Reactive Grasping for Human-Robot Interaction

Daniel Sidobre, Jean-Philippe Saut, Xavier Broquère and Jim Mainprice

LAAS-CNRS, 7 Av. Colonel Roche, 31077 Toulouse
Université de Toulouse ; UPS , INSA , INP, ISAE ; LAAS ; F-31077 Toulouse, France
firstname.lastname@laas.fr

***Abstract*— This paper presents solutions to some of the problems related to the realization of a robotic system that is able to interactively grasp an object given by a human. By interactively, we mean that the robot should be able to adapt both the target grasp and the approach movement of its arm. If the human decides to change the way he/she presents the object, the robot should possibly choose a grasp different from the one that was previously selected and use a different motion to reach the object. Such reactive capacities could be of great interest in the field of assistive robotics and, more generally, human-robot interaction.**

The paper focused on some points: Grasp planning to reactively select how to grasp the object from the human hand, trajectory planning to compute a comfortable approach motion and interpretation of the interaction forces during the object exchange. For the latter point, a smart device was specially conceived and realized to measure the interaction forces.

Some preliminary results are presented as well as the future developments we plan.

I. Introduction

Mobile manipulators are now common in research laboratories, but autonomous manipulation in the presence of human is still a challenging task. In particular, situations where human and robot must interact in a cooperative way raise many problems.

In this paper we consider such an interaction task where the robot has to take an object that a human is handing over. This is a challenging task as the robot can not know *a priori* how the human will present the object. The robot must be able to select a proper way to take the object and quickly decide to keep the same grasp or to select another one if the human moves or changes the way he/she presents the object. Once the robot has decided how it will take the object, it has to plan a safe motion and executes this motion in a manner that is comfortable to the human.

The paper is presented as follows. Section II gives a brief overview of the existing works related to autonomous manipulation. Section III presents our method to precompute a dense list of grasps that can then be used on-line to quickly select a grasp adapted to the way the human holds the object to give. Section IV concerns the planning issues involved with the approach motion of the robot's arm. The robot must first plans a collision-free path to reach the object. The computed path does not explicitly consider time and must so be transformed into a trajectory, by the mean of a soft-motion planner that provides a smooth and natural motion. The robot may also have to change the target grasp in the case the human moves too much or changes the way he/she presents the object. Once the robot has achieved its approach motion, it has to decide to grip the object to effectively take it from the human. Such a decision should rely on a fine knowledge of the forces occurring during the object exchange. To measure and analyze such forces, we conceived and built a smart device presented in section V. Section VI proposes some preliminary results for the different techniques mentioned above. Section VII exposes some of the challenging issues that remain to completely achieve the task considered in the paper.

II. Related Work

Autonomous manipulation has always been a major domain in robotic research but it appears to receive more and more interest as crucial devices like real-time sensors [7] or safer compliant arms become commonly available [5].

Most of the specific issues involved in autonomous manipulation have been addressed. For instance, door opening is a key capability for a robot to evolve in a human environment. However, the problem of opening any kind of door (by pushing or by pulling), when a combined motion of arm and platform is necessary, was only solved recently [2]. Indeed, the robot proposed by Chitta *et al.* plans the combined motion of its arm and platform so that it can fully open the door. This requires to integrate the door state (opening) in the system configuration. The authors advantageously describe the state of the door with a binary instead of its opening angle to reduce the system dimension. This binary simply tells if the door is between its fully-closed configuration and the robot platform or between the robot platform and its fully-opened configuration. The authors can then used a deterministic method (lattice-based discretization plus a A*-like search) to solve the problem.

A more common topic concerns the pick-up tasks. Jain and Kemp [6] have presented the assistive robot EL-E that can pick up objects placed on flat horizontal surfaces like tables or floor. The objects are supposed to be novel *i.e.* the robot has no model of their shapes. A laser range finder is used to build a cloud point of the environment. Various segmentation processes are then performed to extract flat surfaces and retrieve point sets corresponding to objects. The patient roughly indicates the desired object with a laser pointer. The robot uses a simple heuristic to grasp the object. The authors present a complete evaluation of their system performance, that reveals its efficacy in real situation. Saxena *et al.* [8] were also interested in grasping of novel objects. Their approach is

image-based and consider grasping on a small region (like two-fingered pinch grasp). The idea is to predict, on an image of the object to grasp, the 2d location of the grasp via visual features detection. From a set of images of the object, the 2d locations can then be triangulated to obtain a 3d grasping point. The prediction model is the result of a supervised learning. From given grasping parts on the database objects, an automatic supervisor computes their 2d projections on different synthetic views.

Closer to the topic of this paper, Edsinger and Kemp [4] studied the task of a human handing an object to a robot. Their experimental results show how human subjects, with no particular instructions, instinctively control the objects position and orientation to match the configuration of the robots hand while it is approaching the object. The humans spontaneously tries to simplify the task of the robot. Their robot is a humanoid robot called Domo, equipped with two compliant arms and hands. Domo has several motor and perceptual capacities that are very useful for an object exchange task. For instance, it reduces its arm stiffness and measures its hand velocity. If the velocity is greater than the expected velocity, provided by a learned model, it means that the hand is moved by the human placing the object inside it. To accurately track the box-shaped object orientation, it is equipped with a an inertial measurement unit. The robot uses its cameras to locate the box and tries to grasp along its larger axis. A similar object handing task, although simpler, was performed on the Robonaut [3] to grasp the tool handed by a human. A stereo vision system is used to locate the human and the tool. The robot is equipped with a tactile glove and a *grab reflex* command makes the finger automatically close when the object is detected.

III. Grasp Planning

Grasp planning of a complex object has been so far too computationally expensive to consider it can be performed in real-time. Therefore, in a real application, it is preferable to use precomputations as much as possible. In the proposed framework, a grasp list is computed off-line for the considered object so as to capture the best possible the variety of the possible grasps. This list will then be used to select, during interactive grasping, the grasps that are currently reachable and from them the best one according to a scoring function described further.

A. Grasp List Computation

Precomputing a reduced set of good grasps is of no interest because it is mainly the choice of the human that will constrain the way the object can be grasped by the robot. The grasp list has thus to be as general as possible. Consequently, we choose to uniformly sample the possible hand approaches. Each hand approach is characterized by a frame referred as *grasp frame*. A grasp frame is the transform matrix of the relative object/palm pose. The grasp frames are centered on the intersection of the finger workspaces so that they are roughly centered where the contacts may occur. Our algorithm applies the following steps that will be detailed further:

- Build a set of grasp frame samples.
- Compute a list of grasps from the set of grasp frames.
- Perform a stability filter step.
- Compute a quality score for each grasp.

1) Grasp frame sampling: The possible grasp frames are sampled by the mean of a grid. We set as an input the number of positions and the number of orientations, each couple position-orientation defining a frame. The positions are uniformly sampled in the object axis-aligned bounding box with a step computed to fit the desired number of position samples. The orientations are computed with an incremental grid like the one in [10]. For each grasp frame, a set of grasps will be computed.

2) Grasp computation from grasp frame: To compute a grasp from a grasp frame, the most expensive computation –except for collision test– is the finger inverse kinematics. Therefore we make use of two data structures that let us find quickly the intersection between the finger workspaces and the object surface.

3) Object surface model: We propose to approximate the object surface with a contact point set, keeping trace of where it is on the object mesh to be able to get some local information (surface normal and curvature) later. The set is obtained by a uniform sampling of the object surface. The sampling step magnitude is chosen from the fingertip radius. A space-partitioning tree is built upon the point set in order to have a hierarchical space partition of the points (Figure 1). It is similar to a kd-tree. Starting from the original set of points, we compute the minimal axis-aligned box containing all the points. Such a box is usually referred as Axis-Aligned Bounding Box or AABB. This first AABB is the tree root. The root AABB is then splitted in two along its larger dimension. This leads to two new nodes, children of the root, containing each a subset of the original point set. The splitting process is then recursively applied to each new node of the tree. The process ends when a node AABB contains only one point.

