Chapter 3

Learning to reason with uncertainty as humans

The conclusions drawn from the literature survey in Chapter 2 are that non-heuristic methods for planning and control rely heavily on the initial data provided to their respective optimisers. An ideal initial set of behaviour should be comprised of explorative and exploitative actions so that a final optimal policy can quickly achieve the balance between minimising uncertainty and solving the task at hand. This is especially true for Reinforcement Learning (RL) methods which make use of explorative actions to be able to find an optimal policy. In many RL applications random exploration or Gaussian noise perturbation is sufficient to find an optimal policy. This is the case when either an exhaustive search of the action space is possible (mountain cart, inverted pendulum, etc...) or in policy search methods where the policy is parametrised by a few parameters. In continuous action-state space POMDPs, when a generic non-parametric policy is desired this is not feasible, especially when the decision horizon is long. Continuous action-state space POMDPs applications have predominantly focused on cases in which the uncertainty can be quantized by a single Gaussian parametrisation. This representation can be constraining since it requires the observation likelihood to be Gaussian as well. This assumption is restrictive and ill-suited for haptic search tasks in which observations are discontinuous and occur as impulses.

In this chapter, we demonstrate that human foresight and intuition can be leveraged as a means of solving the exploration/exploitation dilemma under partial observable conditions. Human beings are versatile in their ability to accomplish tasks which are considered to be complex by current robotic standards. This perceived ability which we have over current robotic systems, due to our prior domain knoweldge and experience, can be extracted, encapsulated and transferred to a robot apprentice.

To demonstrate the application of the transfer of behaviour from a human teacher to a robotic apprentice we apply the framework outlined in Chapter 2, Section 2.4 (PbD-POMDP) to a blindfolded haptic search task. In our blindfolded search task, both a robot and a human must search for an object on a table whilst deprived of vision and hearing, illustrated in Figure 3.1. The robot and human both 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 3.1, a human has his sense of vision and hearing impeded, making the perception of the environment partially observable and leaving only the sense of touch available for solving the task. 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.

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 hand actions (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 uncertainty into account is more efficient with respect to distance travelled to reach the goal.

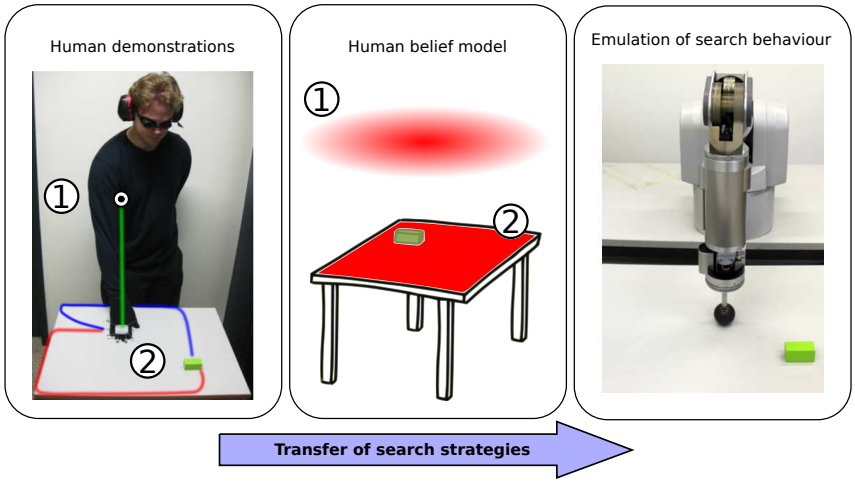

Figure 3.1: Blindfolded search task *Left:* Search task, a human demonstrator searching for the green wooden block on the table given that both his hearing and vision senses have been impeded. He starts (hand) at the white spot near position (1). The the red and blue trajectories are examples of possible 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.

There are **two assumptions** we make when applying Programming by Demonstration, PbD (also known as Imitation Learning), to the POMDP task described above. The first assumption is that the human teacher's *spatial cognitive* abilities are good enough to accomplish the task in a consistent fashion.

In other words demonstrations should not be random and a pattern exists. The second assumption is that human's beliefs inferred by the apprentice are close to the actual belief of the human.

3.1 Outline

- 3.2 Background
  We review aspects of the literature in robotics and cognitive science which are related to spatial navigation which consider scenarios with limited perceptual information. We review related literature from *Spatial Navigation*, *Theory of Mind* and *Programming by Demonstration*.
- 3.3 Experiment
  The table search experiment protocols are described and we detail how to learn and transfer search strategies from human teachers to a robot apprentice. A total of 15 human teachers participated and each gave 10 demonstrations, giving a total of 150 searches.
- 3.4 Formulation
  We detail the implementation of the human belief in terms of a Particle Filter (PF). This includes the measurement and motion models. We describe how we compress the belief particle filter in terms of the most likely state and differential entropy.
- 3.5 Policies
  - 3.5.1 Modelling human search strategies
    We detail the implementation and parametrisation of a Gaussian Mixture Model (GMM) policy encapsulating the human search strategies and how it synthesises new searches.
  - 3.5.2 Coastal Navigation
    We detail the implementation of a Coastal Navigation policy, used as a comparison with the GMM policy.
- 3.6 Results
  We conduct three types of analysis: we quantify the behaviour present in humans and policies in terms of riskiness; we qualitatively evaluate the differences between the GMM policy learned from human demonstrations and the Coastal Navigation policy; we evaluate the distance taken to find the goal for a set of four search policies, including the GMM.

3.2 Background

3.2.1 Spatial navigation

Spatial navigation, Wang (2007), Wolbers and Hegarty (2010), focuses on the role that sensory perception (vision, vestibular, proprioception ...), motor control and mental cognition have on the navigational ability of humans, animals and insects. A central aspect of spatial navigation is the way in which we mentally represent the geographical world, known as a *cognitive map* (mental representation of environment first proposed by Tolman, 1948) and how we update our pose estimation in this map. The aspects of both construction and correction of a cognitive map have been studied in great depth, Wolbers et al. (2008). There is reported evidence that we use both vestibular and proprioception in inferring self-motion in order to update our position through dead reckoning (also known as path integration). Given the estimated position we then use external cues such as geometric (the shape of a room) and features (the colour of the walls), to correct our position. The actual representation of our position and environment in our cognitive map has been proposed, Burgess (2006), to be either encoded in our own frame of reference (egocentric) or in a frame of reference which is independent to us (allocentric) and acts like a standard paper map or both. This cognitive map enables us to reason about the relations between our own position and that of other items and landmarks present. This representation also facilitates our ability to localise ourselves and plan novel routes when needed.

In Wang and Spelke (2000), the authors studied the effect that disorientation has on blindfolded subjects' ability to recover their heading, which is necessary for re-localisation. Through eight different experiments they concluded that humans have an egocentric cognitive map.

Studies have also looked at the difference between congenitally blind, late blind and sighted people in their ability to encode ego-allocentric cognitive maps. In Pasqualotto et al. (2013), the authors dispose a set of seven objects (brush, slipper, pan, dish, book, spoon, bottle) in the form of an array in a 12.5m $\times$ 9m room. The objects are positioned on top of stools. During a training phase, ten congenitally blind, ten late blind and ten blindfolded sighted people were taken through the setup and touched all objects present. This guided exploration (the experimenter leading the subject through the object array) was repeated until the participants could correctly recall all the objects' locations twice consecutively without help. In a testing room (no objects present) the participants were asked "Judgement of Relative Direction" questions and the accuracy and response time were recorded. From the results the authors concluded that blindfolded and late blinded participants used a allocentric representation of the object array, whilst the congenital blind subjects use an egocentric model. The cause of this difference is attributed to the role played by vision in the development of the multisensory brain area, in which vision is necessary for the

development of an allocentric model.

Many similar experiments have been conducted and a summary can be found in the following review Burgess (2006), where the authors explicitly state that a consensus has formed; both egocentric and allocentric representations of the environment are working in parallel. Current questions ponder whether allocentric models are part of the semantic memory as opposed to the procedural memory used by the egocentric model.

Spatial cognition and memory

The quality of the human teacher in search tasks, which are partially observable in the terms of absence of vision, will strongly depend on the teacher's ability to maintain an accurate cognitive map of his environment. This implies that the size of the environment and search task will have an effect on the teacher's ability to provide near optimal demonstrations. Early and influential research into human's short term memory was presented in 1956 by George Miller in a seminal work, Miller (1956) (22'780 citations), in which he described the "so called" magical number of our short term memory as being 7 ± 2 items, known as *Miller's Law*. This research was conducted on a one dimensional task in which no spatial navigation was required. Since then there have been many studies investigating the limits of short term memory.

In Lavenexa et al. a set of subjects had to find either 1, 3, 5 or 7 goal pads, among a grid array of 23 pads in a 4m $\times$ 4m room, within a 1 minute interval. They measured the subjects' error in terms of the number of locations visited before finding the goals. They found that on average the subjects had to visit "$1.6 \times \#num\_goals$" pads before achieving the task. The authors concluded that in this spatial navigation task there was no magical number which represents the limit of short term memory. In another spatial navigation experiment, Iachini et al. (2014), the effect that the scale of the environment has on the ego-allocentric representation is studied in blindfolded, late and early blind subjects. The main findings were that cognitively blind people have more difficulty in developing an allocentric representation of the world.

In Stankiewicz et al. (2006), a search task in a virtual maze is conducted by a set of human subjects. The aim is to investigate the limitations that perception, memory and uncertainty have on human decisions in comparison with an ideal agent (POMDP solution). The authors' main findings were that as the size of the maze increases the performance of the human subject decreases with respect to the ideal agent, as human subjects are limited by the uncertainty in their location and have difficulty in maintaining multiple hypotheses.

Summary: spatial cognition

The studies detailed above reported that if the environment is not overly large and complex our cognitive model is sufficient to produce policies which are on par with an optimal POMDP agent.

Our study seeks to transfer exploratory behaviour from human teachers to a robot apprentice in a partially observable setting. In our search scenario the environment is less than 3 meters in length and 2 meters in depth with a single goal object to be found. Given this setup and the evidence from previous studies, humans should be able to achieve this task with a high level of proficiency.

This is beneficiary since currently both humans and animals are better at spatial navigation than robots Stankiewicz et al. (2006) especially when uncertainty is present. The quality of the demonstrations will strongly depend on the teacher's short term memory in retaining a sufficiently accurate cognitive map of the environment.

3.2.2 Human beliefs

A crucial aspect for the success of PbD-POMDP learning is that the apprentice be able to infer the human's belief of his location whilst he is searching. In others words the apprentice (human or robotic) has to infer the cognitive map of the teacher.

The study of inference of another's mental state is part of Theory of Mind (ToM) Sodian and Kristen (2010), which is concerned with our ability to infer beliefs, desires, intentions, perception, goals and current knowledge. In this study, the apprentice will have to infer the teacher's beliefs which we assume are **rational**. A rational belief is a belief for which observations bring supportive evidence and gradually increase the certainty of the belief. In a recursive formulation this known as Bayesian Theory of Mind (BToM), where the Bayesian component highlights the hypothesis that humans integrate information and update their beliefs in a similar fashion to Bayes rule.

Due to the complexity in the number of sensory sub-components, such as gaze following, and their interplay, required as a precursor to the development of a ToM, much effort has been focused their development. Early work in implementing a ToM in a humanoid robot was introduced in Scassellati (2002) and is based on ToM models of Leslie (1994) and Baron-Cohen (1995). The author focused on building basic skills such as face finding and distinguishing animate and inanimate stimuli but left open the problem of the final interaction between all the components.

In Butterfield et al. (2009), the authors model ToM as a Markov Random Field which defines a joint probability distribution over a set of hidden actions and observation variables. The functions of these variables are hand-crafted for each experiment. The authors demonstrate that through a suitable parameterisation of the MRF they achieve results comparable to the predictions of

ToM. Recently in Devin and Alami (2016), ToM and planning architecture have been integrated in a joint action collaborative human robot task, in which position, goal and action state of the human partner is maintained by his robotic assistant.

Work on modelling human beliefs and intentions has been undertaken in cognitive science, Bake et al. (2011), Richardson et al. (2012). In Baker et al. (2006), the authors present a Bayesian framework for modelling the way humans reason and predict actions of an intentional agent. The comparison between a generic Bayesian model and the humans' predictions yielded similar inference capabilities. This when asked to guess the intentions of a goal oriented agent in a 2D world, which both the Bayesian model and the humans were observing. This provided evidence supporting the hypothesis that humans integrate information using Bayes rule. Further, in Bake et al. (2011), 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 that of their Bayesian model. Again they found the human's inference was comparable to that of the Bayesian model.

In our PbD-POMDP framework we make a similar hypothesis that humans integrate information in a Bayesian way, however in a continuous domain. We infer the belief that humans have of their location in the world during search tasks.

3.2.3 Programming by demonstration & uncertainty

Programming by demonstration (PbD) is advantageous in the POMDP and MDP contexts 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 discussed in Chapter 2).

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 Kasper et al. (2001), 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 Hamner et al. (2006) the parameters of a controller which performs obstacle avoidance are learned from human demonstrations. The uncertainty is inherently handled by learning the relation between

sensor input and control output. In Silver et al. (2010) 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 Nicolescu and Mataric (200[illegible]e 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 Lidoris (2011), 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.

