

PLANNING STACKING OPERATIONS WITH AN UNKNOWN NUMBER OF OBJECTS

Lluís Trilla, Guillem Alenyà

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain
lluistr@gmail.com, galenya@iri.upc.edu

Keywords: Stack planification, symbolic POMDP, Time-of-Flight camera

Abstract: A planning framework is proposed for the task of cleaning a table and stack an unknown number of objects of different size on a tray. We propose to divide this problem in two, and combine two different planning algorithms. One, plan hand motions in the euclidean space to be able to move the hand in a noisy scenario using a novel Time-of-Flight camera (ToF) to perform the perception of the environment. The other one, chooses the strategy to effectively clean the table, considering the symbolic position of the objects, and also its size for stacking considerations. Our formulation does not use information about the number of objects available, and thus is general in this sense. Also, it can deal with different object sizes, planning adequately to stack them. The special definition of the possible actions allows a simple and elegant way of characterizing the problem, and is one of the key ingredients of the proposed solution. Some experiments are provided in simulated and real scenarios that validate our approach.

1 INTRODUCTION

Algorithms for planning explicitly considering uncertainty have been widely used in the field of mobile robots (LaValle, 2004; Thrun et al., 2005), but are less common in robotic-arm manipulation and grasping (Hsiao et al., 2007). In that scenario uncertainty is especially important and should be carefully considered because contact takes place between the robot and the world. In this interaction, the position of the object and the robot in the world cannot be precisely known, even more if we consider uncertainty in sensors we use to sense this world.

In this paper we want to explore object grasping and stacking tasks, as they are interesting and challenging skills (Kemp et al., 2007). In order to deal with these problems the *partially observable Markov decision process* paradigm will be used, specifically the *discrete model based POMDP*. It provides the capacity of dealing with uncertainty in observations and actions, usually a robot will have an approximation of reality when is sensing the environment and evaluating the results of the actions completed. POMDP have been used before in the context arm motion control for



Figure 1: The robotic arm used in the experiments executing the policy computed by the planner

grasping, however perceptions used are simpler than here, i.e. on/off signals from pressure sensors on the fingers of the hand (Glashan et al., 2007).

The system uncertainty is modeled by measuring it in the real system and providing the values to the system, so it can take into account the various difficult situations it could face according to the chosen action. One interesting characteristic of our approach is that two different POMDPs are combined and one of them

can control the other one and get feedback from it. We will apply this approach to solve a real situation: to clean a table and stack an unknown number of objects of different size on a tray.

The first POMDP will control robotic arm trajectory to prepare the grasping task, planning in the *space state* formed by the relative coordinates to the target state. This approach is extensively reported in (Trilla, 2009). Here will be briefly introduced. A naive approach to avoid the POMDP complex mechanism is a simple reactive algorithm. However, within this approach is difficult to take into account the uncertainty in the number of stacked objects and the probability of stacks falling down.

The second POMDP will plan symbolically the strategy of the cleaning task and the actions chosen for the target of the first POMDP. The objective of the second POMDP is either to completely clean the table or fill the tray. The planification is symbolic because it does not rely on the coordinates of the objects or its interaction with the world, but on an abstraction layer. Two main considerations are important. First, observations are partial: the number of objects on the table is unknown, i.e. because of possible occlusions, and some objects maybe are pre-stacked on the table and this is difficult to observe. Second, the tray surface is limited so it has to stack objects. Here the planning has to deal with objects of different size.

Perception in grasping applications is generally performed using artificial vision to recognize some object characteristics, and then plan a correct grasp (Saxena et al., 2008). