PaCMan TR-2015-2

Informational Reward Planning for Dexterous Grasping in Unstructured Environments

C. Zito and J.L. Wyatt*

School of Computer Science, University of Birmingham, UK

Abstract

Dexterous grasping of objects with uncertain pose is a hard unsolved problem in robotics. In this work we extend our proposed solution to this problem using information gain re-planning. We present a new set of algorithms that enable: i) re-planning while maintaining the configuration of the robot in contact with the object when an unexpected contact occurs, ii) planning of dexterous grasping trajectories for non-convex object shapes by implementing efficient collision detection for point clouds, and iii) use of an active complaint controller to allow continuous contact while minimising the risk of perturbing the object being grasped.

First we show how tactile information, acquired during a failed attempt to grasp an object can be used to refine the estimate of that object's pose. Second, we show how this information can be used to re-plan new reach to grasp trajectories for successive grasp attempts. Third, we show how complex, dexterous grasping trajectories can be efficiently planned for a non-convex object described as a point cloud. Finally we present *IR3ne*, our information reward based algorithm, to show how reach-to-grasp trajectories can be modified, so that they maximise the expected tactile information gain, while simultaneously delivering the hand to the grasp configuration that is most likely to succeed.

The method is demonstrated in trials with a simulated robot. Sequential re-planning is shown to achieve a greater success rate than single grasp attempts, and IR3ne, with its trajectories that maximise information gain, requires fewer re-planning iterations than conventional planning methods before a grasp is achieved.

Keywords

Dexterous grasping, planning, informational reward, object-pose uncertainty, object-shape uncertainty, pose density from point cloud, collision detection for point clouds.

1. Introduction

In robot grasping, there is typically uncertainty associated with the location of the object to be grasped. This can be due to poor results of vision or depth sensing, or to simply due to incomplete information about object shape caused by incomplete view coverage. If the object is not in its expected location, then a robot equipped with tactile sensors, or torque sensors at finger joints, may gain some additional information to help refine localisation knowledge from tactile contacts (or lack of thereof) occurring during the execution of a reach to grasp trajectory.

* Corresponding author; e-mail: {cxz004, jeremy.l.wyatt}@cs.bham.ac.uk

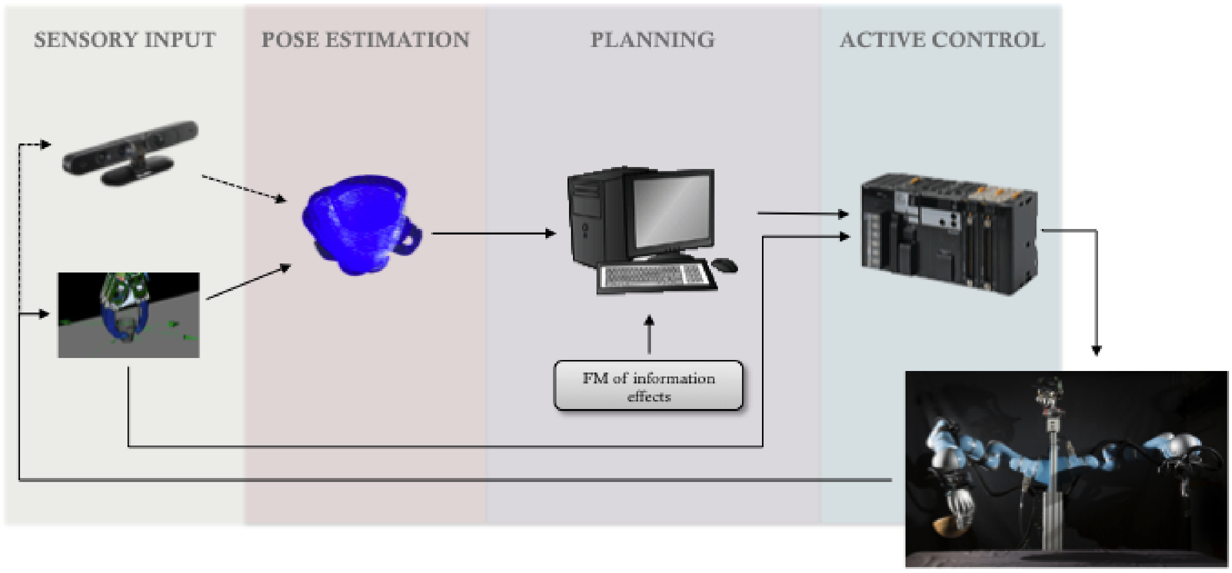

Fig. 1. The architecture of our system is composed of 4 components: sensory input, pose estimation, motion planning and active control. Solid lines highlight the components involved in the re-planning procedure, whilst dashed lines identify interactions that happen only at the first iteration. We employ a depth camera to obtain an incomplete point cloud of the object surface. Using a model fitting procedure a probability density over the object pose is estimated, represented as a particle set. Given this distribution, a trajectory is planned. A forward model (FM) is used to maximise the chance of gathering tactile observations that will reduce pose uncertainty. If unexpected observations occur while the trajectory is executed, then we use them to perform a particle filter update, and the active control enables the robot to keep the contact without applying eccessive force. Re-planning then occurs with the new pose distribution, and a new reach-to-grasp trajectory is constructed. This process is then iterated until a successful grasp is achieved.

In this paper we: i) describe how to efficiently plan collision free, dexterous grasping trajectories for a non-convex object represented as point cloud, ii) describe how to iteratively update the object pose estimate using tactile observations, iii) show how successive grasp trajectories can be planned with respect to these iteratively refined object poses, and iv) show how each reach-to-grasp trajectory can be deliberately planned to maximise information gain from potential tactile contacts, while also planning to terminate at the target grasp given the object is at some assumed pose. This work is an extension of our early work presented in [Zito et al., 2012; 2013].

We achieve these advances by: i) using a hierarchical sample-based path planner, here a Probabilistic Roadmap (PRM) planner, which encodes expected information gain (in a low-dimensional belief space) for each of its trajectory segments, building on the work of [Platt et al., 2001] and [Zito et al., 2013]; ii) refining the belief state for the object pose using an observation model for contact sensing by a multi-finger hand. We show that this approach enables planning for a robot manipulator with 20 DoFs and non-Gaussian object pose uncertainty in 6 dimensions.

We make several assumptions. First we assume a reference model for the object is known. Nevertheless, completeness in this model is not necessary for our algorithm to work. Incomplete point clouds can be used as models. Second we assume that a separate sequence of views given by a depth camera give an incomplete point cloud which can be aligned to this model. The shape incompleteness of both model and observation of the object results in object-pose uncertainty encoded in a belief density for the pose. Given this our approach selects the optimal reach to grasp trajectory with respect to the current belief state. Second we assume that the algorithm takes as input a target grasp configuration on the object (i.e. the set of finger contacts on the model), and a pre-grasp configuration. We use the method described in [**?**] to generate this target grasp for the presented object. Third we assume that the object is stationary and we actively control the compliance of the robot when in contact with the object without altering the object pose. Fourth we assume the availability of a belief state estimate of the object's pose in the form of a particle filter.

The architecture of our system is shown in Fig. 1. In our scenario we employ a depth image obtained from an Asus Xtion Pro depth camera. This gives an incomplete point cloud of the object surface. Using a model fitting procedure similar to the sampling from surflet pairs method presented in [Hillenbrand & Fuchs, 2011], a probability density (or belief state) over the object pose is estimated, represented as a particle set. Given this distribution, a reach-to-grasp trajectory is planned. This trajectory has as its goal configuration the pre-computed grasp under the assumption that the object is at a pose corresponding to the mean position of the particle set. The path to this goal configuration is found using a stochastic motion planner. This planner works with a cost function that allows deviations from the shortest path that maximise the chance of gathering tactile observations that will reduce pose uncertainty in the object location. If unexpected observations occur during the execution of the planned trajectory, then such observations (both tactile contact and no-contact signals) collected at poses along the reach-to-grasp trajectory are used to perform a particle filter update. Re-planning then occurs with the new pose distribution, and a new reach-to-grasp trajectory can be constructed. This process is then iterated until a successful grasp is achieved.

The benefit of planning with beliefs is the ability to reason about the informational effects of sequences of actions. In typical belief space planning this means performing a kind of pre-posterior analysis, in which planned actions (here trajectories) cause imagined observations that are in turn used to perform a Bayes' update of the belief state. As the belief space grows exponentially in the length of the planned action-observation sequence these methods are exact but inefficient.

Our work instead builds on the approach of Platt et al. [Platt et al., 2001; 2012]. That work plans a sequence of actions that will generate observations that distinguish a hypothesised state from competing hypotheses while also reaching a goal position. The informational value of a trajectory is the difference in the expected observations between the hypothesised position and each alternative. This approach allows us to track high-dimensional belief states using an accurate filter defined by the user, but reduces the complexity of planning in belief spaces by approximating the informational value of actions from a low-dimensional subspace of the belief state. Platt et al. applied this to planning for a two degree of freedom manipulator using a laser range finder for observations, and employed an optimisation framework for planning. The algorithm is proved to localise the true state of the system in 1 dimension and to reach a goal region with high probability. In contrast to [Platt et al., 2001], our approach encodes information gathering actions to localise an object to be grasped in 6 dimensions while simultaneously attempting to achieve the grasp. Similarly to [Platt et al., 2001; 2012], our method is guaranteed to converge to the true state of the system in which a reach-to-grasp trajectory succeeds with high probability. Nevertheless, convergence is guaranteed only if the system is not perturbed. In our previous work [Zito et al., 2012; 2013], we thus assumed that the system was i) static (does not change over time) and ii) the robot's palpation skills were sufficiently accurate to retrieve a contact without altering the configuration of the object. For this work we relax the latter assumption by developing an active compliant controller which enables the manipulator to work as a non-linear spring when in contact with the object. We now briefly survey other work on information gathering while grasping.

2. Related work

Other authors have attempted to pose the simultaneous grasping and localisation problem (SLAG) in terms of a partially observable Markov decision process (POMDP). Hsiao et al., [Hsiao et al., 2007], present a simple blocks world problem, wherein a single finger can execute a small selection of actions (left, right, up, down) in response to a small set of possible sensory detections (contact or no contact below, right or left sides of finger). This simple problem is tractable for solving as a POMDP, producing optimal policies for guiding the finger towards a desired location on a 2D blocks world object. The same authors later addressed real-world grasping with an arm and three-finger Barrett hand equipped with tactile fingertip sensors, [Hsiao & Kaelbling, 2010], [Hsiao et al., 2011]. Because the space of possible actions for such a robot is enormous (especially compared to the single finger and 2D blocks-world problem of [Hsiao et al., 2007]), in order to solve the problem as a POMDP, the authors restrict the robot's choice of actions to executing a small number of pre-programmed reach-to-grasp motions, described relative to the pose of the target object. Thus the POMDP method can tractably be used to select between this small number of actions, in response to tactile contacts detected by the three fingertip sensors during the previous action. Thus various actions are sequentially selected, with successive refinements of the object pose collected from sensing during each action, until eventually one of the actions achieves a successful grasp.

In contrast to [Hsiao & Kaelbling, 2010], our method does not select between pre-defined high level actions, but instead we describe how to optimally plan a novel dexterous reach-to-grasp trajectory, in response to recent sensory observations, which maximises the expected information gain from further potential tactile observations.

Petrovskaya [Petrovskaya & Khatib, 2011] also investigated the use of tactile information, derived from collisions of an end effector with an object, to refine localisation estimates for that object. Similarly to our method, Petrovskaya represents the location of an object to be grasped, as a collection of particles forming a distribution, and this distribution is refined by successive manipulator contacts with the object. However, the work is limited in that Petrovskaya does not address the problem of planning manipulator trajectories to achieve these contacts. Instead, the author simply begins with an assumption that a selection of collisions will occur, and then shows how to use such collisions to update an object location distribution.

More recently, another probabilistic framework for sensor-based grasp planning has been presented in [Nikandrova et al., 2013]. Their approach philosophy differs from ours in the sense that the uncertainty in the pose of the object affects only the choice of the (final) grasp configuration of the robot - in terms of set of finger contacts and wrist pose - which maximises some criterions of stability. The reach to grasp trajectory is compute using a state-of-the-art planner which assume the estimated pose to be accurate. They have proposed two strategies: either based on i) stability maximisation or ii) entropy minimisation. The former approach is implemented as a trial and error procedure in which tactile measurements, at the end of the grasp trajectory, are used to update the belief state - represented as a particle filter - and then the robot is move to a fix position at the beginning of each iteration. The latter approach selects the most informative grasp to reduce entropy and fasten the convergency to a stable grasp. However this requires a pre-posterior analysis which is expensive as the belief space grows. Both approaches have been evaluated in simulation with a proof of concept on a real robotic platform, a Melfa RV-3SB 6DoFs arm and a Robotic 3-finger hand. However the experimental setup consists of a simplify 2D problem in which a 2D Gaussian noise is manually added to the object estimate. In contrast our method can cope with non-Gaussian uncertainty in 6 dimensions generated directly from the visual sensory data.

Prentice et al. [Prentice & Roy, 2008] have shown that motion planning for a mobile robot in belief space can be done efficiently for linear Gaussian systems by using a factored form of the covariance matrix. The authors incorporate an estimation of localisation uncertainty in the cost function of a belief-space variant of the PRM. Similarly to [Prentice & Roy, 2008] our approach is capable of balancing shorter paths against those which reduce belief uncertainty through sensory information gain. Instead our work plans trajectories that are able to distinguish hypotheses by using the expected sequence of observations, which are computed according to the current belief state. Furthermore our approach represents the belief space as non-Gaussian and allows arbitrary implementations of Bayes filters to track belief states.

3. Planning trajectory

3.1. Problem formulation

This section is concerned with the problem of planning control actions to reach a goal state in the presence of incomplete or noisy observations. Let us consider a discrete-time system with continuous non-linear deterministic dynamics,

$$x_{t+1} = f(x_t, u_t)$$

where $x_t \in \mathbb{R}^n$ is a configuration state of the robot and $u_t \in \mathbb{R}^l$ is a action vector, both parametrised over time $t \in \{1, 2, ...\}$. Let $p \in SE(3)$ describe the object pose, given an initial prior belief state b_1 we define a set of k hypotheses as $\{p^i\}_{i=1}^k$, where $p^1 = \arg\max b_1$ and $p^i \sim b_1, i \in [2, k]$. We search for a sequence of actions, $u_{1:T-1} = \{u_1, \ldots, u_{T-1}\}$, that distinguish between observations that would occur if the object were in p^1 from any other p^i pose, with $i \in [2, k]$. At each time step, t, the system makes an observation, $y \in \mathbb{R}^m$, that is a non-linear stochastic function of states and hypotheses. Without losing generality, we define y_t to be a column vector of binary values. Each of these values represents whether or not a contact is observed between a given link of the robot and the hypothesis p^i. However, binary values have been shown to be not very informative during the planning phase. Therefore we define,

$$h(x, p^i) = p(y = 1|x, p^i)$$

as a column vector of scores identifying the likehood of observing a contact, $y = 1$, as function of states and hypotheses. More generally, let $F_t(x, u_{1:t-1})$ be the robot configuration at time t if the system begins at state x and takes action $u_{1:t-1}$. Therefore the expected sequence of observations over a trajectory, $u_{1:t-1}$, is:

$$h_t(x, u_{1:t-1}, p^i) = (h(F_2(x, u_1), p^i)^T, h(F_3(x, u_{1:2}), p^i)^T, \ldots, h(F_t(x, u_{1:t-1}), p^i)^T)^T$$

a column vector which describes the likelihood of observing a contact at any time step of the trajectory $u_{1:t-1}$. We then search for a sequence of actions which maximise the difference between observations that are expected to happen in the sampled states, $p^{2:k}$, when the system is actually in the most likely hypothesis, p^1. In other words, we want to find a sequence of action, $u_{1:T-1}$, that minimises

$$J(x, u_{1:T-1}, p^{1:k}) = \sum_{i=2}^{k} N(h(x, u_{1:T-1}, p^i)|h(x, u_{1:T-1}, p^1), \mathbb{Q}) \tag{1}$$

where $N(\cdot|\mu, \Sigma)$ denotes the Gaussian distribution with mean μ and covariance Σ and $\mathbb{Q}$ is the block diagonal of measurement noise covariance matrices of appropriate size. Rather than optimising (1) we follow the suggested simplifications described in [Platt et al., 2001] by dropping the normalisation factor in the Gaussian and optimising the exponential factor only. Let us define for any $i \in [2, k]$

$$\Phi(x, u_{1:T-1}, p^i) = ||h_t(x, u_{1:T-1}, p^i) - h_t(x, u_{1:T-1}, p^1)||^2_{\mathbb{Q}},$$

then the modified cost function is

$$J(x, u_{1:T-1}, p^{1:k}) = \frac{1}{k} \sum_{i=2}^{k} e^{-\Phi(x, u_{1:T-1}, p^i)} \tag{2}$$

it is worth noting that when there is a significant difference between the sequence of expected observations, $h_t(x, u_{1:T-1}, p^i)$ and $h_t(x, u_{1:T-1}, p^1)$, the function $\Phi(\cdot)$ is large and therefore $J(\cdot)$ is small. On the other hand if the sequences of expected observations are very similar to each other, their distance measurement tends to 0 and $J(\cdot)$ tends to 1. Equation (2) can be minimised using different planning techniques such as Rapidly-exploring Random Trees (RRTs) [LaValle, 1998], Probabilistic Roadmap (PRM) [Kavraki & Svestka, 1996], Differential Dynamic Programming (DDP) [Jacobson & Mayne, 1970] or Sequential Dynamic Programming (SDP) [Betts, 2001]. In Section 4.3, we define a new set of heuristics that can be encoded in a PRM planner for generating more reliable trajectories that explicitly reason over the pose uncertainty, and demonstrate the method with experiments on a virtual testbed.

3.2. Belief Update

After a trajectory is planned our algorithm executes it. If an unexpected observation occurs at execution-time the algorithm refines the current belief state using an accurate, high-dimensional filter provided by the user.

In order to define an unexpected observation, it may be convenient for the reader to think of a reach-to-grasp trajectory as composed of two parts: i) an approach trajectory which leads to a pre-grasp configuration of the robot in which the fingers generally cage the object to be grasped without generating any contact, and ii) a grasping trajectory which moves the robot in contact and eventually generates a force closure grasp. In this way any contact which occurs during the approaching trajectory is considered as an unexpected observation. Similarly a insufficient number of contacts to create a force closure at the end of a grasping trajectory is considered as unexpected.

In our implementation we update the belief using Bayes rule assuming deterministic dynamics. In this case we can write the belief update as

$$b_{t+1} = \frac{P(y_{t+1}|x_t, u_t)b_t}{P(y_{t+1})} \tag{3}$$

3.3. Re-planning

Our algorithm plans trajectories assuming only the maximum likelihood observations given the current belief state. Therefore we need to rely on sensory feedback during the execution of the planned trajectory in order to detect whether or not unexpected observations occur. This triggers a belief update, using the observation gathered at execution-time, and consequently a re-planning phase.

In our previous work [Zito et al., 2012; 2013] the manipulator was moved back to a safe configuration (e.g. outside the uncertain region) and before a new reach-to-grasp trajectory was planned. This approach was necessary for technical reasons due to the robotic platform in use. More precisely, our approach was tested on DLR's Rollin Justin robot. The only way to communicate with this platform is via its own controller. This controller accepts trajectories with extra parameters to set, i.e., compliance or activate/disactivate joint's torques thresholds (or guards) to detect contacts minimising the risk of moving the object. However these parameters cannot be changed on-line during the execution of the trajectory and the controller forbids any movement once that an active guard has been triggered. It is possible however to send trajectory disabling guards. Therefore our proposed solution was to withdraw the robot to a safe configuration with no active guard before the next reach to grasp attempt. This however, is inefficient and fails to exploit the existing contact in completing the grasp.

To overcome these limitations we implemented our latest algorithms for our robotic platform called Boris. Our controller for Boris enables us to modify settings on the fly at execution time. This allows us to make a contact with an object, stop the robot, and then re-plan from the same configuration, maintaining contact with the object.

In our experiments, the algorithm uses torque sensors based on current draw at each joint of the robot's hand to detect whether or not a link of the hand is in contact with the environment. We assume that the sensing abilities of the robot are fine enough to perceive the object without moving it. However even in simulation is difficult to maintain such an assumption. Though our results show that small changes in the configuration of the object do not affect the algorithm which is still able to converge to a force closure grasp.

3.4. Terminal conditions

Our algorithm terminates its execution when no unexpected contacts occur and the target grasp is achieved. Since we do not have mesh models of the object to be grasped we cannot rely on such measures to signal successful termination of the algorithm. Nonetheless, in simulation it is possible to measure the displacement error between the grasp configuration for the final object-pose estimate and the ground truth. An user-defined threshold of tolerance is used to identify whether the grasp has succeeded.

4. Implementation

Our approach consists in planning informative tactile observations for a dexterous hand while simultaneously reaching a given target grasp. As described previously, our innovations are i) implementing a hierarchical PRM algorithm which allows us to plan dexterous reach-to-grasp trajectories, ii) encoding a new set of heuristics for a randomised motion planner and iii) formalising an observational model for contact sensing by a multi-finger hand. We tested our approach for a robotic manipulator with 21 DoF under non-Gaussian object pose uncertainty in 6 dimensions.

4.1. *Observation model*

We assume that a robotic manipulator is composed of parts. These parts are considered collections (or chains) of joints, linked together. Without loss of generality, we also assume a single part called the'arm' and a set of m parts called 'fingers'. In addition, we describe the observational model as limited only to a given subset of those parts, i.e. only fingers or a subset of them (e.g. finger tips). Let M be the ordered set of parts which compose the manipulator, then $x(j)$ is the configuration in joint space of the j^{th} part, with $j \in M$. In other words, j is the index of a specific chain. We also define $\bar{M}$ to be the set of indices such that the respective part is used in our observation model. In addition, we use the operator $W(x(j))$ to refer to the workspace coordinates in $SE(3)$ of the j^{th} joint w.r.t. a given reference frame.

Mathematically we formalise the likelihood of observing a contact for each finger of the robot as an exponential distribution over the Euclidean distance, d_{ji}, between the finger tip's pose, $W(x(j))$, and the closest surface of the object assumed to be in pose p^i. Note that, for this work, we limited our observation model to contacts which may occur on the internal surface of fingers. This directly affects our planner which rewards trajectories that would generate contacts on the finger tips rather than on the back side of the fingers. Therefore for any $j \in \bar{M}$ we write

$$p(y(j) = 1|x(j), p^i) = \begin{cases} \eta \exp(-\lambda d_{ji}) & \text{if } d_{ji} \leq d_{max} \\ & \text{and } \langle n_{xj}, \hat{n_{p^i}} \rangle < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\langle n_{xj}, \hat{n}_{p^i} \rangle$ is the inner product of, respectively, the j^{th} finger tip's normal and the estimated object surface's normal, and d_{max} describes a maximum range in which the likelihood of reading a contact is not zero, and η is a normaliser. This allows us to rewrite the likelihood of reading a contact on the force/torque sensors of the robot, $h(x, p^i), i \in [1, k]$ with $j_1, \ldots, j_m \in \bar{M}$ as follows,

$$h(x, p^i) = [p(y(j_1) = 1|x(j_1), p^i), \ldots, p(y(j_n) = 1|x(j_m), p^i)]^T$$

4.2. *Planning a dexterous grasping trajectory for non-convex objects*

The implementation of our planner uses a modified version of Probabilistic Roadmap (PRM) planning, [Kavraki & Svestka, 1996], to plan trajectories. Crucially, sampling-based approaches like PRMs rely on the ability to reject points that are in collision. Testing collision between simple geometrical primitives, such as planes, spheres or cubes, is substantially faster than between non-convex polyhedra. On the other hand, if an object to be grasped is wrapped in a simple primitive bounding box –to achieve computational efficiency– many grasps would simply be impossible to achieve. Fig 2 shows the case of a grasp over the rim of a jug. This grasp requires the thumb to penetrate inside the the convex hull of object. Thus, even if this configuration does not produce any true collisions, it would be rejected by a "naive" collision detection.

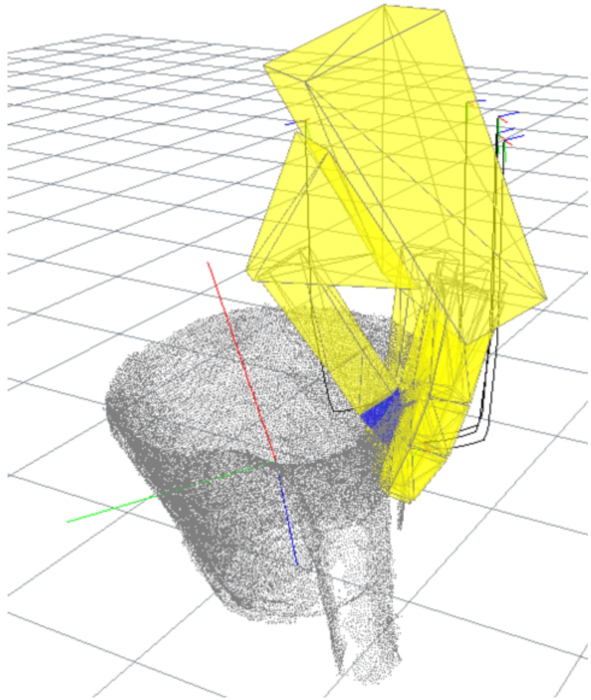

Fig. 2. Rim grasp example on a jug. In this case the target grasp requires to displace the robotic thumb on the internal surface of the object. The grasp configuration is computed using the method described in [**?**].

We propose a fast and efficient collision detection module to cope with object represented by point cloud. Whilst the robot's bodies are represented by a open-chain of convex polyhedra, the object to be grasped is represented by a 2-level structure. The first level of this structure wraps the point cloud in a bounding box. This level can efficiently avoid checking collisions between the robot's links and the object to be grasp that are far apart. The second level contains the point cloud organised in a KD-Tree. Our KD-Tree implementation is based on the FLANN library [Muja & Lowe, 2014]. An example is shown in Fig. 3.

In the planning process we target the object as if it were in its expected location (the mean pose of our density function), therefore only the mean pose is used for collision detection. When at least one bounding box of the robot collides with the bounding box of the object, the second level of the collision detection module is called to determine whether this particular configuration of the robot can be used in the PRM as either a node or in a path to connect two nodes. By querying the KD-Tree it is possible to retrieve the closest points on the surface of the object to the robot's links. Each point is then checked to determine whether it collides or not.

Critically the lack of information about the object's shape may affect collision detection, leading to a reach to grasp trajectory which passes trough the object. In a real scenario, however the tactile observation that will be generated will cause the robot to stop and the current belief state to update. In our current implementation we do not reason about the relative likelihood that the unexpected observation is given by a mis-estimation of the object pose versus lack of shape information; we simply update the object-pose uncertainty. As future work we would like to treat this problem as a SLAM problem, which will enable us to integrate the initial lack of shape information into our object model.

4.3. Planning a trajectory to maximise information gain

The PRM method comprises two phases: i) the learning phase, in which a connected graph G of obstacle-free configurations of the robot is generated and, ii) the query phase, in which a path is searched for a given pair of configurations x_{root}, x_{goal}.

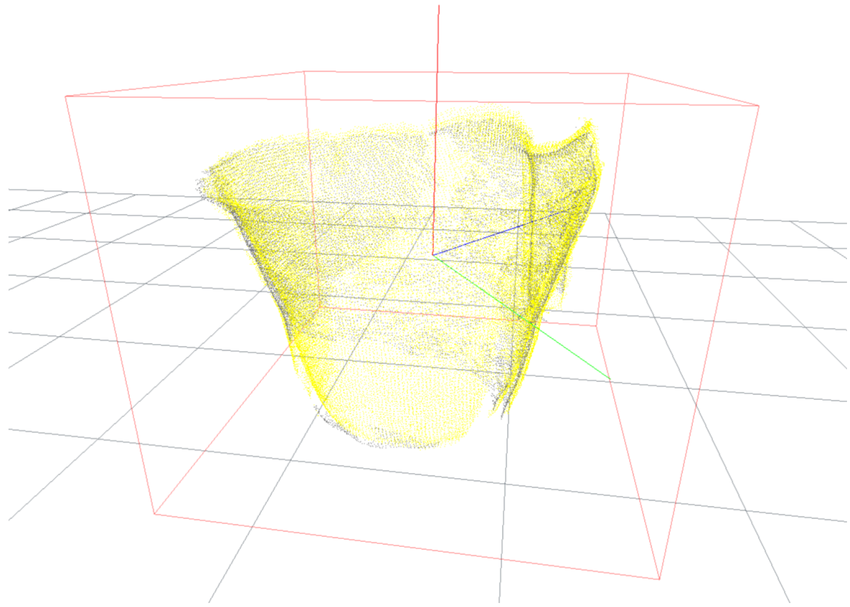

Fig. 3. The image shows the target object to be grasped. In this case the object is a jug. The grey point cloud represents the ground truth pose of the object, as it was acquired by a noiseless input source. The yellow point cloud identifies the best pose estimate. The yellow point cloud is organised as a KD-Tree for faster collision detections. The red box represents the bounding box which avoids unnecessary collision checking between the object and robot's links when they are far apart.

However the computational cost for the learning phase grows fast with respect to the dimensionality of the problem. We therefore incrementally build connections between neighbouring nodes during the query phase. Given a pair $\langle x_{root}, goal\rangle$ which describes the root state in configuration space, $\mathbb{R}^n$, and the goal state in the workspace, $SE(3)$, of the trajectory, we use the A* algorithm to find a minimum cost trajectory in obstacle-free joint space according to:

$$c(x) = c_1(x, x_{root}) + c_2(x, x', \hat{x}_{goal}) \tag{4}$$

where $x, x' \in \mathbb{R}^n$ and $x' \in Neighbour(x)$, $\hat{x}_{goal}$ is a reachable goal configuration for the robot computed by inverse kinematics, $c_1(x, x_{root})$ is the cost-to-reach x from x_{root}, and $c_2(x, x', \hat{x}_{goal})$ is a linear combination of the cost-to-go from x to a neighbouring node x' and the expected cost-to-go from x' to the target. We implemented $c_1(\cdot)$ as a cumulative discount and rewarded travelled distances. In more detail, we define

$$\begin{aligned} c_2(x, x', \hat{x}_{goal}) = \alpha d_{bound}(x, x') + \beta d(x', \hat{x}_{goal}) \\ + \gamma d_{cfg}(x) \end{aligned} \tag{5}$$

where $\alpha, \beta, \gamma \in \mathbb{R}$, $d(\cdot)$ is a distance function in $SE(3)$ which linearly combines rotational and transitional distances in the workspace[1]. For $d_{bound}(\cdot)$, let $\mathcal{B}_n(r) = \{x \in \mathbb{R}^n | x^T x \leq r^2\}$ and $\mathcal{B}(r_l, r_a) = \{A = [\begin{smallmatrix} R & p \\ 0 & 1 \end{smallmatrix}] \in SE(3) | p^T p \leq r_l^2$ and $1 - \langle Q(R), Q(R)\rangle \leq r_a^2\}$[2] denote repectively the r-ball in $\mathbb{R}^n$ and in $SE(3)$, then $b_{bound}(x, x')$ is defined as

$$d_{bound}(x, x') = \begin{cases} \psi(x, x') & \text{if } W(x) - W(x') \in \mathcal{B}(r_l, r_a) \\ & \text{and } x - x' \in \mathcal{B}_n(r) \\ +\infty & \text{otherwise} \end{cases}$$

[1] For the sake of simplicity, we abuse the mathematical notation by writing $d(x, x')$ instead of d(W(x),W(x')).
[2] We simplified the notation $\mathcal{B}_{SE(3)}(\cdot)$ in $\mathcal{B}(\cdot)$ for pratical reasons.

where $Q(\cdot)$ is the Quaternion operator for $R \in SO(3)$, $\langle q_1, q_2 \rangle$ is the inner product of two quaternions, $r_l, r \in \mathbb{R}$, $r_a \in (0, 1)$, and $\psi(x, x') = \zeta d(x, x') + (1 - \zeta)||x - x'||_2$ with $\zeta \in (0, 1)$. Finally, $d_{cfg}(\cdot)$ is a function which penalises dangerous configurations of the robot (i.e. close to joint limits).

