

Action Selection for Touch-based Localisation Trading Off Information Gain and Execution Time

Niccoló Tosi¹, Olivier David¹ and Herman Bruyninckx²

Abstract—In the context of touch-based object localisation, solving the problem of "where to sense next" is a challenging task due to the curse of dimensionality related to belief-state reasoning. We present a constrained optimisation scheme that computes the next best action maximising the trade off between (i) the localisation information gain, (ii) the time required for its computation, and (iii) the motion-execution time. This allows the robot programmer to have a deterministic influence on the length of every sensing action. The proposed methodology is applied to localise a solid object in 3D using a Staubli RX90 robot equipped with a force-torque sensor coupled with a spherical end effector. A case-study comparison of the task executed with two different time constraints is presented.

I. INTRODUCTION

Performing scene-model calibration with efficiency is a major concern for tele-operation applications such as those depicted in [1]. In particular, in case of harsh environments it is necessary to cope with highly-soiled scenes, e.g. by dust, mud and dirty water, which may impede camera-based and tactile-based calibration. In this context, force sensing widens the set of scenarios in which object localisation is made possible, both as a stand-alone tool and as part of a multi-sensor system. Specifically, force sensing is preferred over tactile sensing as its performance is less dependent on the finesse of the touched surface and the scale of the interaction forces are in the range of 1000N. For the same reason, compliant motion may not always be feasible for such applications.

In literature, several works have studied the problem of processing force information to infer object pose [2][3]. However, the problem of "where to sense next" is still a major challenge for nowadays robotics research. In particular, state-of-the-art techniques tackle this problem by selecting the action that maximises the expected information gain among a set of feasible candidates [4][5]. Since making decision in belief state is computationally expensive, following such a pure info-driven approach may cause the robot to stay still for a long time before performing any useful info-gathering action, while significant information could be obtained following less-optimal but cheaper solutions.

Driven by the need to speed up the whole localisation process, we present a new decision-making scheme in which

the time to reason and execute the next action becomes a design parameter to be set explicitly. This allows to tune the robot attitude from fully speculative to purely action-driven.

This paper is organised as follows: Section II summarises the state of the art on touch-based localisation and active force sensing; Section III presents the action selection methodology; Section IV describes the robot implementation.

II. LITERATURE REVIEW

A. Touch-based Localisation

In the early 80s, 3D and 6D polyhedral-object localisation was performed using interpretation trees [6], [7]. Then, probabilistic schemes were developed to accomplish the localisation accounting for geometric and measurement uncertainties. Though powerful, probabilistic approaches tend to become computationally expensive as the problem complexity increases with the initial uncertainty and the number of DOFs. In order to limit the size of the problem, previous works analysed touch-based localisation reducing either or both the number of DOFs and the search space dimensions.

In 2001, Gadeyne et al. [8] performed cube localisation in 3DOFs by sampling the solution space and processing force-sensor information. In 2005, Gadeyne et al. [2] adopted sequential Monte Carlo sampling to estimate contact-state formations during cube-in-corner operations using force sensing. In 2006, Petrovskaya et al. [3] performed 6DOF force-based localisation with uncertainty of 400mm in position and 360 deg in orientation, proposing an enhanced Particle Filter scheme named Scaling Series method in a later paper [9].

B. Active Sensing

In the framework of robotic localisation tasks, the definition of an optimised sequence of actions aimed to reduce the uncertainty over the pose of the environment elements goes under the name of active sensing.

Previous related works analysed the problem of actively sensing the environment for 3D self-localisation tasks. In [10], Markov localisation was applied to infer robot position inside a structured office environment using laser and ultrasound sensors. A cost-vs-utility function was introduced to select the best movements and sensing actions to perform. Costs were represented by time and energy spent, and utility was represented by the expected decrease of uncertainty evaluated through the Kullback Leibler Divergence (see [11]), i.e. the non-symmetric measure of the difference between the current posterior probability distribution and

*This work was sponsored by CEA, LIST, Interactive Robotics Laboratory, France

¹Niccoló Tosi and Olivier David are with CEA, LIST, Interactive Robotics Laboratory Rue Noetzelin F-91190 Gif Sur Yvette, France {niccolo.tosi}{odavid}@cea.fr

²Herman Bruyninckx is professor at the Department of Mechanical Engineering, University of Leuven, Belgium herman.bruyninckx@mech.kuleuven.be

the envisioned Probability Density Function (PDF) after the sensing action. Further to that work and focusing on reducing the sensor resources required, the same task was performed using angular and linear odometers, a compass and a contact sensor, an angular odometer and a contact sensor [12]. More recently, the same operation was accomplished only using a clock and a contact sensor [13].

Active force sensing to perform cube-in-corner tasks with autonomous motions was studied in [14]. The full operation was decoupled into a compliant sequence of contact formations (CFs), stored in the so-called CF Graph. Then, the optimisation process was decoupled into a requirement for the CF sequence and a requirement for the active sensing motions in each CF. Results from this work proved that such an exhaustive approach is computationally expensive and suggested further research to adjust the level of resolution of the estimator and controller to the accuracy required by the single action.

In 2009, Hsiao [4] performed object localisation and grasping in 3DOFs using tactile sensors installed on a three-finger hand. The authors focused on the decision-making side of the problem by implementing an approximated Partially Observable Markov Decision Process (POMDP) [15] scheme to maximise the information gain over a finite-horizon series of motions. The best action is chosen by using the Kullback Leibler Divergence (KLD) as a metric to decide among a pre-determined set of trajectories, defined relatively to the best pose estimate of the object. The measurement model accounts for both contact and no-contact information.

In 2012, further to Hsiao's work, Hebert et al. [5] performed 3DOF localisation using a hand with force sensors. With respect to Hsiao, motion primitives are generated and evaluated as the localisation takes place. Furthermore, Javdani et al. [16] compared the information-based approach followed by Hebert and Burdick with a hypothesis-pruning technique through a series of simulations.

III. ACTION SELECTION DECISION MAKING

A. Motivation

The state of the art on active force sensing is facing two major challenges: coping with high initial uncertainties and selecting sensing actions with efficiency. In a previous paper [17], we studied the former problem setting up an experiment to observe human beings approaching a blind localisation task. Then, we introduced a decoupling scheme which allows to simplify a high-dimensional problem into a sequence of subtasks. In this paper, we focus on the latter problem.

To the best of our knowledge, previous work on action selection for touch-based localisation focused on maximising the expected information gain choosing from a set of candidate actions. The computational and motion costs were not accounted for in the decision making process. In practice, this resulted in the robot devoting most of the time to reasoning instead of acting. With respect to the state of the art, this paper introduces a new decision-making scheme which allows to find a feasible solution to the problem of "where to sense next" constraining the global time spent for

the next action, including reasoning and motion execution. The allocated time is a design parameter allowing to tune the robot behaviour from purely info-gathering to action-oriented. In order to facilitate its adoption in real-life industrial contexts, this new decision-making framework has been applied to localise solid objects using a force-torque sensor and point-to-point motion primitives.

B. Inference Scheme

Adopting a typical Bayes' filter, the problem of localising an object in a N-dimensional space requires the prediction of the object state \mathbf{x}_t after action a_t using its dynamic model, and to update the belief adopting the new measurement \mathbf{z}_t obtained through a_t . In our application, the object to localise is rigidly fixed to the environment, so its pose does not change over time. Therefore, only the update step is needed:

$$P(\mathbf{x}_{t+1}|a_t) = \frac{P(\mathbf{z}_t|\mathbf{x}_t)P(\mathbf{x}_t)}{P(\mathbf{z}_t)} \quad (1)$$

The object is modeled as a polygonal mesh composed by a set of faces $\{f_i\}$ and their corresponding normal vectors $\{n_i\}$. Each measurement is comprised of contact position z_p and normal vector z_n . Such measurements are supposed to be independent.

$$P(\mathbf{z}|\mathbf{x}_t) = P(z_p|\mathbf{x})P(z_n|\mathbf{x}). \quad (2)$$

As in previous works [3][5], the likelihood functions expressing the probability of each face to cause the contact and normal measurement are:

$$P(z_p|f_i) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left[-\frac{1}{2} \frac{\text{dist}(z_p, f_i)^2}{\sigma_p^2}\right] \quad (3)$$

and

$$P(z_n|f_i) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2} \frac{\|z_n - n_i\|^2}{\sigma_n^2}\right]. \quad (4)$$

A maximum-likelihood approach is followed to solve the correspondence problem, i.e. finding which face to consider when calculating the likelihood of a given pose.

$$P(\mathbf{z}|\mathbf{x}_t) = \max_{f_i} (P(z_p|f_i)P(z_n|f_i)) \quad (5)$$

Touch-based localisation problems typically require inference models able to cope with multimodal joint distributions $P(\mathbf{x}_t)$, for instance particle filters or histogram filters. In this work, we adopt the former.

C. Info vs Cost Decision Making

Previous related works [5][4] tackled the problem of choosing the next best action by first generating a complete set of candidate actions, then selecting the one that maximises the expected information gain. In this paper, such scheme is enriched by constraining the overall computation and motion execution.

The information gain associated to a candidate action a_i is measured as the KLD between the prior and the posterior distribution after executing a_i .

$$IG(a_i) = \int_x P(\mathbf{x}_{t+1}|a_i) \log \frac{P(\mathbf{x}_{t+1}|a_i)}{P(\mathbf{x}_t)} dx \quad (6)$$

However, the measurement associated to a_i is only hypothetical, so it is necessary to consider the expected information gain instead.

$$E[IG(\tilde{z}, a_i)] = \int_z P(\tilde{z}) \int_x P(\mathbf{x}_{t+1}|a_i) \log \frac{P(\mathbf{x}_{t+1}|a_i)}{P(\mathbf{x}_t)} dx dz \quad (7)$$

Moreover, since measurement \tilde{z} may occur at any point along the trajectory, the probability $P(\tau)$ of ending the action at time τ has to be taken into account [5], so an additional expectation over τ is to be added to Eq. 7. However, in this work, the decision is made under the assumption that each action a_i terminates on the nominal contact point along the trajectory. Therefore, the next best action is chosen so that it maximises the expected information gain.

$$a^* = \arg \max_{a_i} E[IG(\tilde{z}, a_i)] \quad (8)$$

Several strategies may be adopted to call the localisation done, e.g. setting a threshold on the eigenvalues of the covariance matrix Σ , on the probability entropy or on the probability mass of a target point of the object within a given tolerance.

The action-selection algorithm is presented in pseudo-code in Alg. 1. Each time the robot needs to make a decision, the best solution a^* is initialised to "stay still", unless an open-loop action is available, e.g. a user-defined motion (lines 2-9). Then, the timer starts (line 10). While the execution time of the current best solution is smaller than the available time, candidate actions are generated and evaluated, updating the best solution if the expected information is greater than the current one (lines 13-21). As the available time intercepts the estimated execution time of the current best solution, the reasoning is over and a^* is performed. The condition at line 15 ensures that the next candidate to evaluate respects the time constraint. Hence, it is necessary to estimate the execution time t_{exec} and the evaluation time t_{eval} of a given action a_i .

A set of sensing actions is defined off-line relatively to the object geometry. Their specification depends on the available robot control system and the adopted object representation, e.g. a polygonal mesh. For instance, they may aim at touching the faces of the mesh moving parallel to the normal vector. At run time, candidate actions are generated (line 14) starting from the configuration with the highest posterior then moving to lower-probability configurations proceeding in descending order. To simplify the calculation, the KLD is evaluated considering only the contact measurement \tilde{z} as a

Algorithm 1: Action selection with time allocation

```

1 set  $t_{max}$ ;
2 if  $a_{OL}$  is available then
3    $a^* = a_{OL}$ ;
4    $IG(a^*) = E[KLD(P_{OL}||P_0)]$ ;
5 end
6 else
7    $a^* = \text{stay still}$ ;
8    $IG(a^*) = 0$ ;
9 end
10  $t.start()$ ;
11  $i = 0$ ;
12 while  $t_{exec}(a^*) < t_{max} - t.elapsed()$  do
13    $i := i + 1$ ;
14    $a_i = \text{generateCandidateAction}(i)$ ;
15   if  $t_{exec}(a_i) + t_{eval}(a_i) < t_{max} - t.elapsed()$  then
16      $IG(a_i) = E[KLD(P_i||P_0)]$ ;
17     if  $IG(a_i) > IG(a^*)$  then
18        $a^* = a_i$ ;
19     end
20   end
21 end
22 execute  $a^*$ ;
23 if  $a^* == \text{stay still}$  then
24   set  $t_{max} = 2t_{max}$ ;
25   go back to line 2;
26 end

```

point measurement occurring on the surface of the particle from which it was generated.

Figure 1 graphically illustrates the decision-making at different time steps: the starting time t_0 , a generic instant t_i and t_{final} when a^* is executed. In this case, an open-loop action is available, so a^* is initialised to a_{OL} . As time passes, a^* is updated, until $t.elapsed()$ intersects $t_{exec}(a^*)$, and the current best action is executed.

IV. ROBOT IMPLEMENTATION

A. Set Up

The decision making scheme presented in Section III has been applied to localise a solid rectangle positioned on top of a table using a Staubli RX90 robot equipped with a force-torque sensor coupled with a spherical end effector. This set up allows to work out the contact point and normal vector in case of force interaction between the end effector and the environment [18]. Sensor uncertainty is modeled as $\sigma_p = 0.005\text{m}$ and $\sigma_n = 6\text{deg}$.

B. Problem Description

The initial uncertainty on the pose of the object is $0.8\text{m} \times 0.5\text{m}$ in translation ($\{x, y\}$) and 180 deg in rotation (γ). Further to the observation of human beings facing a similar blind-localisation problem, the 3DOFs localisation task is divided into two subtasks, namely $S1$ and $S2$. Details about the adopted DOF-Decoupled Active Force Sensing

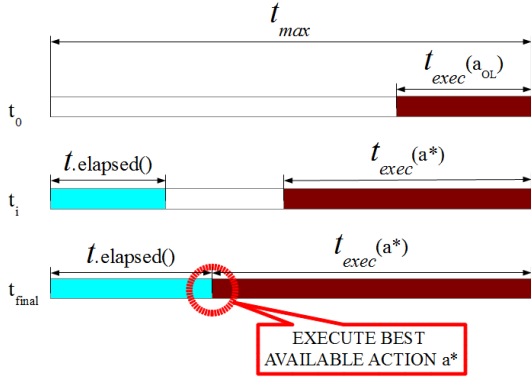


Fig. 1. Decision-making time line

scheme are provided in a previous paper [17]. $S1$ consists of a coarse 2DOFs exploration of the table in which the object is represented by its bounding circle to speed up the calculation. In this first subtask, the decision making is simplified by providing an open-loop action a_{OL} consisting of spanning the table with a zig-zag type of motion. A 2D particle filter scheme is adopted for the inference, using both contact and no-contact information. The first contact triggers the transition to $S2$, in which the joint posterior $P(x, y, \gamma)$ has to be estimated. At this level of resolution, the object is represented as a rectangle, allowing to solve Eq. 5 analytically. Also for $S2$, a particle filter is applied. The task is considered accomplished if the end effector can be placed into one of the two up-facing bottle tops with diameter of 30mm, fixed at symmetric known locations on top of the solid rectangle, as shown in Figure 2. With the diameter of the end effector equal to 25mm, this requires the robot to localise the object with a maximum linear error smaller than 5mm, which is comparable to state-of-the-art examples.