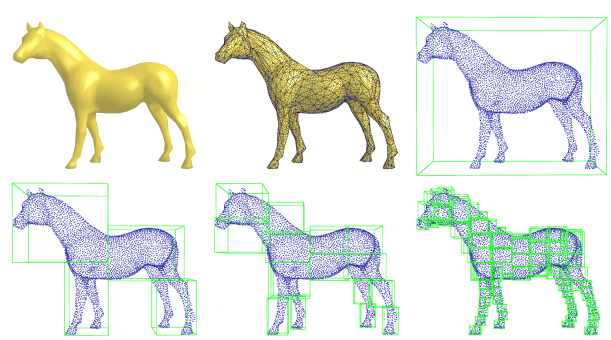

Fig. 1. The object mesh is uniformly sampled with a point set (top images). The point set is then partitioned using a kind of kd-tree (bottom images).

We then need to find the intersection of each finger workspace with the object surface tree. So we introduce

another data structure to approximate the finger workspace and compute this intersection quickly.

4) Finger workspace model: As spheres are invariant in rotation, they are interesting to build an approximation of the finger workspace. Starting from a grid sampling of the finger workspace, we incrementally build a set of spheres fitting strictly inside the workspace. First, points of the grid are marked as being boundary points (on the workspace envelope) or inner points (strictly inside the workspace volume). For each inner point, the smallest distance to the boundary points is computed, referred as $dmin$. The inner point having the biggest $dmin$ is the center of the first sphere S_1, of radius $dmin$. For all the inner points that are not inside S_1, a new $dmin$ is computed, that is the minimum of the old $dmin$ and the minimal distance to S_1. The point that has the biggest $dmin$ is the center of the second sphere S_2, of radius $dmin$. This process is repeated until we have reached the maximal desired sphere number or the last computed sphere has a radius less than a specified threshold. We keep the ordering of the construction so that the sphere hierarchy starts from the biggest ones, corresponding to workspace parts that are the farthest to the finger joint bounds. These bounds were first slightly reduced in order to eliminate configurations where the fingers are almost completely stretched.

Once we have both the contacts tree and the workspace sphere hierarchy, it is very fast and easy to determine the intersection of the two sets and so the contact points.

5) Intersection between object surface and finger workspace: All the operations that have to be performed are sphere-box intersection tests. The intersection is tested from the biggest to the smallest sphere, guarantying that the *best* parts of the workspace will be tested first, *i.e.* the one farthest to singularities due to the joint bounds. Starting from the tree root, we test if there is a non null intersection between a AABB-node and the sphere. If not, we stop exploring this branch, otherwise we test the sphere against the two node children, until we arrive to a leaf node *i.e.* a single point. We then just have to test if the point is included in the sphere volume. Figure 2 shows the different steps to compute a candidate contact points from a given grasp frame, for the Schunk Anthropomorphic Hand (a four-fingered hand with three DOFs for each finger except for the thumb having four DOFs). At this stage, we just know that the points will pass the finger inverse kinematics test. No collision tests have been performed yet. For a given grasp frame, the grasp is computed finger by finger, that means that, if we have the contact and configurations of the fingers 1 to $i-1$, we search a contact point for finger i and test collision only with the fingers 1 to i as the other finger configurations are not yet known. We start from the thumb as no stable grasp can be obtained without it. If a finger can not establish a contact, it is left in a *rest* (stretched) configuration. If we have three contacts or more, we can proceed to the stability test. Note that, at this stage, we have a collision-free grasp *i.e.* no collision between the hand and the object and do not yet consider collision with the environment or the robot arms or body.