3.3 Experiment: table search

In our search task setup, Figure 3.2 and Figure 3.3 (*top left*), a group of 15 human volunteers were asked to search for a wooden green block located at a fixed position on a bare table. Each participant repeated the experiment 10 times from each of 4 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 where 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 with certainty. We make the critical assumption that the belief is observable in the first time step of the search and all following beliefs

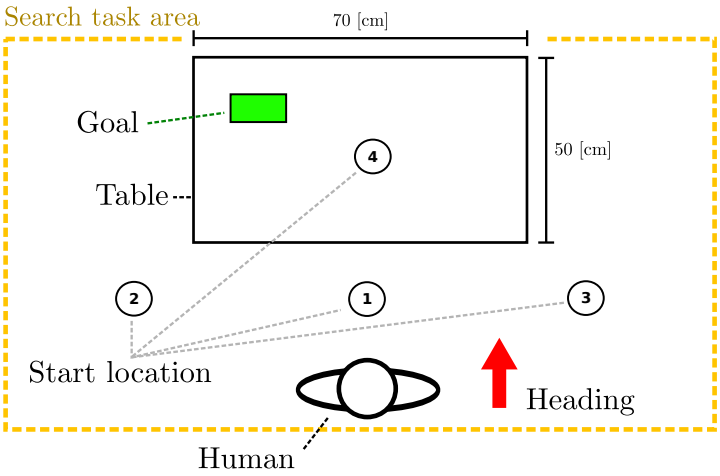

Figure 3.2: Table search task. Blindfolded human subjects after a disorientation step are placed in one of the four starting locations. The heading of the subject is always kept the same. The human's objective is to locate the green block on the table. Throughout all experiments the green wooden block is kept in the same location.

are assumed correct through applying Bayes integration. The belief is always initialized to be uniformly distributed on top of the table, see Figure 3.3 (*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 translational 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 3.3 *bottom left*, we illustrate four representative recorded searches whilst in the *bottom right*, we illustrate a set of 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.

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 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.

3.4 Formulation

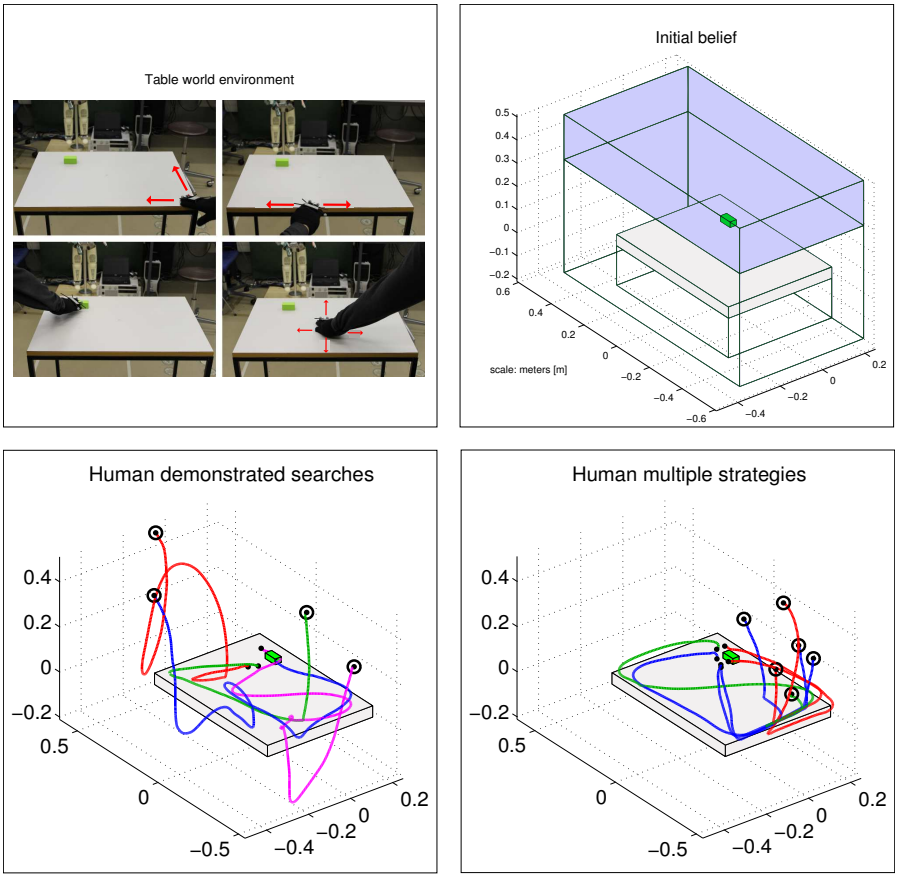

Figure 3.3: *Top left*: A participant is trying to locate the green wooden block on the table given that both vision and hearing senses have been inhibited. The location of his hand is being tracked by the OptiTrack® system. *Top right:* Initial distribution of the uncertainty or belief we assume the human has with respect to his position. *Bottom left:* Set of recorded searches, the trajectories are with respect to the hand. *Bottom right:* Trajectories starting from 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 corner. 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.

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 the hand's displacement $\dot{x}_t$. In our case since the environment is partially observable we have a belief or probability density function, $p(x_t|y_{0:t}, \dot{x}_{0:t})$, which is conditioned on all sensing information, $y_{0:t}$, (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 and the history of applied actions, $a_{0:t}$. We seek to learn this mapping, $f : p(x_t|y_{0:t}, \dot{x}_{0:t}) \mapsto \dot{x}_{t+1}$, from demonstrations. During each demonstration we record a set of variables consisting of the following:

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

A statistical controller was learned from the tuple dataset: $\{(\dot{x}, \hat{x}, U)\}$ recorded during the search trials of the human subjects. 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* $p(y_t|x_t)$, 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.

$$p(x_t|y_{0:t-1}, \dot{x}_{0:t}) = \int p(x_t|x_{t-1}, \dot{x}_t)\, p(x_{t-1}|y_{0:t-1}, \dot{x}_{0:t-1})\, dx_{t-1} \quad (3.1)$$

$$p(x_t|y_{0:t}, \dot{x}_{0:t}) = \frac{p(y_t|x_t)p(x_t|y_{0:t-1}, \dot{x}_{0:t})}{p(y_t|y_{0:t-1})} \quad (3.2)$$

The probability distribution over the state $p(x_t|y_{0:t}, \dot{x}_{0:t})$ 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* (Arulampalam et al., 2002, p.182). From previous literature Bake et al. (2011) 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 model

The sensing model tells us the likelihood, $p(y_t|x_t)$, of a particular sensation y_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 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 $y_t = [p_c, p_e, p_s, p_a]$, where p refers to probability and the subscript corresponds to the first letter of the feature it is associated with. In Equation 3.3, the sensing function, $h(x_t, x_c)$, 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.

$$p_c = h(x_t, x_c; \beta) = \exp\left(-\left(\beta \cdot \|x_t - x_c\|\right)^2\right) \tag{3.3}$$

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 1cm way 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 3.3, and gives the likelihood, $p(y_t|x_t)$ 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), $p(y_t|x_t) = 1 - JSD(y_t||\hat{y}_t)$, between true sensing vector, z_t, obtained by the agent and that of the hypothetical sensation

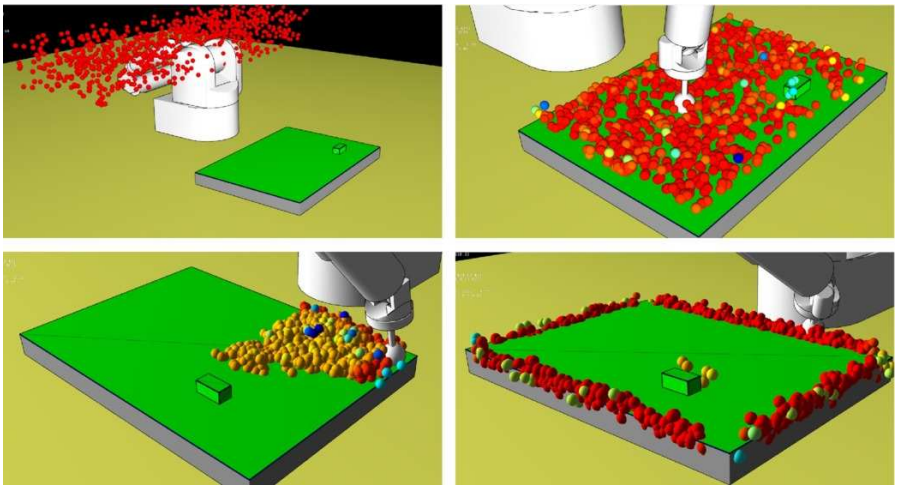

Figure 3.4: Four different time frames of the evolution of the belief particle filter. *Top left*: Initial belief distribution; a lot of uncertainty. *Top right:* First contact is made with the table, the measurement likelihood restrains the samples to be on the table's surface. *Bottom right:* First contact is an edge. *Bottom left:* Gradual localisation.

$\hat{y}_t$ generated at the location of a particle. In Figure 3.4, four different beliefs are shown.

Motion model

The motion model is straight forward 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 $p(x_t|x_{t-1}, \dot{x}_t) = \mathrm{J}(q)\dot{q} + \epsilon$ 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. In our experiment we consider the noise from motion to be negligible. An increase in uncertainty already results from the re-sampling stage of Sampling Importance Resampling (SIR) particle filter and we found no need to add additional motion noise. The particles' positions were updated by applying the measured velocity obtained from either the visual tracking system (when recording the human demonstrations) or the robot's forward kinematics.

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 less the uncertainty. In our context, a set of weighted

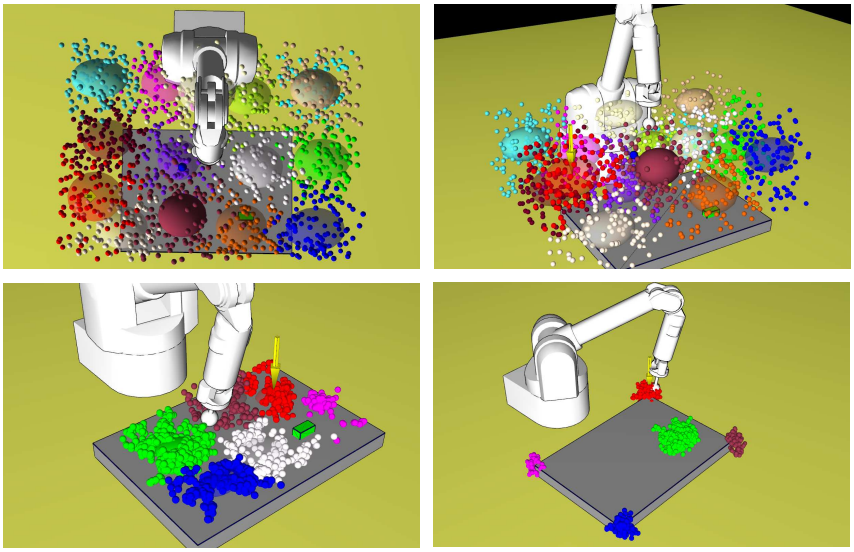

Figure 3.5: Representation of the estimated density function. *Top Left and Right:* Initial starting point, all Gaussian functions are uniformly distributed with uniform priors. The red cluster always has the highest likelihood which is taken to be the believed location of the robot's/human's end-effector. *Bottom Left:* Contact with the table has been established, the robot location differers 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).

samples $\{w_i, x_i\}^{i=1...N}$ replaces the true probability density function of the belief, $p_u(x_t|y_{0:t}, \dot{x}_{0:t})$. A reconstruction of the underlying probability density is achieved by fitting a Gaussian Mixture Model (GMM), Equation 3.4, to the particles,

$$p_u(x_t|y_{0:t}, \dot{x}_{0:t}\,; \{\pi, \mu, \Sigma\}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_t\,; \mu_k, \Sigma_k) \tag{3.4}$$

where K is the number of Gaussian components, the scalar π_k represents the weight associated to 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 maximizing the likelihood of the density function via Expectation-Maximization (EM).

Given the estimated density we can compute the upper bound of the differ-

ential entropy Huber et al. (2008), H,

$$H\left(p_u(x_t|y_{0:t}, \dot{x}_{0:t}\,; \{\pi, \mu, \Sigma\})\right) = \sum_{k=1}^{K} \pi_k \left(-\log(\pi_k) + \frac{1}{2}\log((2\pi e)^D |\Sigma_k|) \right) \tag{3.5}$$

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.

$$\hat{x}_t = \arg\max_{x_t} p_u(x_t|z_{0:t}\,; \{\pi, \mu, \Sigma\}) = \mu_{(k=\max(\pi))} \tag{3.6}$$

Figure 3.5 depicts different configurations of the modes (clusters) and believed position of the end-effector (indicated by a yellow arrow).

3.5 Policies

3.5.1 Modelling human search strategies

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 use 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 Billard et al. (2008b).

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.

The training data set comprised a total of 20'000 tuples $(\dot{x}, \hat{x}, U)$, from the 150 trajectories gathered from the demonstrators. The fitted GMM $p_s(\dot{x}, \hat{x}, U)$ had a total of 7 dimensions, 3 for direction, 3 for position and 1 scalar for uncertainty. The definition of the GMM is presented below in equation 3.7.