We redefine the heuristic $c_2(\cdot)$ in order to reward informative tactile explorations while attempting to reach the goal state (described as a target configuration of the manipulator).

$$\begin{aligned} \bar{c}_2(x, x', \hat{x}_{goal}, A, p^{1:k}) =& \alpha J(x, x', p^{1:k}) d_{bound}(x, x') \\ &+ \beta d_A(x', \hat{x}_{goal}) + \gamma d_{cfg}(x') \end{aligned} \tag{6}$$

where A is the diagonal covariance matrix of our sampled states, for any column vector $a, \mu \in \mathbb{R}^n$, $d_A(a, \mu) = \sqrt{(a - \mu)^T A^{-1} (a - \mu)}$ is the Mahalanobis distance centered in μ and $J(x, x', p^{1:k}) \in (0, 1]$ is a factor which rewards trajectories with a large difference between expected observations if the object is at the expected location, p^1, versus observations that would be expected if the object is at other poses, $p^{2:k}$, sampled from the distribution of poses associated with the object's positional uncertainty:

$$\bar{J}(x, x', p^{1:k}) = \frac{1}{k - 1} \sum_{i=2}^{k} e^{-\Phi(x, x', p^i)} \tag{7}$$

where:

$$\Phi(x, x', p^i) = ||h_t(x, x', p^i) - h_t(x, x', p^1)||_2$$

for each $i \in [2, k]$ and $h_t(x, x', p^i)$ is sequence of the probabilities of reading a contact travelling from state x to x'. In our implementation $h_t(x, x', p^i) = h(x', p^i)$. In other words, we evaluate the likelihood of making a contact while moving from state x to x' as the likelihood of making a contact only in the next state x'. Note that our current observational model is designed to conserve (6) as in (5) when the likelihood of observing a tactile contact is zero. In fact, for robot configurations in which the distance to the sampled poses is larger than a threshold, d_{max}, the cost function $J(\cdot)$ is equal to 1. However we also encode uncertainty in the second factor of our heuristics, $d_A(\cdot)$, which evaluates the expected distance to the goal configuration. In this way the planner also copes with pose uncertainty at the early stages of the trajectory, when the robot is still too far away from the object to observe any contacts.

4.4. Planning for Dexterous manipulator

In order to compute a dexterous trajectory which allows us to plan movement for both arm and fingers we need to address the 'curse of dimensionality' or, equivalently, increase the number of sampled configurations to properly cover the configuration space.

Our proposed solution is to build a hierarchical planner. First we generate a PRM only in the arm configuration space in order to find a global path between the $x_{root}, \hat{x}_{goal}$. It is worth noticing that in this phase the rest of the joints of the manipulator are interpolated in order to have a smooth passage from x_{root} to $\hat{x}_{goal}$. We then refine the planned trajectory generating a new PRM in the entire configuration space of the manipulator (e.g. arm + hand joint space) along the global path. In other words, we limited the new PRM to explore only the subspace nearby the configurations which compose the global path. Subsequently an optimisation procedure is executed along the trajectory to generate a smoother transition from one configuration to the next.

This approach enables us to plan dexterous reach-to-grasp trajectories up to 21 DoFs with only 1,000 sampled configurations. Note that this is the same order of magnitude that we use in practice for planning trajectories of 6 DoF robot manipulators.

4.5. Belief update

Once a trajectory is executed and a real (unexpected) observation y is detected, we update our belief state according to Bayes' rule. We represent our belief state as a set of N particles $b_t = \{b_t^z\}_{z=1}^N$. In a particle filter fashion we update the weight of each particle b_t^z, $z \in \{1, \ldots, N\}$, as follows

$$p(y|x, b_t^z) = \prod_{j \in \hat{M}} p(y_t(j)|x_t(j), b_t^z)$$

and then we execute a resampling step which generates a posterior distribution b_{t+1} as new set of particles $\{b_{t+1}^z\}_{z=1}^N$.

In simulation we assume that there are no false dectections. However it is possible to distinguish whether or not a contact occurs between the object to be grasped and the robot's end-effector. For example, in case a contact with the environment is detected we skip the belief update step and we move the robot back to a safe configuration before triggering the re-planning.

5. Results

In this work we aim to show that sequential re-planning is capable of achieving higher grasp success rates than single grasp attempts in presence of non-Gaussian object-pose uncertainty in 6 dimensions. We also show that planning trajectories that maximise information gain requires fewer re-planning iterations to achieve a grasp.

We ran 45 trials in a virtual environment. Each trial has a different initial probability density over the object pose. We tested the ability of different strategies to achieve a grasp configuration. The algorithm has a model of the object to be grasped, as a dense point cloud. We collected the object model with a depth camera from 7 different views. These views have been pre-processed to generate a single point cloud of the object model. The pre-processing aligns the single view point clouds, registering and eliminating outlier points. We used the same object for all the trials. Fig. 2 shows the pre-processed point cloud of the plastic jug used as the object model. Using the procedure described in [Kopicki et al., 2014], we computed a possible grasp on the model.

We tested our algorithms under the hypothesis that the same object is displaced in a different position - but still in the dexterous workspace of the robot - and the robot has to attempt a grasp even if the new point cloud is not as dense as the model point cloud. In simulation, we achieve this by applying a rigid body transformation to the single-view point cloud to move it to a different location within the dexterous workspace of the manipulator. We tested our algorithm in three different conditions. In each condition we randomly selected respectively 1, 3 or 7 single-view point clouds from the 7 single-views that composed the model.

Fig 5 summarises the data collected in our experiments. First we computed, for each condition mention above, how much the combination of the single views covered the model surface. All the results are compared with respect to the model coverage in percentage terms. We ran three different algorithms:

- ELEMENTARY: an open-loop trajectory towards the expect object pose without re-planning.
- MYCROFT: a sequential re-planning algorithm without information gathering.
- IR3NE: a sequential re-planning algorithm with information gathering.

For each strategy we present their success rate in the sense of their ability to converging to a grasp. The top left chart in Fig. 5 shows that the sequential re-planning approaches always led to a grasp, while a single grasp attempt often fails in the presence of uncertainty.

We also show the error in the initial estimate of the object pose with respect to the ground truth, which is available in the simulation environment. We decomposed this error into translational and rotational components. The translational component measures displacement as Euclidean distance in a 3 dimensional space between the estimated location and the ground truth. The rotational or angular displacement is evaluated using quaternions. The bottom line of Fig. 5 shows

