

# Active Touch for Grasping

Claudio Zito · Valerio Ortenzi · Maxime Adjigble · Marek Kopicki ·  
Rustum Stolkin · Jeremy L. Wyatt

Received: date / Accepted: date

**Abstract** Incomplete object shape representation and imperfect perception are the prime cause of robotic failures, especially when it comes to fine dexterous grasping. In this work, we present a novel solution to such problem. We propose a trade-off between grasping and information gain in that we plan to gather information about the object pose whilst simultaneously attempting to grasp it. Comparatively little attention has been paid to embed information gain strategies in the reach-to-grasp planning, yet there is evidence that the sensorimotor control in humans is affected by pose uncertainty at the very first stages of the reach-to-grasp trajectory.

In this work, we present how to integrate visual and contact clues in a probabilistic fashion to estimate and refine the pose of the object to be grasped. Secondly, we describe how unexpected contacts, acquired during a failed attempt to grasp, can be used to re-plan new reach-to-grasp trajectories for successive grasp attempts. Third, we show how 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

We gratefully acknowledge support of FP7 grant IST-600918, PacMan.

Zito, Ortenzi, Adjigble, Kopicki, Stolkin and Wyatt  
University of Birmingham,  
Edgbaston, Birmingham  
United Kingdom, B15 2TT  
Tel.: +44-121-4144788  
Fax: +44-121-4144281  
E-mail: {claudio.zito.81,jeremy.l.wyatt}@gmail.com  
E-mail: {marek.kopicki, valerio.ortenzi}@gmail.com  
E-mail: R.Stolkin@cs.bham.ac.uk

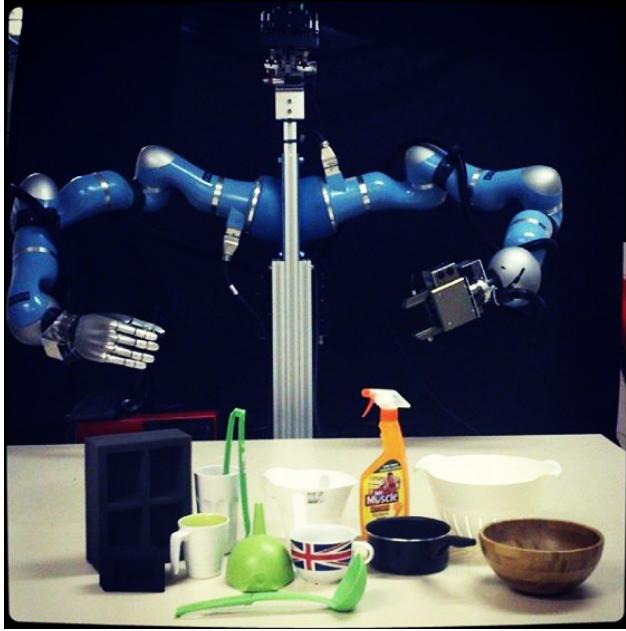
simultaneously delivering the hand to the grasp configuration which is most likely to succeed.

The method is demonstrated in trials on a simulated robot and on a half-humanoid platform called Boris. Sequential re-planning is shown to achieve a higher success rate than single grasp attempts. Also, our algorithm IR3ne requires fewer re-planning iterations with its trajectories maximising information gain than conventional planning methods.

**Keywords** Dexterous Grasping · Information Gain · Active Vision · Active reach to grasp planning

## 1 Introduction

Robot grasping is typically affected by uncertainty associated with the location of the object to be grasped. This is a challenging problem because it requires the robot to find collision-free trajectories that are robust in the face of such uncertainty. There are two fundamentally different approaches. First, a trajectory may be formulated that is robust to current uncertainty, but does not reason about how future information may reduce that uncertainty. Second, the robot may plan a trajectory to gather information that will reduce the uncertainty, so as to make a final grasping trajectory more reliable. Previous work typically separates these two aspects, separately planning information gathering trajectories and grasping trajectories. The two can be theoretically joined in a continuous state and action POMDP, but this leads to an infinite dimensional belief space planning problem that is hard to solve. In this paper we propose and validate a way to combine information gathering and reach-to-grasp trajectories. Our main insight is we can embed the value of information in the much lower dimensional physical space to avoid



**Fig. 1** Boris: half-humanoid robot platform.

the full complexity of belief state planning. This gives a well posed and tractable problem for reach-to-grasp planning under uncertainty. The specific contributions of this work are to:

1. plan information gain whilst simultaneously attempting to grasp the target object;
2. encode expected information gain by warping distances in the workspace, creating a non-Euclidean metric that is information-sensitive;
3. employ a hierarchical planning approach to reduce planning complexity in this space;
4. update the belief about the object's pose using a tactile observation model for a multi-finger hand palpating the object;
5. evaluate different methods, proving that our approach improves reach-to-grasp planning for a dexterous robot.

We use a state of the art grasp generation algorithm to obtain a target grasp on the fly, i.e. a set of finger contacts, recomputing it as we update information about the object pose. We assume a possibly incomplete shape model of the object is previously obtained from sensing. The work is demonstrated in trials in simulation and on Boris, a half-humanoid robot platform. Empirical results confirm that sequential re-planning achieves a greater success rate than single grasp attempts, and that the information gain approach requires fewer iterations before a grasp is achieved.

In the rest of the paper we start with reviewing related work (Section 2), then we describe the problem formulation (Section 3). Successively we delineate the

core information gain planning algorithm on a dexterous robot platform (Section 4), and report the experimental results (Section 5). We finish with concluding remarks (Section 6).