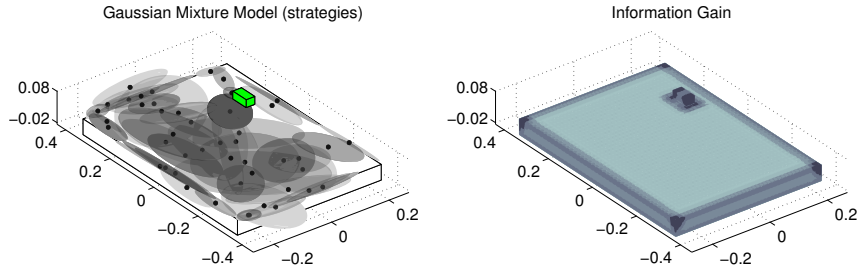

Figure 3.6: *Left:* Resulting search GMM, a total of 67 Gaussian mixture components are present. We note the many overlapping Gaussians: this results from the level of uncertainty over the different choices taken. For example humans follow along the edge of the table in different directions and might 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.

$$p_s(\dot{x}, \hat{x}, U\,; \{\pi, \mu, \Sigma\}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\dot{x}, \hat{x}, U\,; \mu_k, \Sigma_k) \tag{3.7}$$

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

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 Expectation-Maximisation (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 3.6 *(left)* we illustrate the model learned from human demonstrations where we plot the 3 dimensional slice (the position) of the 7 dimensional GMM to give a sense of the size of the model.

3.5.2 Coastal Navigation

Coastal navigation Roy et al. (1999) is a path planning method in which the objective function, Equation 3.8, is composed of two terms.

$$f(x_{0:T}) = \sum_{t=0}^{T} \lambda_1 \cdot c(x_t) + \lambda_2 \cdot I(x_t) \tag{3.8}$$

The first term, $c(x_t)$, 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, $I(x_t)$, 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, $1cm^3$, giving us a total of a 125'000 states. The action space was discretised to 6 actions, two for each dimension meaning that all motion is parallel to the axis. For each state, x_t, an $I(x_t)$ value is computed by evaluating Equation 3.9,

$$I(x_t) = \mathbb{E}_{p(y_t|x_t)}\{H(p_u(x_t|y_{0:t}, \dot{x}_{0:t})\} - H(p_u(x_t|y_{0:t-1}, \dot{x}_{0:t})) \tag{3.9}$$

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 3.9). 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 3.6. 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 the 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.

3.5.3 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 the resulting conditional GMM, Equation 3.10, which is a distribution over velocities or directions.

$$p_s(\dot{x}|\hat{x}, U) = \sum_{k=1}^{K} \pi^k_{\dot{x}|\hat{x},U} \cdot \mathcal{N}\left(\dot{x}\,; \mu^k_{\dot{x}|\hat{x},U}, \Sigma^k_{\dot{x}|\hat{x},U}\right) \tag{3.10}$$

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 3.11

$$\dot{x} = \mathbb{E}\{p_s(\dot{x}|\hat{x}, U)\} = \sum_{k=1}^{K} \pi^k_{\dot{x}|\hat{x},U} \cdot \mu^k_{\dot{x}|\hat{x},U} \tag{3.11}$$

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.

$$\alpha_k(\dot{x}) = \pi^k_{\dot{x}|\hat{x},U} \cdot \exp(-\cos^{-1}(< \dot{x}, \mu^k_{\dot{x}|\hat{x},U} >)) \tag{3.12}$$

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

$$\dot{x} = \mathbb{E}_\alpha\{p_s(\dot{x}|\hat{x}, U)\} = \sum_{k=1}^{K} \alpha_k(\dot{x})\, \mu^k_{\dot{x}|\hat{x},u} \tag{3.13}$$

The final output of equation 3.13 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,

$$\nu = \max(\min(\beta_1, K_p(x_\mathrm{g} - \hat{x}), \beta_2) \tag{3.14}$$

where the β's are lower and upper amplitude limits, x_g is the position of the goal, and K_p 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 3.7.

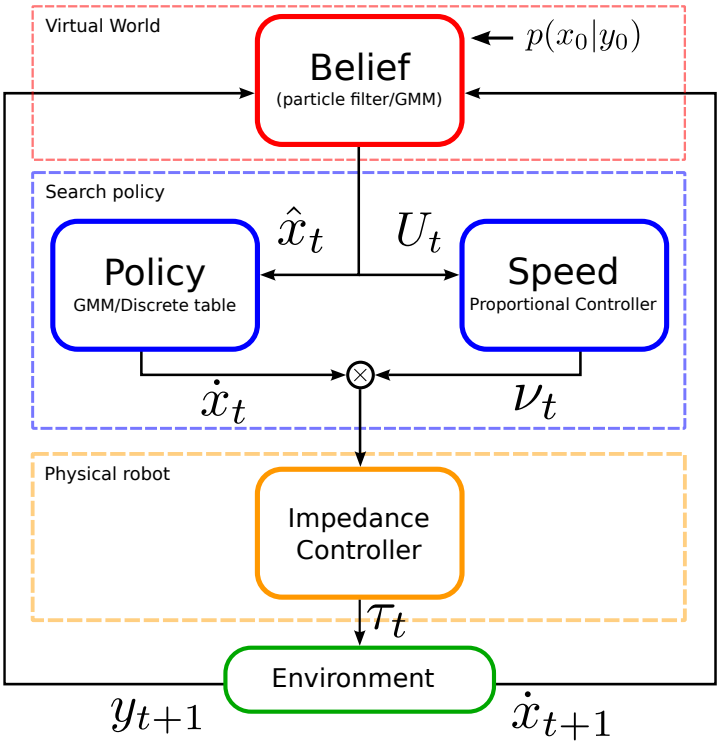

Figure 3.7: Overview of the decision loop. At the top a strategy is chosen given an initial belief $p(x_0|y_0)$ of the location of the end-effector (initially through sampling the conditional). A speed is applied to the given direction based on the believed distance to the goal. This velocity is passed 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.

3.6 Results and discussion

Throughout our evaluation of our GMM PbD-POMDP control policy we will be considering four search policies: Greedy, GMM, Hybrid and Coastal. We evaluate behaviour present in the human demonstrations, and the four above mentioned policies in terms of their riskiness. We qualitatively compare the policies of the GMM model and the Coastal Navigation algorithm and highlight the effect of uncertainty. We finish with a quantitative evaluation of search efficiency in terms of distance travelled until the goal is found. The layout of this section follows as:

- Section 3.6.1, we analyse the types of behaviour present in the human demonstration as well as in four different search algorithms: Greedy, GMM, Hybrid and Coastal.
- Section 3.6.2, we qualitatively analyse the GMM search policy (namely the different modes/decisions present) with respect to the Coastal navigation policy.
- Section 3.6.3, we evaluate the search performance, with respect to the distance taken to reach the goal and the uncertainty profiles towards the end of the searches in 5 different experiments (different types of initializations).

3.6.1 Search & behaviour analysis

For each method (Greedy, GMM, Hybrid, Coastal) 70 searches were performed with all starting positions drawn from the uniform distribution used during the teaching stage (depicted in Figure 3.3 *top right*, page 52). In Figure 3.8 we illustrate the expected sensation $\mathbb{E}\{y\}$ and variance $\text{Var}\{y\}$ for each trajectory with respect to the edge and corner of the table.