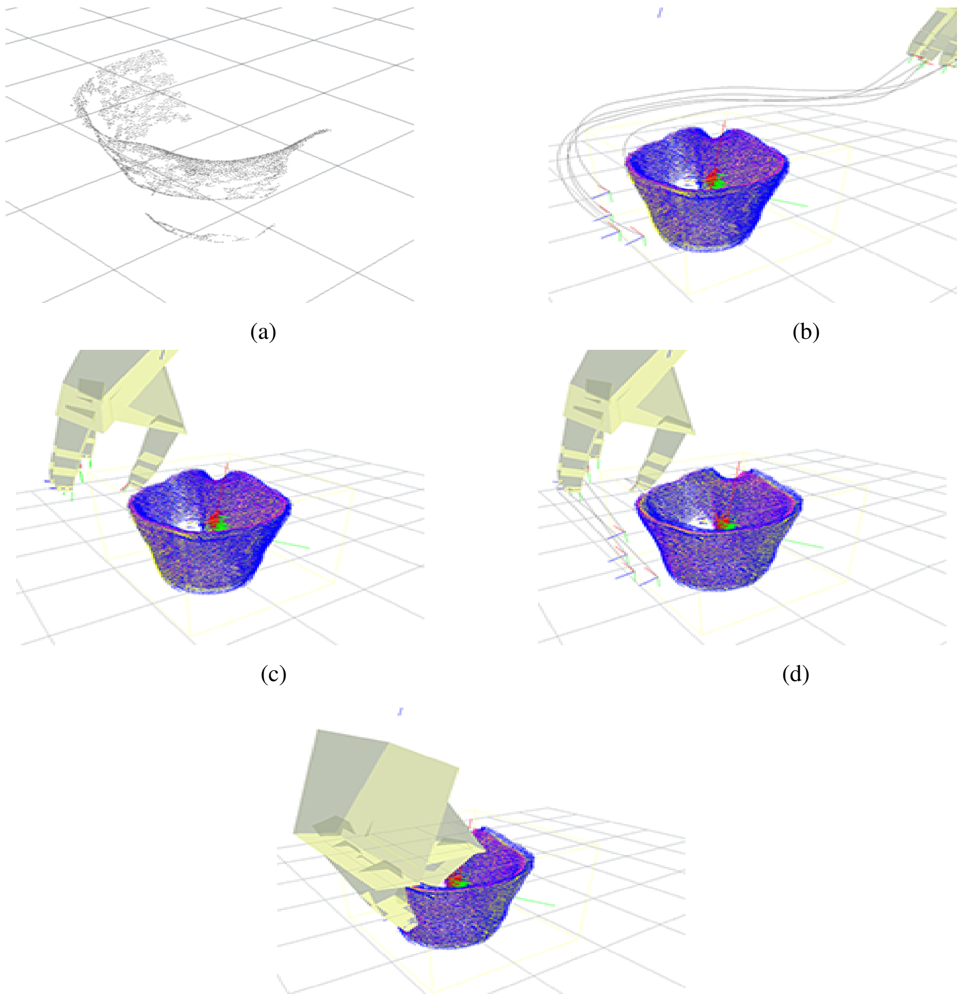

Fig. 4. Example of plan execution for IR3NE. Top row: a partial point cloud is acquired (a), a pose density represented by a set of particles is computed, and a reach-to-grasp trajectory is planned (b). The red point cloud identifies the ground truth which matches the partial point cloud in (a). The blue point clouds are the poses comprising the low dimensional belief state sub-sampled from the corresponding high dimensional belief state. The yellow point cloud represents the estimated pose for the object. Second row: despite the tiny misplacement between the estimated pose and the ground truth, a contact occurs before the grasp configuration with respect to the mean pose (yellow point cloud) could be reached (c). The belief state is therefore updated from the contact (d) and a new trajectory is planned with respect to the new estimate. Bottom row: the hand reaches the target grasp pose (d).

the error reduction, from the initial estimate to the final one, produced by using tactile information acquired during a failed attempt to grasp. Although both sequential re-planning approaches show a similar behaviour it is noticeable that the reach-to-grasp trajectories which maximise information gain (IR3NE) enable us to generate more informative contacts that lead to a better estimate with fewer iterations. Nevertheless, the bottom left chart in Fig. 5 shows that the rotational error seems to increase during the trials. There are two reasons for this: i) the initial rotational uncertainty in our trials was always very low since we chose an asymmetrically shaped object and ii) we stopped the robot at the first contact which most of the time is caused by only a finger, thus this information allows a better estimation in terms of linear accuracy but may reinforce hypotheses in the particle filter with a poor estimate of rotation.

To illustrate the behaviour of our re-planning system we show a typical sequence generated by one simulation trial. We assume a known point cloud model of the object shape, and uncertainty in the object pose. In this trial the robot observes the object as a point cloud and applies a model fitting process described in [Hillenbrand & Fuchs, 2011]. The model fitting process is stochastic and so the resulting pose of the object is uncertain. We sample multiple poses by repeating this process, and obtain a belief state consisting of the resulting set of possible poses of the object. A trajectory is then planned to achieve a given grasp on the object. At each step a trajectory for the wrist and fingers are generated that will move to the desired grasp, while deviating from a minimum length trajectory to maximise information gathered through tactile observations. The belief state is updated and re-planning occurs each time a tactile contact is made. In the example shown in Fig. 4 the initial belief state over the object pose is quite narrow, however an error of few millimetres in the pose estimate leads to a potentially dangerous contact with the object. Instead, we use the contact to update the belief and plan a new trajectory from the current configuration of the robot. The second trajectory transfers successfully the hand to the target grasp configuration.

6. Conclusion and future work

We have shown how to solve the problem of dexterous grasping of objects with uncertain pose by using information gain re-planning. We have proposed a method for tactile information gain planning for dexterous, high DoF manipulators by showing how to: i) efficiently plan collision free, dexterous grasping trajectories for non-convex objects represented as a point cloud; ii) iteratively update localisation knowledge using tactile observations from a previous grasp attempt; iii) use successive grasp trajectories to plan with respect to these iteratively refined object poses; and iv) deliberately plan each reach-to-grasp trajectory to maximise new tactile information gain, while also moving toward the expected grasp location. We have shown that this approach enables planning for a robot manipulator with 20 DoF and non-Gaussian object pose uncertainty in 6 dimensions. We have also shown how sequential re-planning can achieve better quality grasps than single attempts to directly grasp an object at its expected pose, and that re-planning with trajectories designed to maximise tactile information gain, achieves successful grasps with fewer iterations than sequential attempts to move directly towards the object's expected pose.

This work extends that of [Platt et al., 2001], which offers a way to avoid the complexity of planning in a high dimensional belief space. It does this in two ways, i) by approximating the informational value of actions from a low-dimensional subspace of the belief state; and ii) by embedding that informational value into the physical space, creating a non-Euclidean distance metric in that space. This enables standard motion planning techniques to trade off directly between information gain and achievement of the goal pose for the manipulator. We aim to test our approach on a real robotic platform. This will require adaptation of our current observation model to cope with a robotic hand provided not only with current sensing at the joints, but also with force-torque sensors on the finger tips.

References

J. Betts (2001). *Pratical methods for optimal control using nonlinear programming*. Siam.

U. Hillenbrand & A. Fuchs (2011). 'An experimental study of four variants of pose clustering from dense range data'. In *Computer Vision and Image Understanding*.

K. Hsiao & L. Kaelbling (2010). 'Task-Driven Tactile Exploration'. In *IEEE Proc. of Robotics: Science and Systems (RSS)*.

K. Hsiao, et al. (2007). 'Grasping POMDPs'. In *IEEE Proc. Int. Conf. on Robotic and Automation (ICRA)*.

K. Hsiao, et al. (2011). 'Robust Grasping Under Object Pose Uncertainty'. In *Autonomous Robots*, no. 31 in (2-3), pp. 253–268.

