the eof. However, this time, both are right on top of the table. The starting points in #3 are a variant on #1 but with the difference that the believed eof position is above the table whilst the true eof location is not. The last experiment, #4, was a setup which would be favourable to algorithms which are inclined to be greedy; both true and believed eof locations are close to one another.

#3 experiment (p value <0.1). However, the GMM and coastal trajectories for the #3 experiment appear to be quite different when the trajectories are off the table's surface (see Figure 11, *bottom left*) but share similar characteristics such as edge following behaviour.

From our ANOVA analysis, we conclude that the behaviour exhibited by the three search strategies are significantly different. This is certainly the case for the greedy and GMM methods, even though in certain situations the greedy and coastal policies display similar behaviour such as in experiment #1. The reason for this is that both the greedy and coastal policies start in a situation where there are no salient features available, and their polices take the true end effector location to an even more feature deprived region. In this situation, the GMM policy is the clear winner with respect to the distance taken to reach the goal.

In experiment #2, both greedy and coastal policies perform equally well and will usually perform faster than the GMM model if the true and believed locations of the end effector do not leave the surface of the table. If this is not the case, they will both reduce the uncertainty in a very inefficient way as the modes will often change during the period of the search, where they are in contact with

Table 2 Mean distance and variance taken to reach the goal for three methods in five experiments

	Greedy	GMM	Coastal
Uniform	1.5396 (0.4580)	**0.9981 (0.1440)**	1.1267 (0.5678)
#1	3.0205 (0.3567)	**1.8220 (0.2314)**	3.4383 (1.5044)
#2	**0.8025 (0.0129)**	1.4129 (0.1446)	0.9392 (0.0126)
#3	**1.1429 (0.0804)**	1.8036 (0.1670)	2.1432 (0.8136)
#4	0.7505 (0.0383)	1.3451 (0.0762)	**0.6820 (0.0094)**

Values are expressed as mean (variance). The entries in bold correspond to the results of the search algorithm which obtained the fastest time to reach the goal in each type of experiment/search.

Table 3 ANOVA test of the null hypothesis (rejected): all searches are the same

Search method				
Uniform	**#1**	**#2**	**#3**	**#4**
2.01e−06 (14)	5.03e−07 (19)	7.17e−11 (36)	4.1e−06 (15)	4.21e−16 (67)

Values are expressed as p value (F). All the p values are extremely small which indicate that the null hypothesis can safely be rejected.

Table 4 ANOVA between paired search methods

	Greedy vs GMM	Greedy vs coastal	GMM vs coastal
Uniform	3.59e−08 (30)	3.32e−04 (13)	**1.90e−01 (2)**
#1	5.80e−08 (46)	**1.88e−01 (2)**	4.58e−06 (28)
#2	3.60e−08 (47)	4.68e−04 (14)	4.54e−06 (28)
#3	3.57e−07 (37)	2.07e−05 (23)	**1.25e−01 (2)**
#4	6.70e−10 (64)	**1.58e−01 (2)**	6.34e−13 (107)

Values are expressed as *p* value (*F*). The entries in bold are those in which the null hypothesis could not be rejected. The first column gives an indication of the probability that both the greedy and GMM searches are statistically the same (the null hypothesis). This was rejected with a tolerance of below 1%. In the second column, greedy vs coastal searches #1 and #4 are statistically closer than the rest with a *p* value threshold of 10% required to be able to reject the null hypothesis. In the third column, the uniform and #3 are not statistically different and would require a higher threshold on the *p* value to be so.

the table. This leads to the believed position (most likely state, $\hat{x}_t$) varying greatly, resulting in an increased time period before the uncertainty has been narrowed down sufficiently for a contact to occur with the table (or simply by chance).

Figure 12 shows the normalised uncertainty with respect to the distance remaining to the goal for all experiments, (#3 is excluded being similar to the #2).

The results show which methods actively minimise the uncertainty and which methods found the goal whilst being more dependent on chance. For all the reported experiments, the GMM (learned from human searches) reaches a lower expected uncertainty than all other search algorithms. For the Uniform and #1 search experiment, all methods reach the same final uncertainty level. However, for the #2 and #4 experiments, the GMM reaches the goal with significantly lower uncertainty. It is inferred that the GMM model actively minimises the uncertainty which is also reflected in the distance taken for this method to reach the goal in comparison with the other methods.

The rows in Table 2 for the greedy (#2) and coastal navigate (#4) are an order of magnitude faster than the GMM method. However, both have a far higher level of uncertainty at the arrival which leads to the assumption that chance has a non-negligible effect on their success.

Conclusions

In this work, we have shown a novel approach in teaching a robot to act in a partially observable environment. Through having human volunteers demonstrate the task of finding an object on a table, we recorded both the inferred believed position of their hand and associated action (normalised velocity). A generative model mapping the believed end effector position to actions

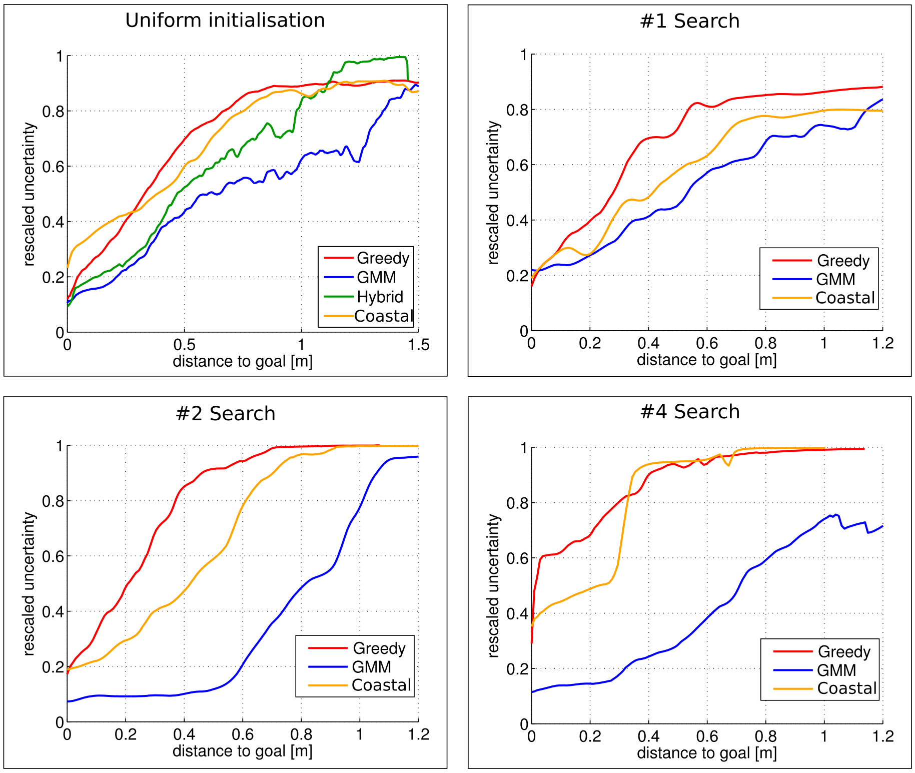

Figure 12 Reduction of the uncertainty for the uniform, #1, #2 and #4 experiments. The expected value is reported. *Top left*: uniform initialisation, expected uncertainty for the greedy (red), GMM (blue), hybrid (green) and coastal (orange) search strategies. *Top right*: experiment #1. *Bottom left*: experiment #2. *Bottom right*: experiment #4.

was learned, encapsulating this relationship. As speculated and observed, multiple strategies are present given a specific belief. This can be interpreted as the fact that humans act differently given the same situation.

The behaviour recorded from the human demonstrations, encoded as set of expected sensations, showed the presence of not only trajectories which both remained near the edge and corner features but also trajectories which remained far away. The fact of risk-prone and risk-averse behaviour was further confirmed by the overlap of the risk factor of human and GMM-generated trajectories with that of the greedy risk factor. According to the feature-based factor, more than 70% of the human search trajectories were considered to be risk-averse, whilst 93% according to the risk factor. Similarly, the GMM search trajectories showed to be 89% to 88% risk-averse.

In terms of the comparative study, the GMM controller is more adapted to dealing with situations of high uncertainty and better takes it into account than greedy or coastal planning approach. This is evident in the experiment where the believed position and true position of the end effector were significantly far apart and distant from salient areas. Future questions of scientific value to be addressed are to which extent do humans follow the reasoning of a Markov decision process in a partially observable situation where the state space is continuous (the problem has been partially addressed in [19] for discrete states and actions). A further aspect of interest is to study the situation where multiple beliefs are present and investigate how humans perform simultaneous localization and mapping as opposed to active localization which was the area of interest of this research.

Competing interests
The authors declare that they have no competing interests.

Competing interests
GDC carried out the experimental design and data analysis. GDC and AB equally contributed to the methodology design and manuscript writing and editing. Both authors read and approved the final manuscript.

Acknowledgements
This research was supported by the European project, Flexible Skill Acquisition and Intuitive Robot Tasking for Mobile Manipulation in the Real World (First-MM), in Robotic Research.