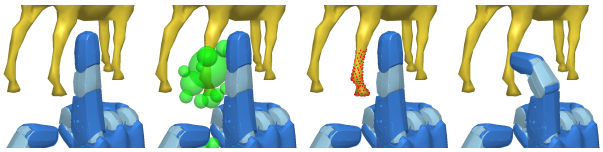

Fig. 2. From the intersection between the finger workspace approximation (second image from the left), a set of reachable points is obtained (third image), before on of them is selected as a contact points (fourth image).

6) Stability filter and quality score: The stability test is based on a *point contact with friction* model, that explains why at least three contacts are required. From the contact positions and normals, we compute a stability score. It is based on a force closure test and stability criterion [1]. All the grasps that do not verify force-closure are rejected. We also compute and add a second score that is the distance to the mass center of the object. The stability score is not sufficient to discriminate good grasps so we build a more general quality score.

Several aspects can be taken into account to compute a grasp quality measure [9]. A tradeoff is often chosen with a score that is a weighted sum of several measures. We chose to combine the previous stability criterion with two other criteria: A finger force ellipsoid major axis score and a contact curvature score. The idea behind the first one is that it is preferable to favor contact such that the contact normal is in a direction close to the direction of the major axis of the force ellipsoid, corresponding to the better force transmission ratio.

The curvature score is used to favor contacts where the mean curvature of the object surface is low. In real situation, it will reduce the impact of a misplaced contact as the contact normal will be susceptible to smaller change in a low curvature area than in a high curvature one. Figure 3 shows, on some objects, how low curvature areas are preferable to establish contacts. Curvature is computed for each vertex and then interpolated for each point on the surface from its barycentric coordinates. The curvature is then normalize to be always included in $[0; 1]$.

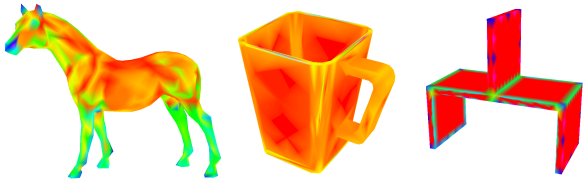

Fig. 3. The mean curvature of the object surface is used as a quality criterion on the contact position. Surface color varies from red (low curvature) to blue (high curvature), through green.

Fig. 4 shows some of the best quality grasps computed with our algorithm for an object with a complex shape.

B. Selecting the good grasp

Once the grasp list of the specified object is computed, the on-line phase begins, whose first step is the selection of a grasp from the list. If there were no additional constraints, the

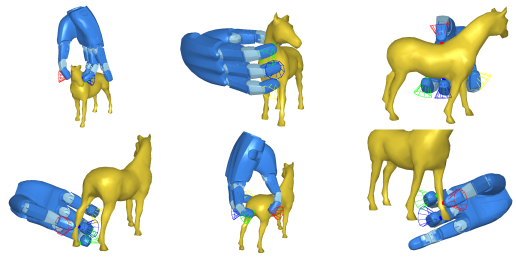

Fig. 4. Some of the various grasps that were computed for an object with a complex shape. (Friction cones are also represented).

best choice would be the grasp with the highest quality score. However, the robot must consider the limited reachability of its arm and the safety of the human handing over the object. Ensuring the grasp is reachable just requires an inverse kinematics computation. Human safety is more complex. We could expect the human to stay passive as he/she is supposed to play a passive role in the exchange. However, as it was demonstrated by the work of Edsinger and Kemp [4], the human tends to move his/her hand to make it match the robot's hand estimated grasp (see section II). This leads us to consider the general case of a cooperative human and the less likely but possible case of an uncooperative human.

1) Cooperative Human: The human interprates the robot's motion and tries to assist it. To facilitate the convergence of the robot's and human's motions, it seems then better to make the robot's motion as easy as possible to interpret. Minimizing the rotation movement of the hand and its length will help the human understand where the robot wants to grasp the object. Therefore, from the robot's initial position, we select, from the list, the grasp that minimizes these two criteria. In the general case, to keep the legibility of the approach motion, the robot must keep the same target grasp.