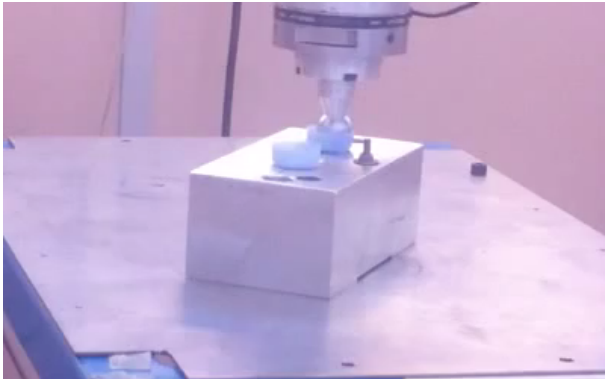


Fig. 2. Peg-in-hole to validate the localisation

C. Decision Making

A set of 12 attraction points and their related candidate actions were defined off-line relatively to the rectangle geometry, as depicted in Figure 3. Basically, each action aims at a

target point moving parallel to the surface normal. After each contact, particles' weights were calculated as in Eq. 3 and Eq. 4 and the posterior distribution is updated with Eq. 1. While the available time is greater than $t_{exec}(a^*)$, candidate actions are generated and evaluated starting from the best-estimate particle then moving to lower-posterior configurations in a sequential order.

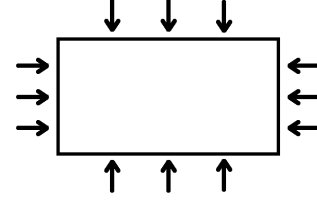


Fig. 3. Set of pre-defined attraction points and actions

In order to estimate the execution time required by a generic action $t_{exec}(a)$, the motion to position the end effector in front of the target surface and the one required to reach the target point are to be taken into account. In our implementation, the robot is position-controlled in cartesian space using a trajectory generator to feed the real-time low-level position requirement for the PID controller. Let us call D the distance between the initial and final setpoint, v_{max} the maximum allowed speed and a_{max} the maximum acceleration. A symmetric trapezoidal velocity profile is adopted, so it is possible to calculate the execution time to cover D in an analytical way. Specifically,

$$t_{exec}(a) = t_{acc} + t_c + t_{dec} \quad (9)$$

Where t_{acc} and t_{dec} are the acceleration and deceleration time, respectively. Moreover as the velocity profile is symmetric we have:

$$t_{acc} = t_{dec}. \quad (10)$$

In case

$$D > \frac{v_{max}^2}{a_{max}} \quad (11)$$

the velocity profile is trapezoidal and the execution time expressed in Eq. 9 is calculated as:

$$t_{exec}(a) = t_c + 2t_{acc} = \frac{D}{v_{max}} - \frac{v_{max}}{a_{max}} + 2\frac{v_{max}}{a_{max}}. \quad (12)$$

In case

$$D \leq \frac{v_{max}^2}{a_{max}} \quad (13)$$

the velocity profile is triangular and symmetric. Assuming a constant acceleration a_{max} , the maximum reached speed v^* is calculated as:

$$v^* = t_{acc} a_{max}. \quad (14)$$

Since

$$v^* t_{acc} = D \quad (15)$$

the motion time in Eq. 9 is calculated as:

$$t_{exec}(a) = 2t_{acc} = 2\sqrt{\frac{D}{a_{max}}}. \quad (16)$$

Figure 4 represents the velocity profiles used to control the robot motion.