## 2 Related Work

Traditional manipulators were designed to be deployed in industry under perfect knowledge of the environment. Instead, the lack of a complete model of the environment is the main source for uncertainty in robotic manipulation in unstructured environments, hence the necessity of developing robust strategies to act in partial ignorance. Thus the problem of robotic grasping under object pose uncertainty can be decomposed in: state estimation, grasp synthesis, grasp planning and control. A typical approach is to represent the belief state using prior distributions, select a grasp robust w.r.t. uncertainty and finally use tactile feedback to adjust the grasping trajectory, see e.g. (Nikandrova et al. 2013). The reach-to-grasp trajectory is generally computed using some conventional sampling-based techniques which minimise the cost, in Euclidean space, of moving the robot's end effector to the selected grasp configuration. Comparatively little work has explored the more complex problem of reasoning about uncertainty while planning this dexterous reach-to-grasp trajectory. This is mainly due to the high dimensionality of the configuration space of a dexterous manipulator. In this work we consider this problem in depth.

An open-loop grasp trajectory may lead to failure if the object is not in its estimate pose. Thence maintaining a density over the pose of the object yields to probabilistic strategies that are capable of maximising the likelihood of success as well as gathering information if necessary, as in (Hsiao et al. 2011). In (Hillenbrand 2008; Tuzel et al. 2005) object pose estimation is addressed using mean-shift clustering in continuous parameter space. More recently in (Hillenbrand and Fuchs 2011), the authors investigate the relative estimation accuracy and robustness of four variants of the pose clustering algorithm.

Since many robotic problems have multi-modal uncertainty, there are benefits in using non-parametric representation of the belief space, i.e. *particle filter* (PF). There are many examples of PFs used for state estimation in manipulation problems (Petrovskaya and Khatib 2011), (Nikandrova et al. 2013) and (Platt et al. 2001). Nevertheless, most of these methods sample an initial set of particle from an user-defined distribution attached to the *maximum a posteriori* estimate (MAP) obtained from vision, and often the uncertainty is limited to 2D. Yet, at our knowledge, there are no existing

**Table 1** The problem of grasping under uncertainty at a glance.

Problem	Methods	Pros	Cons
State Estimate	ML estimate	- Global estimator - Robust data correction	- No representation of uncertainty - Model-based
	Belief as PDF	- Representation of uncertainty	- Hard to estimate true distribution
	PF	- Non-parametrisation of belief	- Computational expensive
Grasp Synthesis	Analytical	- Pose as optimisation problem	- Computational expensive
	Learn by demonstration	- Require a teacher - Learn task operational space	- Large parameter space - Hard to generalise
	Object-centric learning	- Learn from experimental data - Generalise within and across object categories	- Learn from incomplete and erroneous data
Grasp Planning	Global policies	- Reason on informational effects - Posteriori analysis	- Belief space grows exponentially - Posteriori analysis is expensive
	Local policies	- Reason on informational effects - No posteriori analysis - Computational treatable	- Rely on ML observations - Require re-planning strategies
Grasp Execution	Mechanical	- Robust on collisions - Simple control - Do not need to model uncertainty	- Not adjustable compliance - Under-actuation - Reduced achievable postures
	Feedback control	- Adjustable compliance - Fully actuation	- Limited control bandwidth

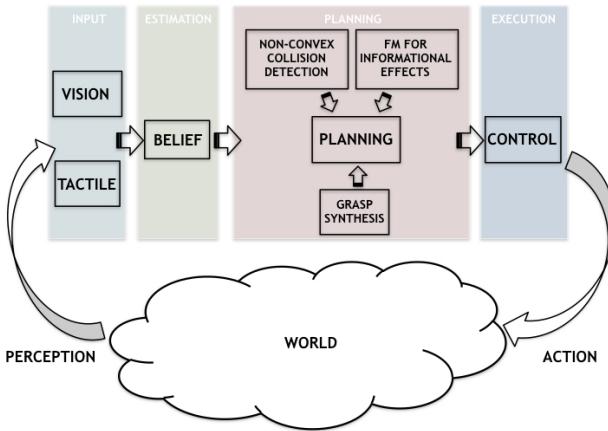
methods for robotic manipulation problems that can efficiently cope with non-Gaussian uncertainty in 6D spaces, i.e. position and rotation. Oppositely, the approach of this work estimates a set of hypotheses from real RGB-D data, and each hypothesis is the result of a ML estimation.

To synthesise grasps for robotic grippers and hands, two categories of approaches can be distinguished (Sahbani et al. 2012): i) analytical and ii) empirical approaches . Analytical approaches are typically associated with an optimisation problem, and thus the computational effort grows with the dimensionality of the grasp solution space which is related to the number of fingers and number of contact points. On the other hand, empirical approaches are centred on observations and learn a mapping between the parameters of the object and the grasp space, as in (Saxena et al. 2008; Detry 2010; Kopicki et al. 2014). Learning this mapping from incomplete or erroneous data remains challenging. In (Detry 2010), the authors address the problem of associating a grasp to a partially perceived object form vision proposing a method to learn a pre-shape configuration of the hand parametrised with respect to the object pose. The grasp is then achieved relying on compliance or tactile sensing.

When probabilistic grasp planning techniques encounter a mismatch in the object expected position, additional information is commonly gained through contacts. Hence the majority of these algorithms pose the grasping problem as a stochastic decision-making problem in belief spaces. Belief planning requires to select later actions taking into account the information gained by earlier actions and vice-versa. However these meth-

ods are inefficient in practice. Typical implementations rely on simplifying the grasping problem by constraining the belief space to Gaussian distributions. Nonetheless Gaussian distributions do not represent fairly the uncertainty for real problems. Extensions to non-parametrisation of belief spaces as in (Nikandrova et al. 2013), typically result in intractable planning problems due to the high-dimensionality of non-Gaussian parametrisation. In contrast, this work is related to the approach of Platt et al. (Platt et al. 2001), (Platt et al. 2012), which allows to construct on-line dexterous grasping trajectories more robust against pose uncertainty as well as to track high-dimensional belief states. In (Platt et al. 2001) a sequence of actions is planned which will generate observations that distinguish a state from competing hypotheses, while also reaching a goal position. This method is applied 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 one dimension and to reach a goal region with high probability. Similarly to (Platt et al. 2001), (Platt et al. 2012), this method is guaranteed to converge to the true state of the system in which a reach-to-grasp trajectory succeeds with high probability, under the assumption that the system is not perturbed by previous grasping attempts. In contrast to (Platt et al. 2001), our approach encodes information gathering actions to localise an object to be grasped in six dimensions while simultaneously attempting to achieve the task of grasping.