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 3.6 *right*, page 58). 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, the policy 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

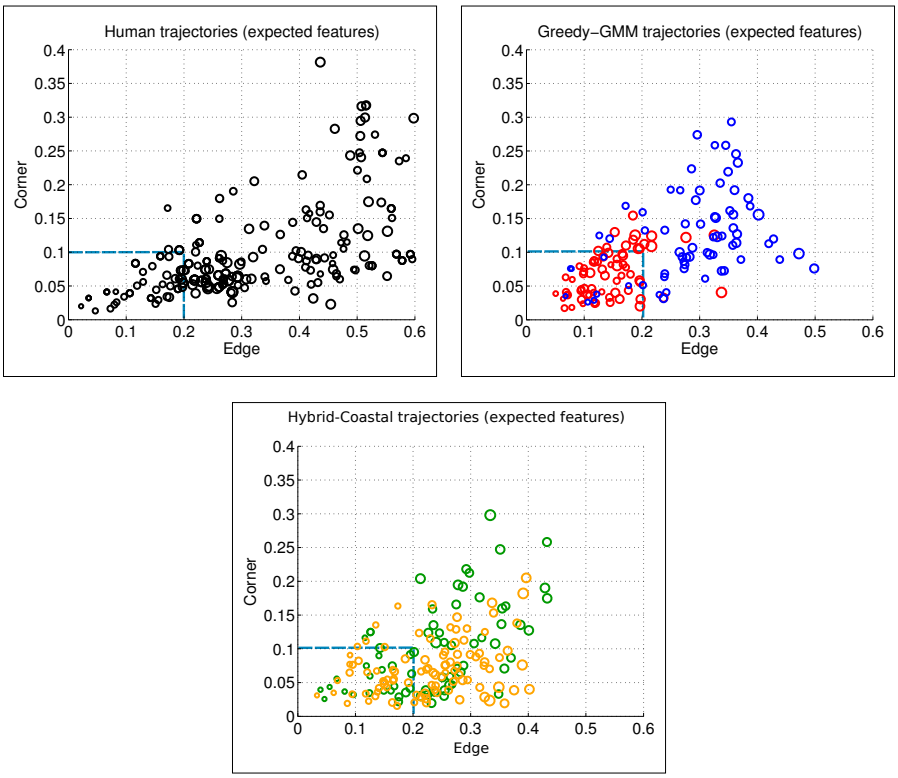

Figure 3.8: Expected sensation. Plots of the expected sensation of the edge and corner feature for all trajectories. The axes are associated with the sensor measurements, 0 means that the corresponding feature is not sensed and 1 the feature is fully sensed. A point in the plots summarises a whole trajectory 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 doted 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.

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

Table 3.1: Percentage of risk-prone trajectories based on two decision criteria, the feature (f) and the risk (r) (information gain) metrics discussed above.

which consists of a trade-off between 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-adverse for the GMM.

We make **the assumption** that Greedy trajectories are risk-prone by nature. We performed a SVM classification on the Greedy-GMM expected features (Figure 3.8 *right*) 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 3.1 *first row* shows that the GMM and Human search trajectories are mostly risk-averse. Surprisingly the Coastal policy seems to be very risk-prone given that it seeks paths close to highly informative areas. 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 3.9 (*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 3.9 (*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 while those with a high expectation of sensing features have a high information gain. Since the metric lies exclusively in the range [0,1] we define that every trajectory which has a Risk factor lower than than 0.5 will be considered risk-averse whilst those above are risk-prone. Table 3.1 *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.

Figure 3.10 (*top left & right*), shows risk-prone (red) and 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

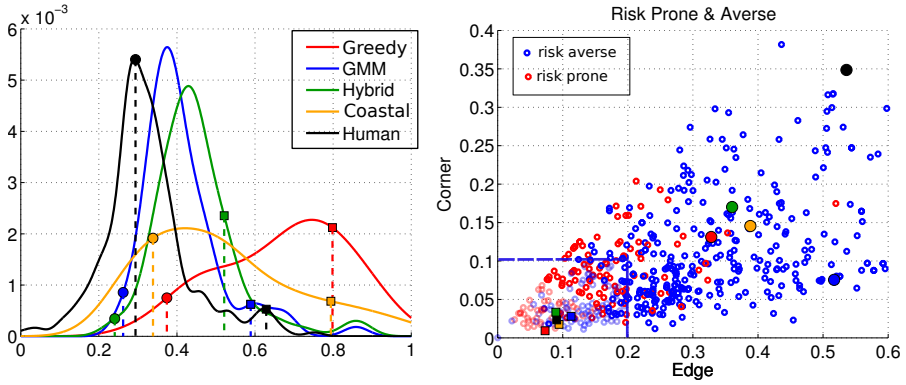

Figure 3.9: Risk of searches. Illustration of risk-prone and risk-averse searches in terms of a Risk factor (*left*) and expected sensation (*right*). *Left:* Each trajectory was reduced to a single scalar, which we call the Risk factor, quantifying the risk of a trajectory. The Risk factor is inversely 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 3.10, to illustrate that these correspond to risk averse and 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.

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 3.10 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 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.

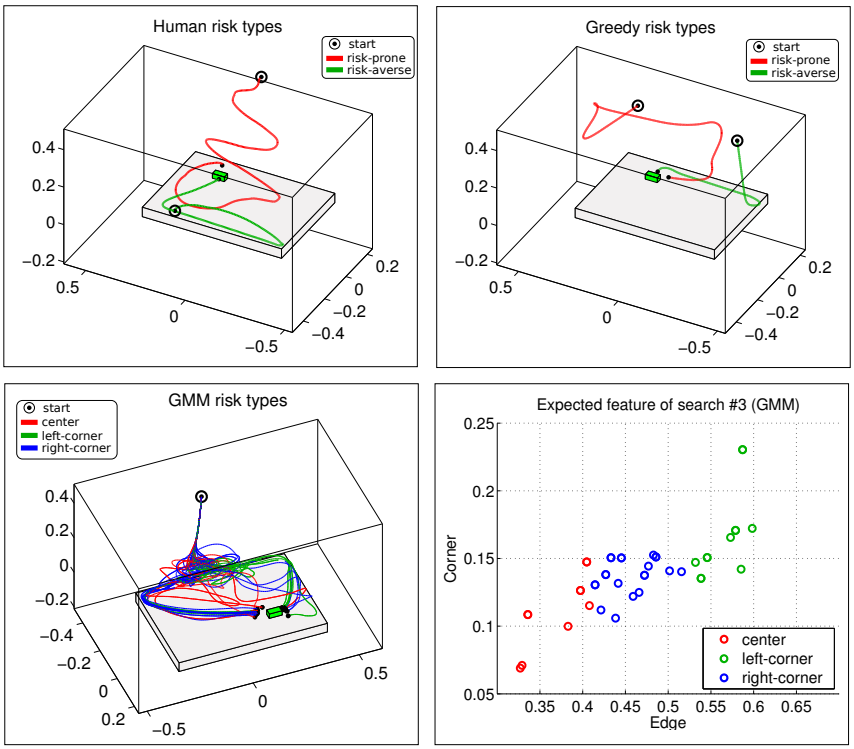

Figure 3.10: Risk prone & averse searches (red & green trajectories). *Top left:* Two human trajectories taken from data shown in Figure 3.9. *Top right:* Two Greedy trajectories. *Bottom left:* GMM trajectories, all starting from the same location, the colour coding is to illustrate the different policies 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 previous).