2) Uncooperative Human: From the previous considerations, we expect the human to move his/her hand to help the robot catching the object, in a quite smooth manner. However, it may happen that the human changes abruptly the way he/she holds the object, by game of for another reason. In this case, the robot may have to select another, more adapted, grasp. The ideas behind the grasp selection remain valid but the robot has to know when such an event occurs. ...

IV. Approach motion of the robot's arm

The canonical task of exchanging an object between human and robot shows the necessity for different elementary planner. One of the first lment is to choose the object transfert point. The figure 5 show an example of cost associated to the postures of a human. For safety and comfort of the human, the robot must not come too close to the human and stay in sight. The figure 6 shows two examples of cost maps used to define the safety zone. Rapidly-Exploring Random Trees RRT algorithms are used to compute a path taking into account the cost maps.

The path computed is then transformed in a trajectory. We use the soft motion planner that limit velocity, acceleration

Fig. 5. Cost map associated to the arm comfort based on kinematic configuration of the arm.

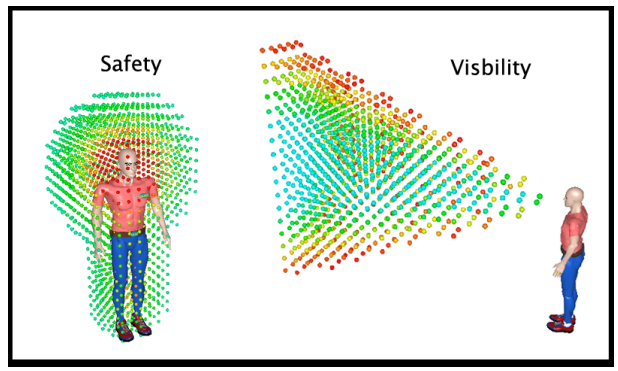

Fig. 6. Safety and visibility cost maps

and jerk to build a trajectory as a series of cubic functions for each axes. This trajectory planner use the same collision checker as the RRT planner. The figure 7 shows how a path defined by straight line segment is transformed in a trajectory defined by a serie of cubic functions.

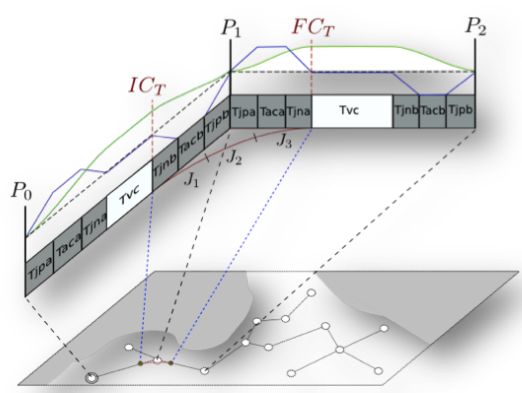

Fig. 7. The soft motion planner transform a path in a trajectory

V. Force feedback of the object exchange

The exchange, one the robots and the human are both contacting the object, needs to define the instant when to open or close the fingers. In a first time, we developed an object to record forces during exchanges (see figure 8. Using wavelet

analysis, we propose a solution to detect when the human is in contact with the object and the robot can release the object.

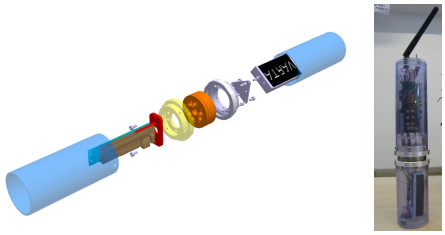

Fig. 8. The object Bidule composed of a 6D force sensor, a 3D accelerometer sensor and a Gumstix embedded computer with Wi-Fi link

VI. Preliminary Results

The robot Jido is built up with a Neobotix mobile platform MP-L655 and a Kuka LWR-IV arm. All the module presented here are embedded on this platform and are very promising. But results are still preliminary and demonstrate each elements. We are working on a generic system capable to do autonomous tasks interactively with humans. Pick and give or receive and place are the first tasks we are investigating. The figure 9 shows the robot giving objects to a human.