Compliant control approaches can also be used to exploit contacts during the execution of reach to grasp



**Fig. 2** The system architecture comprises sensory input, pose estimation, motion planning and active control.

trajectories. One class of approaches is based on mechanical compliance derived by the hand design. These devices are cheaper, simple to control and are able to perform operations with no need to model uncertainty, but they lack controllability. A second class of approaches implement a feedback control to mimic compliance. Because of the limits of the bandwidth of the controller, the controller cannot react fast enough to external forces, thus no shock can be absorbed. In our grasp planner we rely on a feedback controller. This choice is forced by the fact that we need to know with accuracy the pose of our device to refine our beliefs over the object pose using the contacts.

Interestingly, there is also evidence (e.g. METTI UN PAIO DI RIFERIMENTI QUI) that sensorimotor control in humans selects actions in order to maximise some reward function associated with the motor outcome. These models of optimal decision making provide an explanation on how humans compensate for uncertainty and this aspect is very similar to our approach. Table 1 summarises the problem of grasping under uncertainty at glance.

### 3 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,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) \quad (1)$$

where  $\mathbf{x}_t \in \mathbb{R}^n$  is a configuration state of the robot and  $\mathbf{u}_t \in \mathbb{R}^l$  is a action vector, both parametrised over time  $t \in \{1, 2, \dots\}$ . Let  $p \in SE(3)$  describe the object pose,

given an initial prior belief state  $\mathbf{b}_1$  and let us define a set of  $k$  hypotheses as  $\{\mathbf{p}^i\}_{i=1}^k$ , where  $\mathbf{p}^1 = \arg \max \mathbf{b}_1$  and  $\mathbf{p}^i \sim \mathbf{b}_1, i \in [2, k]$ .

We know that in general the problem of planning in belief space is intractable. Instead let us consider a method to search for a sequence of actions,  $\mathbf{u}_{1:T-1} = \{\mathbf{u}_1, \dots, \mathbf{u}_{T-1}\}$ , that distinguish between observations that would occur if the object were in some pose  $\mathbf{p}^1$ , from those that would occur in some other poses  $\mathbf{p}^i$ , with  $i \in [2, k]$ . At each time step,  $t$ , the system will make an observation,  $\mathbf{y} \in \mathbb{R}^m$ , that is a non-linear stochastic function of the joint state of the robot and some object state. 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 object pose hypothesis  $\mathbf{p}^i$ . However, binary values have been shown to be not very informative during the planning phase. Therefore let us define,

$$\mathbf{h}(\mathbf{x}, \mathbf{p}^i) = Pr(y = 1 | \mathbf{x}, \mathbf{p}^i) \quad (2)$$

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

$$\begin{aligned} \mathbf{h}_t(x, \mathbf{u}_{1:t-1}, \mathbf{p}^i) &= (\mathbf{h}(F_2(x, \mathbf{u}_1), \mathbf{p}^i)^T, \\ &\mathbf{h}(F_3(x, \mathbf{u}_{1:2}), \mathbf{p}^i)^T, \dots, \mathbf{h}(F_t(x, \mathbf{u}_{1:t-1}), \mathbf{p}^i)^T)^T \end{aligned} \quad (3)$$

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

$$\begin{aligned} \mathcal{J}(x, \mathbf{u}_{1:T-1}, \mathbf{p}^{1:k}) &= \\ &\sum_{i=2}^k N(\mathbf{h}(x, \mathbf{u}_{1:T-1}, \mathbf{p}^i) | \mathbf{h}(x, \mathbf{u}_{1:T-1}, \mathbf{p}^1), \mathbb{Q}) \end{aligned} \quad (4)$$

where  $N(\cdot | \mu, \Sigma)$  denotes the Gaussian distribution with mean  $\mu$  and covariance  $\Sigma$  and  $\mathbb{Q}$  is the block diagonal of the measurement noise covariance matrix. Rather than optimising (4) 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  $i \in [2, k]$

$$\Phi(\mathbf{x}, \mathbf{u}_{1:T-1}, \mathbf{p}^i) = \|\mathbf{h}_t(x, \mathbf{u}_{1:T-1}, \mathbf{p}^i) - \mathbf{h}_t(x, \mathbf{u}_{1:T-1}, \mathbf{p}^1)\|_Q^2 \quad (5)$$

then the modified cost function is

$$\mathcal{J}(x, \mathbf{u}_{1:T-1}, \mathbf{p}^{1:k}) = \frac{1}{k} \sum_{i=2}^k e^{-\Phi(x, \mathbf{u}_{1:T-1}, \mathbf{p}^i)} \quad (6)$$

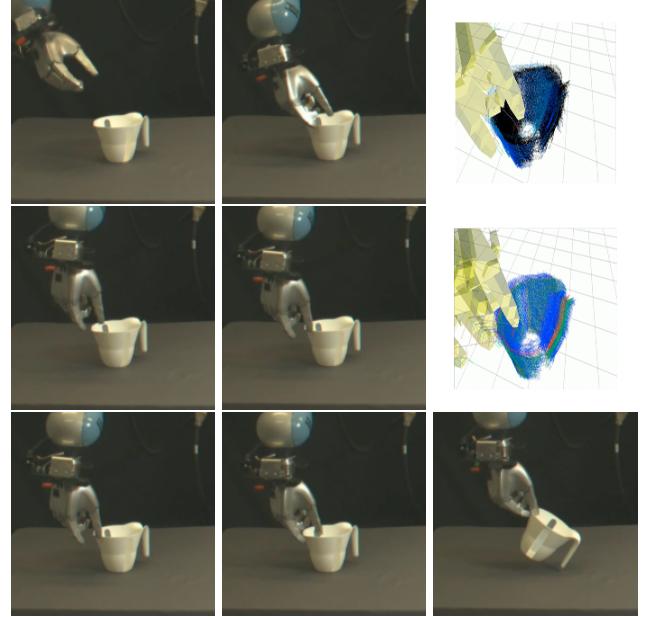
it is worth noting that when there is a significant difference between the sequence of expected observations,  $\mathbf{h}_t(x, \mathbf{u}_{1:T-1}, \mathbf{p}^i)$  and  $\mathbf{h}_t(x, \mathbf{u}_{1:T-1}, \mathbf{p}^1)$ , the function  $\Phi(\cdot)$  is large and therefore  $\mathcal{J}(\cdot)$  is small. On the other hand if the sequence of expected observations are very similar to each other, their distance measurement tends to 0 and  $\mathcal{J}(\cdot)$  tends to 1. Equation (6) can be minimised using different planning techniques such as Randomly-exploring Random Trees (RRTs) (LaValle 1998), Probabilistic Roadmap (PRM) (Kavraki and Svestka 1996), Differential Dynamic Programming (DDP) (Jacobson and Mayne 1970) or Sequential Dynamic Programming (SDP) (Betts 2001). We next use this measure to define a new non-Euclidean cost function that can optimised using any of these methods.

## 4 Information Gain Planning Algorithm

### 4.1 Belief state estimation

We employ a non-parametric representation of the belief state, in this case a particle filter, to model multi-modal uncertainty in object pose. Each particle is the result of a sample-based model-fitting procedure similar to the one presented in (Hillenbrand and Fuchs 2011). This procedure samples a random pair of features from the query point cloud (such as a pair of points with their relative normals) and computes the rigid body transformation to the closest pair of features in the model. Once this set of particles is computed, it is possible to calculate the object's pose estimate by using a clustering algorithm and taking a representative pose from the largest cluster.

This procedure uses the mean-shift algorithm to generate a set of cluster centres from the particle set. Each associated with: i) a score which identifies the “goodness” of the estimate, in terms of how well the estimate represents the entire set of particles, and ii) the number of particles that have contributed to construct the estimate. Let  $c_i$  be the centre particle of a cluster,  $C_i$ , of the entire set of particles  $[p_1, \dots, p_K]$ . Each particle is associated with an importance sampling weight



**Fig. 3** The sequence of images presents an interesting case in which the initial belief state does not cover the ground truth, however IR3ne is capable to converge to a grasp in 3 iterations. The simulated images show the belief update as PF. The colour of each hypothesis is associated with its likelihood, from red (high likelihood) to black (zero likelihood). Top row: due to an erroneous localisation the first grasp attempt collides with the plastic jun on the rim (middle). The contact is used to refine the belief state but, since none of the hypotheses match the contact, the hypotheses have all low probability associated (right image). Middle row: a second attempt fails but this time the contact allows Boris to localise the object and, finally, to grasp it (third row).

$w_k$ , then the score for the cluster  $C_i$  is computed as

$$s_{C_i} = \sum_{k \in [1, \dots, K]} w_k e^{\|c_i - p_k\|_Q} \quad (7)$$

where  $\|a\|_Q$  is defined as  $a^T Q a$  and  $Q$  is the covariance matrix of the set of particle  $[p_1, \dots, p_K]$ . Once a fixed number of clusters  $C_i$  is generated, the procedure agglomerates the most similar clusters, if any, and computes the best estimates as the mean of the most promising cluster  $C_{max}$ , which is defined as:

$$C_{max} = \arg \max_{C_i} \frac{|C_i|}{\sum_j |C_j|} s_{C_i} \quad (8)$$

where  $s_{C_i}$  is the score associated with the cluster  $C_i$ , and  $|C_i|$  is the cardinality of the cluster  $C_i$  in term of number of members.

When tactile observations occur the algorithm refines the current belief state using this particle filter. We think of the reach-to-grasp trajectory as composed of two parts: i) the 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 finger closing trajectory which moves the fingers into contact and generate a force closure grasp. In this way any contact which occurs during the approach trajectory is considered as an unexpected observation. Similarly an insufficient number of contacts for a force closure at the end of a grasping trajectory can also be used as an observation. In our implementation, the belief is updated assuming deterministic dynamics.

#### 4.2 Tactile observation model

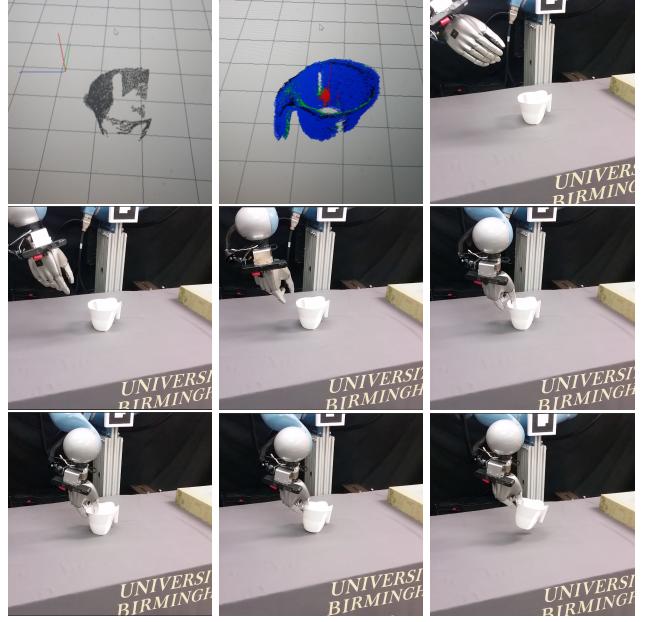
The manipulator is composed of rigid links organised in a kinematic chain and tree, comprising an *arm* and a set of  $\mathbf{M}$  multi-joint *fingers*. In our robot only the final finger phalanges (finger tips) are able to detect a contact with the object. Let  $\mathbf{M}$  be the ordered set of 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 the observational model. In addition, we use the operator  $\mathcal{W}(x(j))$  to refer to the workspace coordinates in  $SE(3)$  of the  $j^{th}$  joint with respect to a given reference frame.