Received: 23 April 2014 Accepted: 30 August 2014
Published online: 07 November 2014

References

1. Kaelbling LP, Littman ML, Cassandra AR (1998) Planning and acting in partially observable stochastic domains. Artif Intell 101(1):99–134
2. Smith T (2007) Probabilistic planning for robotic exploration. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA
3. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
4. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics (intelligent robotics and autonomous agents). The MIT Press, Cambridge
5. Pineau J, Gordon G, Thrun S (2003) Point-based value iteration: an anytime algorithm for POMDPS. In: IJCAI, Mexico, 9–15 August 2003. pp 1025–1030
6. Kurniawati H, Hsu D, Lee WS (2008) SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Oliver Brock JT, Ramos F (eds). Proceedings of robotics: science and systems (RSS), Zurich, 25–28 June 2008
7. Smith T, Simmons R (2004) Heuristic search value iteration for POMDPS. In: Proceedings of the 20th conference on uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington. pp 520–527
8. Shani G, Brafman RI, Shimony SE (2007) Forward search value iteration for POMDPS. In: Proceedings of the 20th international joint conference on artifical intelligence
9. Shani G, Pineau J, Kaplow R (2013) A survey of point-based POMDP solvers. Autonomous Agents Multi-Agent Syst 27(1):1–51
10. Roy N, Pineau J, Thrun S (2000) Spoken dialogue management using probabilistic reasoning. In: Iida H (ed). Proceedings of the 38th annual meeting of the association for computational linguistics, Hong Kong, 2000. pp 93–100
11. Thrun S (2000) Monte carlo POMDPs. In: Solla SA, Leen TK, Müller K-R (eds). Advances in neural information processing systems 12. MIT Press, Cambridge. pp 1064–1070
12. Hsiao K, Kaelbling L, Lozano-Perez T (2010) Task-driven tactile exploration. In: Yoky Matsuoka HD-W, Neira J (eds). Proceedings of robotics: science and systems (RSS)
13. Hebert P, Howard T, Hudson N, Ma J, Burdick JW (2013) The next best touch for model-based localization. In: International conference on robotics and automation (ICRA), Karlsruhe, 6–10 May 2013. pp 99–106
14. Kasper M, Fricke G, Steuernagel K, von Puttkamer E (2001) A behavior-based mobile robot architecture for learning from demonstration. Robot Autonom Syst 34(2):153–164
15. Hamner B, Singh S, Scherer S (2006) Learning obstacle avoidance parameters from operator behavior. Field Robot 23(11/12):1037–1058
16. Silver D, Bagnell JA, Stentz A (2010) Learning from demonstration for autonomous navigation in complex unstructured terrain. IJRR 29(12):1565–1592
17. Nicolescu MN, Mataric MJ (2001) Learning and interacting in human-robot domains. IEEE Trans Syst Man Cybern Syst Hum 31(5):419–430
18. Lidoris G (2011) State estimation, planning, and behavior selection under uncertainty for autonomous robotic exploration in dynamic environments. Kassel University Press GmbH, Kassel
19. Bake C, Tenenbaum J, Saxe R (2011) Bayesian theory of mind: modeling joint belief-desire attribution. In: Thirty-third annual conference of the Cognitive Science Society, Boston, 20 July 2011. pp 2469–2474
20. Richardson H, Bake C, Tenenbaum J, Saxe R (2012) The development of joint belief-desire inferences. In: Proceedings of the 34th annual meeting of the Cognitive Science Society (COGSCI), Sapporo, 1 Aug 2012
21. Baker CL, Tenenbaum JB, Saxe RR (2006) Bayesian models of human action understanding. In: Advances in neural information processing systems 18, Nevada, 4 December 2006. pp 99–106
22. de Chambrier G, Billard A (2013) Learning search behaviour from humans. In: IEEE international conference on robotics and biomimetics (ROBIO), Shenzhen, 12 December 2013. pp 573–580
23. Arulampalam MS, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188
24. Huber MF, Bailey T, Durrant-Whyte H, Hanebeck UD (2008) On entropy approximation for Gaussian mixture random vectors. In: Multisensor fusion and integration. pp 181–188
25. Billard A, Calinon S, Dillmann R, Schaal S (2008) Robot programming by demonstration. In: Bruno Siciliano Prof. OKP (ed). Springer handbook of robotics. Springer, Berlin. pp 1371–1394
26. Roy N, Burgard W, Fox D, Thrun S (1999) Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In: IEEE international conference on robotics and automation. pp 35–40

doi:10.1186/s40638-014-0008-1
Cite this article as: de Chambrier and Billard: **Learning search polices from humans in a partially observable context.** *Robotics and Biomimetics* 2014 **1**:8.