3.6.2 GMM & Coastal Navigation policy analysis

We next illustrate some of the modes (action choices) present during simulation and evaluate their plausibility. Figure 3.11 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 3.12 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 λ_2=0 the Coastal policy is the same Greedy algorithm.

It can be further seen that when the uncertainty tends towards it's maximum value ($U \to 1$) all behaviour tends to go towards the edges and corners. As the uncertainty reduces ($U \to 0$) 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.

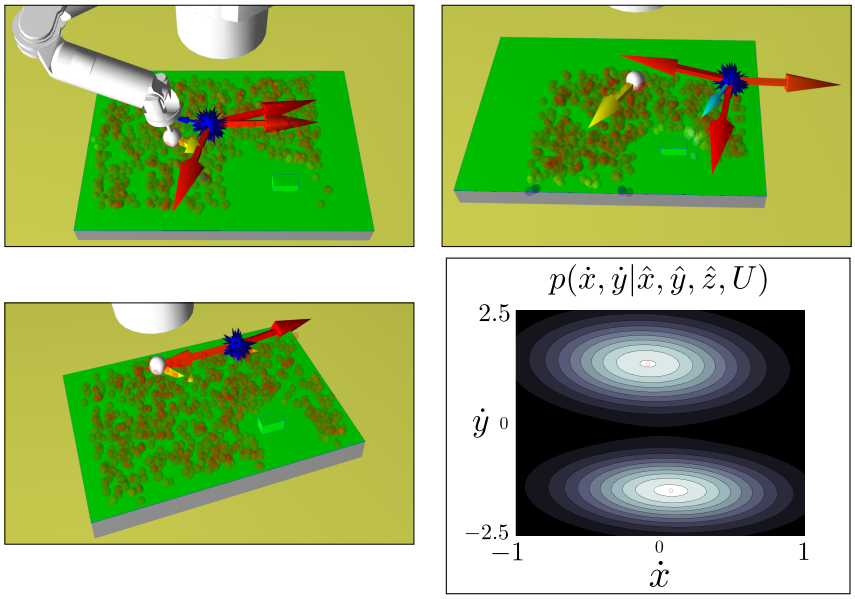

Figure 3.11: Illustration of three different types of modes present during the execution of the task where the robot is being controlled by the learned GMM model. The white ball represents the actual position of the robot's end-effector. The blue ball represents the believed position of the robot's end-effector 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).

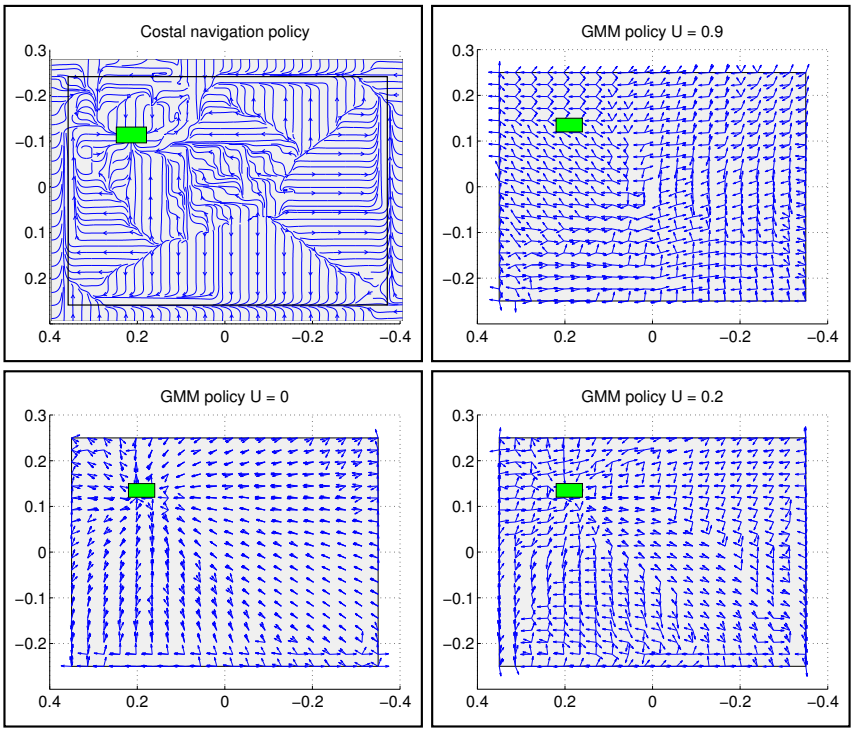

Figure 3.12: Illustration of the vector field for the Coastal and GMM policy. *Top Left* Coastal policy, there is only one possible direction for every state at any time, the values of λ_2 in the cost function were set experimentally. *Others:* The GMM policy for three different levels of uncertainty. For each point multiple actions are possible which is reflected by the number of arrows (only the first three most likely actions). As the uncertainty decreases the policy becomes less multi-model, but remains around the edges and corners. Note that once certain of being close to an edge there is a possibility to go either straight to the goal or stay close to the edge and corners.

3.6.3 Distance efficiency & Uncertainty

We seek to distinguish the most efficient method in terms of two metrics, the distance (in meters) taken to reach the goal and the level of uncertainty upon arriving at the goal. We report results on 5 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 3.3, *top right*) reflecting the default setting. The initializations (both real and believed end-effector locations) for the remaining 4 experiments were chosen in order to reflect particular situations which highlight the differences and drawbacks between each respective search method. For the first experiment (Uniform search experiment), a 100 trials were carried out in which the end-effector position and belief were initialized uniformly. As for the other 4 search experiments, 40 separate runs were carried for each of the three algorithms.

Table 3.2 reports the mean and variance of the distance taken (in meters) to reach the goal for each search method for all 5 experiments. We report on an Analysis of Variance (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 5 search experiments. Before performing the ANOVA, we verified that our dependent variable, distance [m] 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 3.3 we report the p-values and F-statistics for an ANOVA on the 5 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

Experiment	**Greedy**	**GMM**	**Coastal**
Uniform	1.54 (0.46)	0.99 (0.14)	1.13 (0.57)
#1	3.02 (0.36)	1.82 (0.23)	3.44 (1.50)
#2	0.80 (0.01)	1.41 (0.14)	0.94 (0.01)
#3	1.14 (0.08)	1.80 (0.17)	2.14 (0.81)
#4	0.75 (0.04)	1.34 (0.07)	0.68 (0.01)

Table 3.2: Mean distance and (variance) taken to reach the goal for 3 methods in 5 experiments. The grey shaded entries correspond to the results of the search algorithm which obtained the fastest time to reach the goal in each type of experiment/search.