The likelihood of observing a contact for each finger of the robot is an exponential distribution over the Euclidean distance,  $d_{ji}$ , between the finger tip's pose,  $\mathcal{W}(x(j))$ , and the closest surface of the object assumed to be in pose  $\mathbf{p}^i$ . The observation model is limited to contacts which may occur on the internal surface of fingers. This directly affects the 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

$$Pr(y(j) = 1|x(j), \mathbf{p}^i) = \begin{cases} \eta \exp(-\lambda d_{ji}) & \text{if } d_{ji} \leq d_{max} \\ & \text{and } \langle n_{xj}, \hat{n}_{\mathbf{p}^i} \rangle < 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where  $\langle n_{xj}, \hat{n}_{\mathbf{p}^i} \rangle$  is the inner product of, respectively,  $j^{th}$  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,  $\eta$  is a normaliser. That allows us to rewrite the likelihood of reading a contact on the force/torque sensors of the robot,  $\mathbf{h}(x, \mathbf{p}^i)$ ,  $i \in [1, k]$  with  $j_1, \dots, j_m \in \bar{M}$  as follows,

$$\mathbf{h}(x, \mathbf{p}^i) = [Pr(y(j_1) = 1|x(j_1), \mathbf{p}^i), \dots, Pr(y(j_n) = 1|x(j_m), \mathbf{p}^i)]^T \quad (10)$$



**Fig. 4** Example of plan execution for IR3NE. The top row shows: the partial query point cloud (3 merged views), the belief state (as sub-sampled hypotheses (blue), mean pose (green) and ground truth (black)), the real pose of the object. This example shows the worst case in which the query point cloud covers only a 22.5% of the model and the handle is not visible. Note that the ground truth (black point cloud, middle image) is also estimated with the wrong orientation. Nevertheless, IR3NE executes the planned trajectory (middle row) and achieve a grasp. In the right bottom image, Boris lifts successfully the jug.

#### 4.3 Planning a trajectory to maximise information gain

The implementation of this planner uses a modified version of Probabilistic Roadmap (PRM), (Kavraki and Svestka 1996), to plan trajectories and detect collisions. The PRM algorithm is composed of two phases: i) *learning phase*, in which a connected graph  $\mathbf{G}$  of obstacle-free configurations of the robot is generated and, ii) *query phase*, in which a path is searched for a given pair of configurations  $\mathbf{x}_{root}, \mathbf{x}_{goal}$ . However the computational cost for the learning phase grows very fast with respect to the dimensionality of the problem. This planner therefore incrementally builds connections between neighbouring nodes during the query phase. Given a pair  $\langle \mathbf{x}_{root}, \mathbf{x}_{goal} \rangle$  which describes the root state in configuration space,  $\mathbb{R}^n$ , and goal state in workspace,  $SE(3)$ , of the trajectory, this planner uses an A\* algorithm to find a minimum cost trajectory in obstacle-free joint space according to:

$$c(x) = c_1(x, \mathbf{x}_{root}) + c_2(x, x', \hat{x}_{goal}) \quad (11)$$

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,  $c_1(x, \mathbf{x}_{root})$  is the cost-to-reach  $x$  from  $\mathbf{x}_{root}$  and  $c_2(x, x', \hat{x}_{goal})$  is a linear combination of the cost-to-go from  $x$  to a neighbour node  $x'$  and the expected cost-to-go from  $x'$  to the target. I implemented  $c_1(\cdot)$  as a cumulative discounted and rewarded travelled distances. More specifically, I define

$$c_2(x, x', \hat{x}_{goal}) = \alpha d_{bound}(x, x') + \beta d(x', \hat{x}_{goal}) + \gamma d_{cfg}(x) \quad (12)$$

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

$$d_{bound}(x, x') = \begin{cases} \psi(x, x') & \text{if } W(x) - W(x') \in \mathcal{B}(r_l, r_a) \\ & \text{and } x - x' \in \mathcal{B}_n(r) \\ +\infty & \text{otherwise} \end{cases} \quad (13)$$

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).

$$\bar{c}_2(x, x', \hat{x}_{goal}, A, \mathbf{p}^{1:k}) = \alpha \mathcal{J}(x, x', \mathbf{p}^{1:k}) d_{bound}(x, x') + \beta d_A(x', \hat{x}_{goal}) + \gamma d_{cfg}(x') \quad (14)$$

where  $A$  is the diagonal covariance matrix of the 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  $\mu$  and  $\mathcal{J}(x, x', \mathbf{p}^{1:k}) \in (0, 1]$  is a factor which rewards trajectories with a large difference between expected observations if the object is at the expected location,  $\mathbf{p}^1$ , versus observations that would be expected if the object is at other poses,  $\mathbf{p}^{2:k}$ , sampled from the distribution of poses associated with the object's positional uncertainty:

$$\mathcal{J}(x, x', \mathbf{p}^{1:k}) = \frac{1}{k-1} \sum_{i=2}^k e^{-\Phi(x, x', \mathbf{p}^i)} \quad (15)$$

<sup>1</sup> For the sake of simplicity, I reduce the mathematical notation by writing  $d(x, x')$  instead of  $d(W(x), W(x'))$ .

<sup>2</sup> I simplified the notation  $\mathcal{B}_{SE(3)}(\cdot)$  in  $\mathcal{B}(\cdot)$  for practical reasons.

where:

$$\Phi(x, x', \mathbf{p}^i) = \|\mathbf{h}_t(x, x', \mathbf{p}^i) - \mathbf{h}_t(x, x', \mathbf{p}^1)\|_2 \quad (16)$$

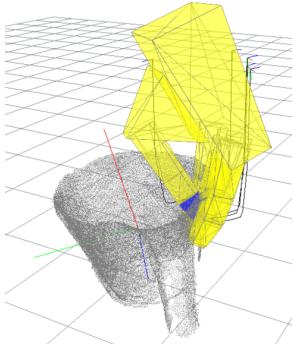
for each  $i \in [2, k]$  and  $\mathbf{h}_t(x, x', \mathbf{p}^i)$  is sequence of probability of reading a contact travelling from state  $x$  to  $x'$ . In this implementation  $\mathbf{h}_t(x, x', \mathbf{p}^i) = \mathbf{h}(x', \mathbf{p}^i)$ . In other words, I evaluate the likelihood of making a contact while moving from state  $x$  to  $x'$  as the likelihood of making a contact only in the next state  $x'$ . Note that this observational model is designed to conserve (14) as in (12) 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  $\mathcal{J}(\cdot)$  is equal to 1. However I also encode uncertainty in the second factor of the 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 break down the curse of dimensionality or, equivalently, increase the number of sampled configurations to properly cover the configuration space.