D. Jacobson & D. Mayne (1970). *Differential dynamic programming*. Elsevier.

L. Kavraki & P. Svestka (1996). 'Probabilistic roadmaps for path planning in high-dimensional configuration spaces'. In *IEEE Trans. on Robotics and Automation*.

M. Kopicki, et al. (2014). 'Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models'. In *IEEE Proc. Int. Conf. on Robotic and Automation (ICRA)*.

S. M. LaValle (1998). 'Rapidly-exploring random trees: A new tool for path planning'. Tech. rep., Computer Science Dept, Iowa State University.

M. Muja & D. Lowe (2014). 'Scalable Nearest Neighbor Algorithms for High Dimensional Data'. *Pattern Analysis and Machine Intelligence (PAMI)* **36**:2227–2240.

E. Nikandrova, et al. (2013). 'Towards informative sensor-based grasp planning'. *Robotics and Autonomous Systems* pp. 340–354.

A. Petrovskaya & O. Khatib (2011). 'Global localization of objects via touch'. *IEEE Trans. on Robotics* **27**(3):569–585.

R. Platt, et al. (2001). 'Simultanious localization and grasping as a belief space control problem'. Tech. rep., CSAIL, MIT.

R. Platt, et al. (2012). 'Non-Gaussian Belief Space Planning: Correctness and Complexity'. In *IEEE Proc. Int. Conf. on Robotic and Automation (ICRA)*.

S. Prentice & N. Roy (2008). 'The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance'. In *12th Int. Symposium of Robotics Research*.

C. Zito, et al. (2013). 'Sequential trajectory re-planning with tactile information gain for dextrous grasping under object-pose uncertainty'. In *IEEE Proc. Intelligent Robots and Systems (IROS)*.

C. Zito, et al. (2012). 'Exploratory reach-to-grasp trajectories for uncertain object poses.'. In *Proc. Workshop on Beyond Robot Grasping: Modern Approaches for Dynamic Manipulation. Intelligent Robots and Systems (IROS)*.

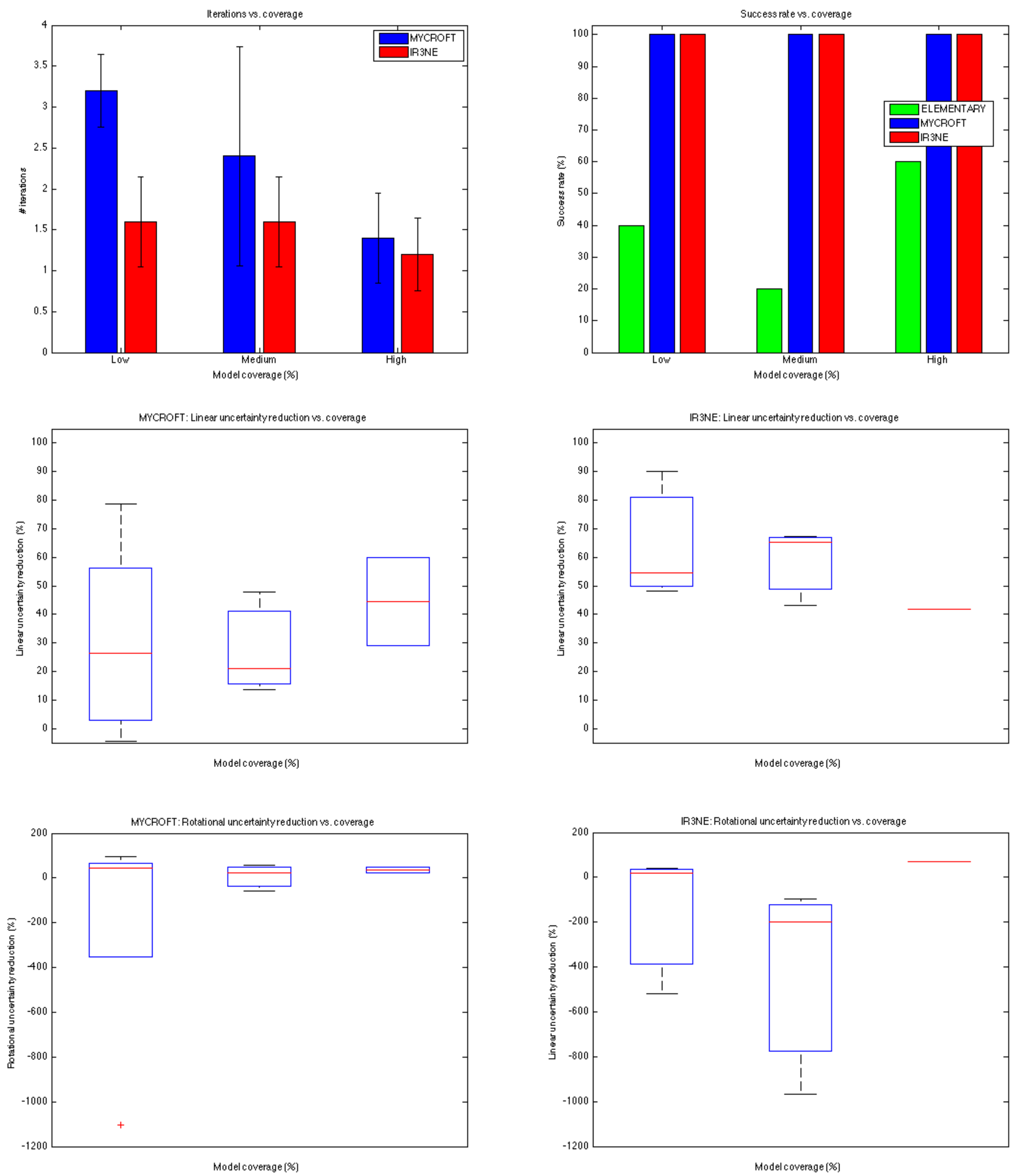

Fig. 5. Simulated results for 45 trials and three strategies: ELEMENTARY (green), MYCROFT (blue), IR3NE (red). All the results are presented plotted against the initial percentage of model coverage given by one or more views by the depth camera. The first chart (top left) presents the average number of iterations required to converge to a grasp. IR3NE shows that maximising information gain requires fewer re-planning iterations to achieve a grasp. The second chart (top right) shows that sequential re-planning is capable of achieving higher success rates than single grasp attempts in presence of uncertainty. The third chart (bottom left) shows how the initial linear uncertainty is reduced by sequential re-planning until the algorithm converges to a grasp. Trajectories that maximise information gain (IR3NE) are capable of localising the objects with a better accuracy. Finally, the last chart (bottom right) presents the reduction in the rotational uncertainty. In this case simple re-planning (MYCROFT) shows better performance, however the rotational uncertainty is usually low for asymmetric objects like the object we used for the experiments.