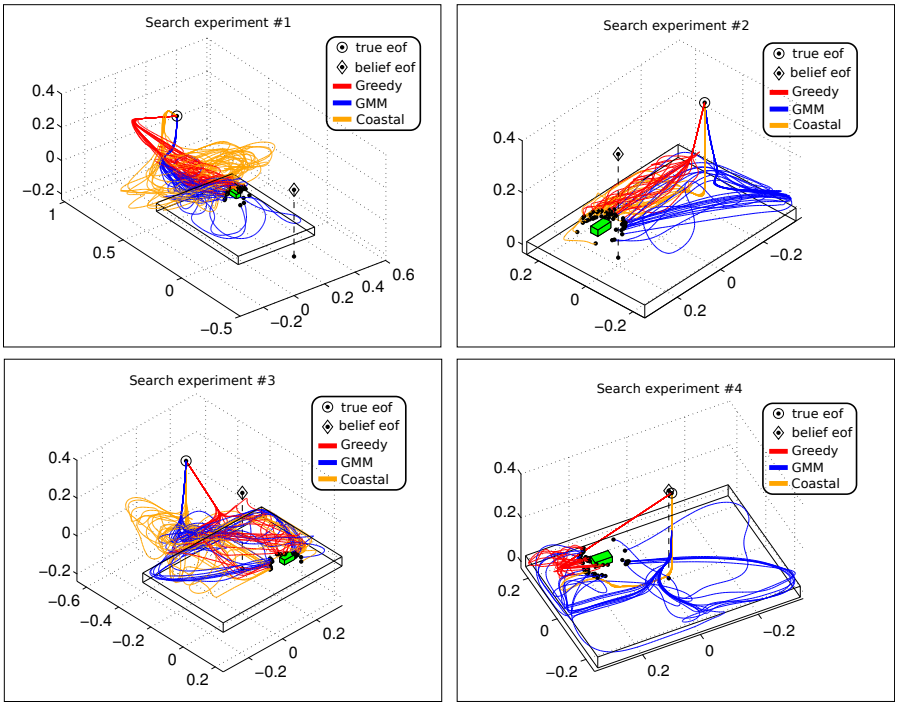

Figure 3.13: Four search initializations, from *top left* to *bottom right* we refer to them as #1-4. The circle indicates the true starting point of the end-effector (eof), whilst the triangle is the initial believed location of the eof. The initialisation in #1 was chosen such that the true and believed eof locations were at opposite sides of the table. This setting was selected to highlight the draw back in methods which do not take into account uncertainty. The second initialisation #2, reflects the situation where once again there is a large distance between true and believed location of the eof. However this time both are above the table. The starting points in #3 are a variant on #1 with the difference being 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 that are inclined to be greedy. Both true and believed eof locations are close to one another.

search method	Uniform	#1	#2	#3	#4
p-value (F)	2e-06 (14)	5e-07 (19)	7e-11 (36)	4-06 (15)	4e-16 (67)

Table 3.3: ANOVA tests the null hypothesis that all search experiments produced the same type of search with respect to the distance taken to reach the goal. All the p-values are extremely small which indicate that the null hypothesis can safely be rejected.

p-value (F)	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)

Table 3.4: ANOVA between paired search methods. 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.

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 3.4 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 #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 3.13 *(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 is 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 remain on the surface of the table. Otherwise 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 search. This leads to the believed position (most likely state, $\hat{x}_t$) varying greatly, resulting in an increased time before the uncertainty has been narrowed down sufficiently for a contact to occur

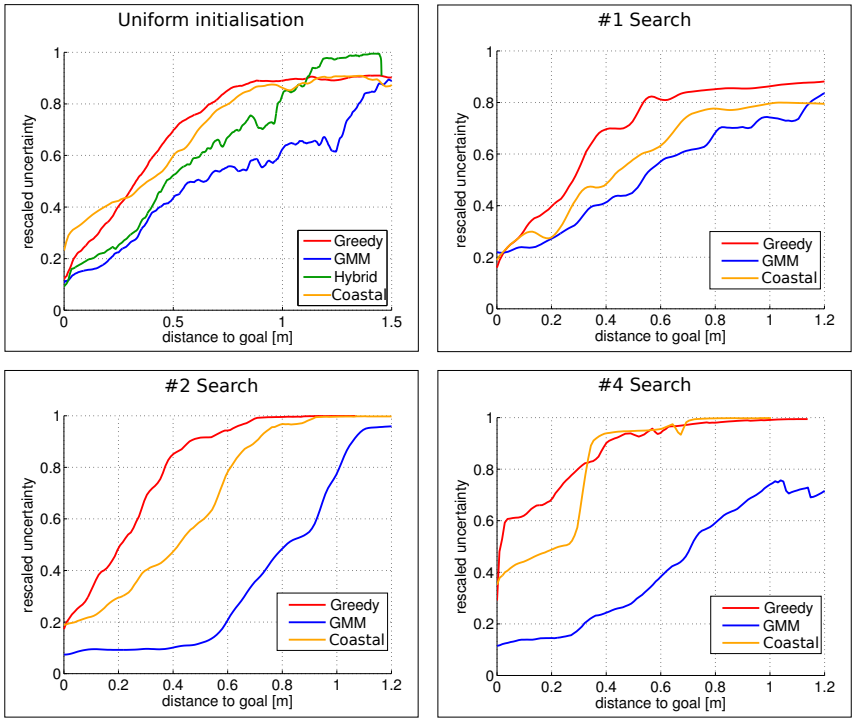

Figure 3.14: Reduction of the uncertainty for the Uniform, #1, #2 and #4 experiment, the expected value is reported *Top left*: Uniform initialisation, expected uncertainty for the Greedy (red), GMM (blue), Hybrid (green) & Coastal (orange) search strategies. *Top right:* Experiment #1. *Bottom left:* Experiment #2. *Bottom right:* Experiment #4.

with the table (or simply by chance).

Figure 3.14 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 find 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 it takes reach the goal in comparison with the other methods.

While the Greedy (#2) and Coastal (#4) are faster than the GMM method, Table 3.2, 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.

3.7 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 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 trajectories which both remained near to the edge and corner features but also trajectories which remained at a distance. 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 89-88% risk-averse.

In terms of the comparative study, the GMM controller is more adapted to dealing with situations of high uncertainty and accounts for it better than Greedy or Coastal planning approaches. 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 Bake et al. (2011) for discrete states and actions).

A drawback of the PbD-POMDP approach is that the quality of the learned policy is dependent on the abilities of the human teacher. If the teacher is good (on average) then the transferred policy will be adequate, if however the human is suboptimal at performing the task, then the resulting policy will be poor. An autonomous way of evaluating the quality of the demonstrations whilst learning a policy is necessary. In the next chapter, "Chapter 4", we demonstrate that by introducing a cost function and using a Reinforcement Learning approach we can account for poor demonstrations and increase the quality of the policy.