The proposed solution is to build a hierarchical planner. First a PRM is constructed only in the arm configuration space in order to find a global path between the  $\mathbf{x}_{root}, \hat{x}_{goal}$ . 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  $\mathbf{x}_{root}$  to  $\hat{x}_{goal}$ . Then the planned trajectory is refined by constructing a new PRM in the entire configuration space of the manipulator (e.g. arm + hand joint space) along the global path. In other words, this approach limits 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 enable us to plan dexterous reach-to-grasp trajectories up to 21 DoF with only 1,000 sampled configurations. Note that this is the same order of magnitude that it is used in practise for planning trajectories of much simpler 6 DoF robot manipulators.



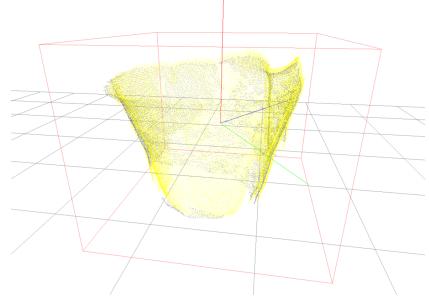
**Fig. 5** Rim grasp example on a jug. In this case the target grasp requires to displace the robotic thumb on the internal surface of the object. The grasp configuration is computed using the method described in (?).

#### 4.5 Planning a dexterous grasping trajectory for non-convex objects

The implementation of our planner uses a modified version of Probabilistic Roadmap (PRM) planning, (Kavraki and 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 5 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.

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 and Lowe 2014). An example is shown in Fig. 6.

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



**Fig. 6** 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.

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.

#### 4.6 Belief update

Once a trajectory is executed and a real (unexpected) observation  $y$  is detected, the belief state is updated according to the Bayes’ rule. The belief state is represented as a set of  $N$  particles  $\mathbf{b}_t = \{\mathbf{b}_t^z\}_{z=1}^N$ . In a particle filter fashion the weight of each particle  $\mathbf{b}_t^z$ ,  $z \in \{1, \dots, N\}$  is updated as follows

$$Pr(y|x, \mathbf{b}_t^z) = \prod_{j \in \hat{M}} Pr(y_t(j)|\mathbf{x}_t(j), \mathbf{b}_t^z) \quad (17)$$

and then re-sampling is performed to generate a posterior distribution  $\mathbf{b}_{t+1}$  as new set of particles  $\{\mathbf{b}_{t+1}^z\}_{z=1}^N$ .

In simulation this approach assumes that there are no false detections. 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, the algorithm skips the belief update step and moves the robot back to a safe configuration before triggering the re-planning.

#### 4.7 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.

#### 4.8 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.

### 5 Experiments

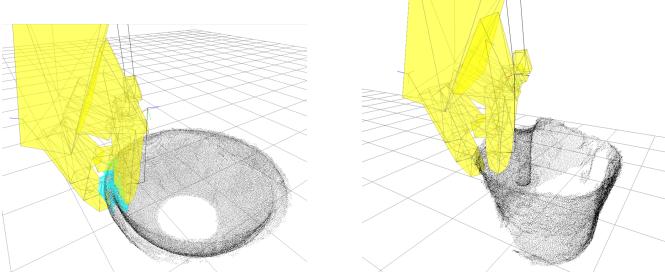
This section presents the experimental results used to evaluate the new set of algorithms presented in this paper. Three strategies for planning dexterous reach-to-grasp trajectories are evaluated in both a half-humanoid robotic platform, Boris, and a simulated environment:

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

In this evaluation the aim is 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. It also shows that planning trajectories that maximise information gain result in fewer re-planning iterations to achieve a grasp.

First, Sec. 5.1 presents empirical results collected on 20 trials for a single object (a jug) using a real robot. The ability of these three different strategies to achieve a grasp configuration is tested and we highlight the benefits of using a mean-shift algorithm with hierarchical clustering to compute the estimate of the object's pose.

In addition, Sec. 5.2 presents the results collected on 120 trials in simulation for four objects: a jug, a bottle of coke, a stapler and a Mr Muscle spray bottle. In these experiments, a mean-shift algorithm is still used to compute the pose estimate, and the aim is to evaluate the ability of the three strategies to achieve a grasp configuration.



**Fig. 7** Pinch with support grasp learned on a bowl and transferred to a jug, using the method described in (Kopicki et al. 2014). The image on the left shows the learned contact model (blue points) for all the fingers involved in the grasp. The right image shows a possible grasp adaptation on a jug.

### 5.1 Experiments on the robot Boris

These experiments are composed of 2 runs of 20 trials each (10 trials using MYCROFT and 10 using IR3NE). The results for the ELEMENTARY strategy are extrapolated from the results collected for MYCROFT, since these two approaches construct a reach-to-grasp trajectory minimising the same cost function, therefore if MYCROFT achieves (or fails to achieve) a grasp at the first iteration, the ELEMENTARY also would succeed (or fail) equally.

The jug in Fig. 6 has been used for both sets of trials. A dense point cloud representing the object model, is assumed to be available at the beginning of each run. This object model is acquired by scanning the operational workspace with a depth camera from 6 different views. These views are merged to generate a single point cloud of the object model. This pre-processing aligns the single view point clouds, registering and eliminating outlier points.

In each trial, the target object is placed in a different configuration, on a table in front of the robot. The query point cloud is acquired by scanning the operational workspace from 3 different views. Once these views are aligned and registered, a belief state with a set of particles is estimated. The first run used a mean-shift algorithm combined with the hierarchical clustering algorithm to estimate the object's mean pose while the second run used only the single mean-shift algorithm.