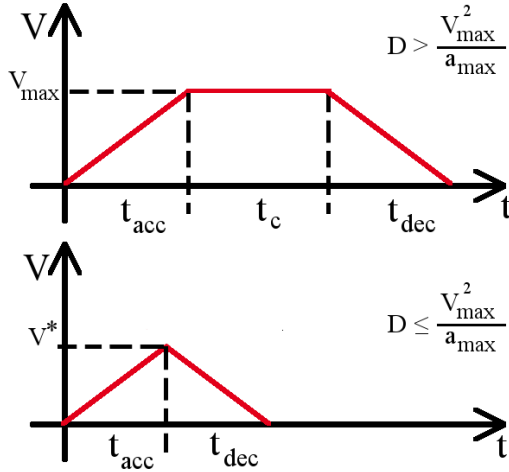


Fig. 4. trapezoidal and triangular velocity profiles

The computational time required to process each action is measured empirically and depends on the number of particles used by the inference scheme.

D. Case Study

Here, we present an example of the localisation task performed setting t_{max} to 10s and 20s. To underline the difference in the robot attitude, we require at least four contacts on at least two different sides of the rectangle. In this application, such condition is sufficient to perform the peg-in-hole operation. Figure 5 and 6 illustrate the different contact obtained with the two allocated times. The sensing actions are summarised in Figure 7 for the sake of clarity.

As a result of the constrained optimisation, in the 10s case, three local actions are selected after the first touch, as longer sensing motions would make the robot violate the 10s constraint. Instead, in the 20s case, the robot explores opposite sides three times (actions no. 2, 3 and 4). Table I presents the time elapsed on the four contact points (C1, C2, C3 and C4) and the global task execution time in both cases. Setting $t_{max} = 10s$, the global localisation task and the following peg-in-hole is performed about 12s faster than setting $t_{max} = 20s$. Therefore, for this specific application, a motion-oriented attitude of the robot is more efficient than a more info-driven approach. A comparative video of the two demos is available on line [19].

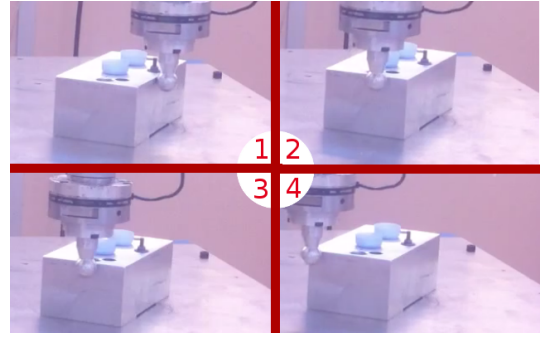


Fig. 5. Chosen actions with $t_{max} = 10s$

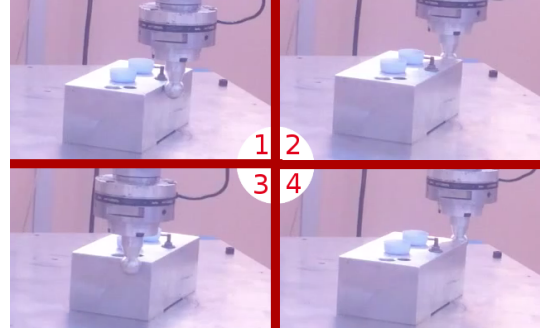


Fig. 6. Chosen actions with $t_{max} = 20s$

V. CONCLUSIONS

In this paper, the problem of "where to sense next" in touch-based localisation has been formulated as a constrained optimisation in which the information gain is explicitly traded off with the time cost of computation and execution. This enables to deterministically specify the length of the following sensing action, which becomes an explicit parameter to be set by the task programmer. This new formulation allows to tune the robot attitude from pure info-driven to more action-oriented.

A set of sensing actions is defined off-line relatively to the geometry of the object. At run time, candidate motions are generated starting from the most probable configuration then moving to lower-probability poses in a sequential order.