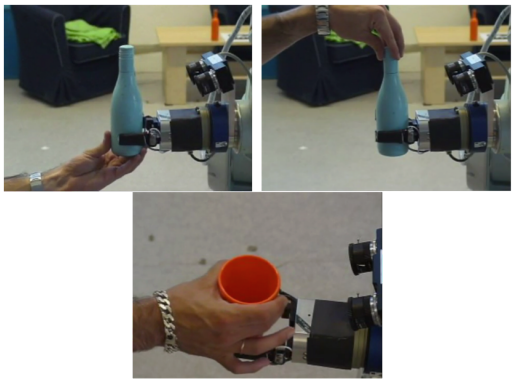

Fig. 9. A human grasp a bottle from the bottom (left) and from the neck (Middle). A power grasp of a glass (right)

VII. Future Work

Several problems remain to be solved to have a complete framework to reactively and safely take the object given by the human. Visual perception is clearly a major difficulty, as it must be real-time, deal with the likely partial occlusion of the object by the human hand and track objects and human movements. As it is not a focus of the DEXMART project or of our research group, we plan to rely on existing techniques or to bypass the difficulty with the use of specific hardware like motion capture system. More interesting topics, from our point of view, concern the decision procedures of the robot. Particularly, knowing when the robot grasps the object is not an easy task and should rely on several sensor modalities like vision and tactile or finger joint torque sensors. At last, the quality of the realized grasp must be evaluated. Unfortunately the Schunk hand we use is not equipped with tactile sensors to determine the contact points.

VIII. Conclusion

We have presented our progress in the development of a robotic framework to reactively grasp an object handed over by a human. The work achieved so far concerns the phases of decision (computation and selection of an adapted grasp) and planning of the approach motion. The remaining work mainly concerns the phase when robot and human briefly hold the object together. During this phase, the robot must decide when it can grip the object and bring it back. This decision requires a fine interpretation of the interaction forces between human and robot grasps. We have introduced a smart device we designed and built, that is used to measure such forces, in order to analyze them. We presented some preliminary result on the force interpretation.

Acknowledgment

The research has been funded by the EC Seventh Framework Programme (FP7) under grant agreement no. 216239 as part of the IP DEXMART.

References

[1] B. Bounab, D. Sidobre, and A. Zaatri. Central axis approach for computing n-finger force-closure grasps. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1169–1174, May 2008.

[2] S. Chitta, B. Cohen, and M. Likhachev. Planning for autonomous door opening with a mobile manipulator. In *ICRA*, 2010.

[3] M. A. Diftler, R. O. Ambrose, K. S. Tyree, S. M. Goza, and E. L. Huber. A mobile autonomous humanoid assistant. In *Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robot*, 2004.

[4] A. Edsinger and C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *IEEE RO-MAN 2007, 16th IEEE International Symposium on Robot Human Interactive Communication*, pages 1167–1172, Jeju Island, Korea, 2007.

[5] S. Haddadin, A. Albu-Schaffer, and G. Hirzinger. Safety evaluation of physical human-robot interaction via crash-testing, robotics. In *Robotics: Science and Systems Conference, RSS*, Atlanta, USA, June 2007.

[6] A. Jain and C. Kemp. El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28(1):45–64, 2010.

[7] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc. An all-solid-state optical range camera for 3d real-time imaging with sub-centimeter depth resolution (swissranger). volume 5249, pages 534–545. SPIE, 2004.

[8] A. Saxena, J. Driemeyer, and A. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, 2008.

[9] R. Suarez, M. Roa, and J. Cornella. Grasp quality measures. In *Universitat Politecnica de Catalunya (UPC), Technical Report*, 2006.

[10] A. Yershova and S. LaValle. Deterministic sampling methods for spheres and so(3). In *IEEE International Conference on Robotics and Automation*, 2004.