In both runs, a target grasp for the jug is computed by adapting a pinch with support grasp learned on a bowl. The training and test grasps are shown in Fig. 7. Then a reach-to-grasp trajectory is computed and performed. A trial is considered successful if Boris converges to the target grasp configuration and lift the object from the table surface.

Figure 8 summarises the empirical results collected. In order to test the ability of a sequential re-planning algorithm to converge to the “true” pose of the object,

a ground truth pose of the object is calculated at the beginning of each trial using the same model-fitting algorithm, sampling 500 times more features. The algorithms have no knowledge of the ground truth pose.

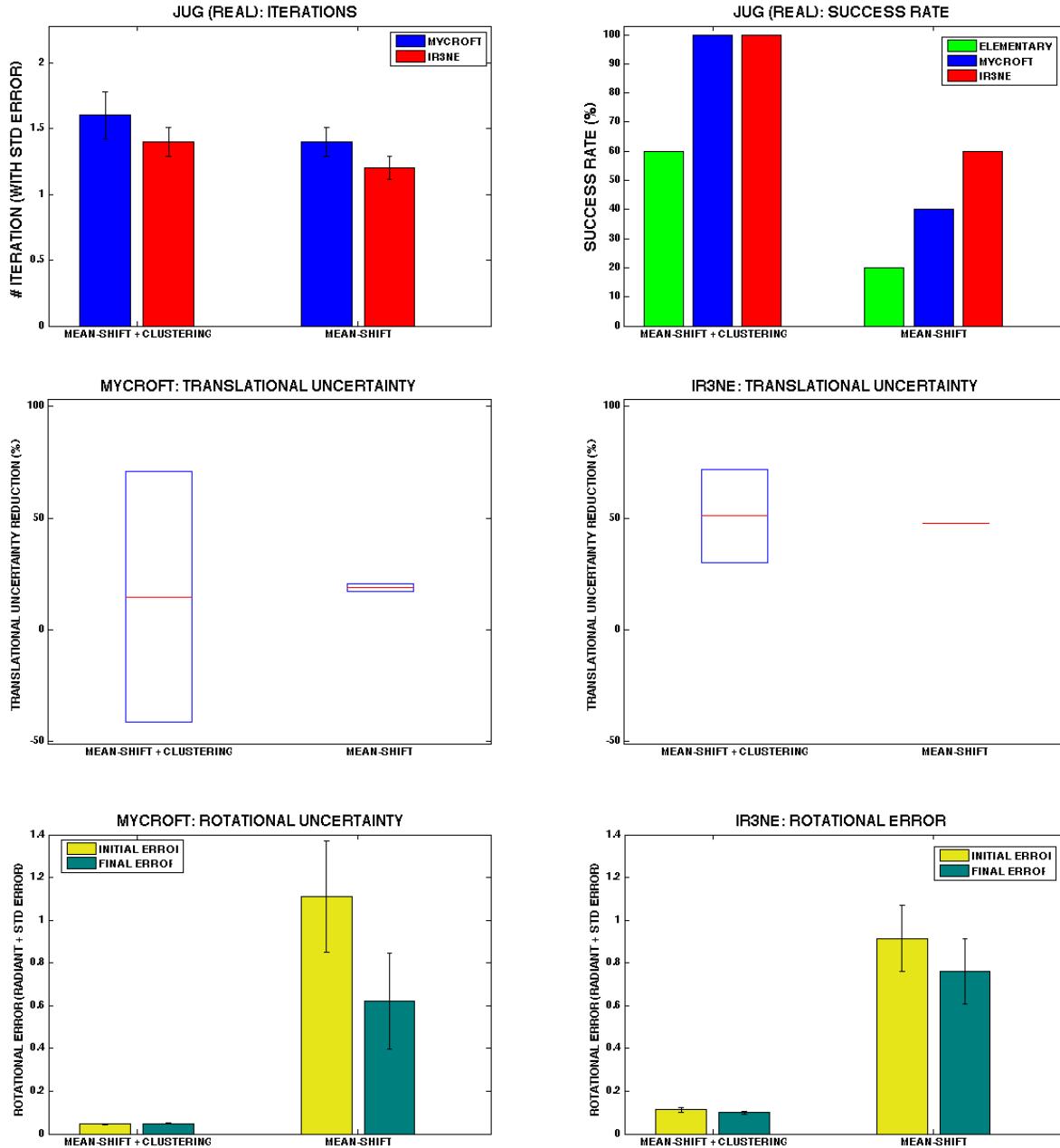
The results in Fig. 8 are organised as follows:

- The number of average planning iterations across all the trials for each run, for both the re-planning strategies: MYCROFT and IR3NE (Fig. 8 top left chart).
- The success rate across all the trials for each run, in the sense of their ability to converge to the planned grasp for all the three strategies (Fig. 8 top right chart).
- The averaged reduction in the positional error between the estimated location and the ground truth across all the trials for each run. The error is computed as the Euclidean distance in a 3D space for both strategies: MYCROFT (middle left chart) and IR3NE (Fig. 8 middle right chart).
- The averaged reduction in the rotational error between the estimated rotation and the ground truth across all the trials for each run. The error is computed as the distance in the quaternion space for both strategies: MYCROFT (bottom left chart) and IR3NE (Fig. 8 bottom right chart). The rotational error is computed for each orientation on the unit hypersphere in 4D quaternion space, and then measure displacement as the length of the arc of the geodesic which connects these two points.

### 5.2 Experiments in a virtual environment

The experiments are composed of 120 trials per object in a virtual environment, and they are run over four different objects: a jug, a coke bottle, a stapler and a Mr Muscle spray bottle. 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, in the form of a dense point cloud, acquired scanning the object with a depth camera from 7 different views. As before, these views are 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. Figure ?? shows the pre-processed point clouds of each simulated object, and the associated target grasp configurations. For these experiments, the models are composed of 7 single-view point clouds, apart from the bottle of coke which is composed of only 5 views.

The algorithms have been tested under the hypothesis that, in each trial, the object is in a different posi-



**Fig. 8** Empirical results on the Boris robot platform. The results refer to 2 sets of 10 trials on a jug. Three strategies were used: ELEMENTARY (green), MYCROFT (blue), IR3NE (red). Two conditions are presented. The first uses a mean-shift algorithm combined with a hierarchical clustering algorithm to estimate the object's pose. The second uses only a mean-shift algorithm. The top left chart presents the averaged number of iterations across the trials to reach a grasp. The top right chart shows the success rates across the trials. The middle row shows how the initial linear uncertainty (in percentile) is reduced by sequential re-planning until the algorithms converge to a grasp. The bottom row presents the reduction in the rotational uncertainty. Yellow bars represent the initial rotational error, while green bars are the final error. The rotational error is computed in quaternions where similar rotations yield to error value close to zero and rotations with 180 degrees difference yield an error value of 1.

tion - 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. Four different conditions have been tested. In each condition we randomly selected either 1, 3, 5 or 7 view point clouds from the 7 single-views collected, in order to simulate real situations in which the

robot's depth camera has been able to observe smaller or larger parts of the object. Once the subset of views has been selected and moved to the simulated object location, the state estimation procedure described in ?? is performed.

Figures ??, ??, ??, and 4 summarise the data collected in our experiments. For each condition mentioned above, a model coverage is computed, in terms of which percentage of the model surface was covered by the merged point clouds from the individual views. All the results are compared with respect to the model coverage in percentage terms. Again, a "ground truth" pose for the object is calculated at the beginning of each trial by using the same model-fitting algorithm used in Sec. ??, but sampling 500 times more features. The algorithms have no knowledge of the ground truth pose, however in the virtual environment the ground truth is used to model the real object location, and is used to trigger simulated contacts with the robot hand. These simulated contacts cause the sequential re-planning algorithm to stop and update the belief state.

The results in Figures ??, ??, ??, and 4 are organised similarly to the experiments on the real robot:

- The number of average planning iterations across all the trials for each condition (1, 3, 5, and 7 single-view query), for both the re-planning strategies: MYCROFT and IR3NE (top left chart).
- The success rate across all the trials for each condition, in the sense of their ability to converge to a grasp for all the three strategies (top right chart).
- The average reduction in the positional error between the estimated location and the ground truth across all the trials for each condition. The error is computed as the Euclidean distance in a 3D space for both strategies: MYCROFT (middle left chart) and IR3NE (middle right chart).
- The average reduction in the rotational error between the estimated rotation and the ground truth across all the trials for each condition. The error is computed as the distance in the quaternion space for both strategies: MYCROFT (bottom left chart) and IR3NE (bottom right chart). The rotational error is computed for each orientation on the unit hypersphere in 4D quaternion space, and then measure displacement as the length of the arc of the geodesic which connects these two points.

## 6 Conclusion

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.

## References

- Betts, J.: (2001). *Practical methods for optimal control using nonlinear programming*. Siam
- Detry, R.: (2010). *Learning of multi-dimensional, multimodal features for robotic grasping*. Ph.D. thesis, University of Liege
- Hillenbrand, U.: (2008). Pose clustering from stereo data. In: *VISAPP Int. Workshop on Robotic Perception*
- Hillenbrand, U., Fuchs, A.: (2011). An experimental study of four variants of pose clustering from dense range data. In: *Computer Vision and Image Understanding*
- Hsiao, K., Ciocarlie, M., Brook, P.: (2011). Bayesian grasp planning. In: *IEEE Proc. Workshop on Mobile Manipulation: Integrating Perception and Manipulation, Int. Conf. on Robotic and Automation (ICRA)*
- Jacobson, D., Mayne, D.: (1970). *Differential dynamic programming*. Elsevier
- Kavraki, L., Svestka, P.: (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In: *IEEE Trans. on Robotics and Automation*
- Kopicki, M., Detry, R., Schmidt, F., Borst, C., Stolkin, R., Wyatt, J.: (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)*
- LaValle, S.M.: (1998). Rapidly-exploring random trees: A new tool for path planning. Tech. rep., Computer Science Dept, Iowa State University
- Muja, M., Lowe, D.: (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence (PAMI)* **36**, 2227–2240
- Nikandrova, E., Laaksonen, J., Kyrki, V.: (2013). Towards informative sensor-based grasp planning. *Robotics and Autonomous Systems* pp. 340–354
- Petrovskaya, A., Khatib, O.: (2011). Global localization of objects via touch. *IEEE Trans. on Robotics* **27**(3), 569–585
- Platt, R., Kaelbling, L., Lozano-Perez, T., Tedrake, R.: (2001). Simultaneous localization and grasping as a belief space control problem. Tech. rep., CSAIL, MIT
- Platt, R., Kaelbling, L., Lozano-Perez, T., Tedrake, R.: (2012). Non-gaussian belief space planning: Correctness and complexity. In: *IEEE Proc. Int. Conf. on Robotic and Automation (ICRA)*
- Sahbani, A., El-Khoury, S., Bidauda, P.: (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems* **60**(3), 326–336

- Saxena, A., Driemeyer, J., Kearns, J., Ng, A.: (2008). Robotic grasping of novel objects using vision. *Int. Journal of Robotics Research* **27**(2), 157–173
- Tuzel, O., Subbarao, R., Meer, P.: (2005). Simultaneous multiple 3d motion estimation via mode finding on lie groups. In: *Int. Conf. Computer Vision*
- Zito, C., Kopicki, M., Stolkin, R., Borst, C., Schmidt, F., Roa, M.A., Wyatt, J.: (2013). Sequential trajectory re-planning with tactile information gain for dexterous grasping under object-pose uncertainty. In: *IEEE Proc. Intelligent Robots and Systems (IROS)*
- Zito, C., Stolkin, R., Kopicki, M.S., Luca, M.D., Wyatt, J.L.: (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)*