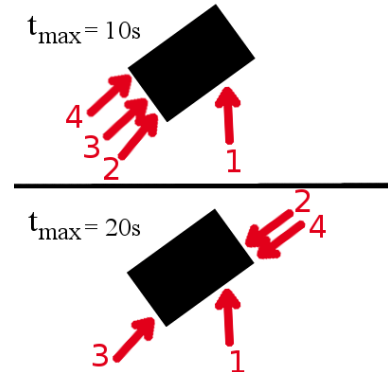


Fig. 7. Action selection with $t_{max} = 10s$ and $t_{max} = 20s$

TABLE I
CONTACT POINTS AND TASK ELAPSED TIMES

	$t_{max} = 10s$	$t_{max} = 20s$
C1	36.9s	36.2s
C2	53.9s	57.2s
C3	68.4s	77.0s
C4	83.5s	97.4s
Peg-in-hole	98.1s	110.0s

This allows to focus the search by prioritising actions with high expectation of information gain.

The decision-making scheme has been applied to a 3DOFs localisation task consisting of finding a solid rectangle on top of a table. A case study example has been presented to illustrate the info-driven and the motion-oriented attitudes of the robot in performing the localisation task with different allocated time for the next action.

FUTURE WORK

The presented decision making framework is to be extended to applications with more complex objects and higher dimensional space, studying the benefit of the trade-off between information gain and cost of computation and motion.

REFERENCES

- [1] P. Garrec, F. Geffard, O. David, F. Russotto, Y. Measson, and Y. Perrot, "Telerobotics research and development at cea, list," in *ANS EPRRSD - 13th Robotics & remote Systems for Hazardous Environments . 11th Emergency Preparedness & Response Knoxville, TN*, 2011.
- [2] K. Gadeyne, T. Lefebvre, K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model-state estimation applied to simultaneous contact formation detection and geometrical parameter estimation," *Int. J. Robotics Research*, vol. 24, pp. 615–630, 2005.
- [3] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *In Proc. of ICRA*, 2006.
- [4] K. Hsiao, M. I. of Technology, Dept. of Electrical Engineering, and C. Science, *Relatively Robust Grasping*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2009. [Online]. Available: <http://books.google.fr/books?id=tiT0SAAACAAJ>
- [5] P. Hebert, J. Burdick, T. Howard, N. Hudson, and J. Ma, "Action inference: The next best touch," *Robotics: Science and Systems Workshop on Mobile Manipulation*, 2012.
- [6] P. C. Gaston and T. Lozano-Perez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 257–265, 1983.
- [7] T. Lozano-Prez and W. E. L. Grimson, "Model-based recognition and localization from sparse range or tactile data," 1983.
- [8] K. Gadeyne and H. Bruyninckx, "Markov techniques for object localization with force-controlled robots," 2001.
- [9] A. Petrovskaya, "Towards dependable perception: Guaranteed inference for global localization," 2010.
- [10] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.
- [11] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [12] J. M. O'Kane and S. M. LaValle, "Localization with limited sensing," *IEEE Transactions on Robotics*, vol. 23, pp. 704–716, August 2007.
- [13] L. Erickson, J. Knuth, J. M. O'Kane, and S. M. LaValle, "Probabilistic localization with a blind robot," in *Proc. IEEE International Conference on Robotics and Automation*, 2008.
- [14] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Task planning with active sensing for autonomous compliant motion," *The International Journal of Robotics Research*, vol. 24, no. 1, p. 61, 2005.
- [15] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman, "Acting optimally in partially observable stochastic domains," pp. 1023–1028, 1994.
- [16] S. Javdani, M. Klingensmith, D. Bagnell, N. Pollard, and S. Srinivasa, "Efficient Touch Based Localization through Submodularity," *ArXiv e-prints*, Aug. 2012.
- [17] N. Tosi, O. David, and H. Bruyninckx, "Dof-decoupled active force sensing (d-dafs): A human-inspired approach to touch-based localisation tasks," in *IEEE International Conference on Advanced Robotics*, 2013 (forthcoming).
- [18] K. Kitagaki, T. Suehiro, and T. Ogasawara, "Monitoring of a pseudo contact point for fine manipulation," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 2. IEEE, 1996, pp. 757–762.
- [19] (2013). [Online]. Available: <http://people.mech.kuleuven.be/%7Er0267168//Videos/ICRA2014/>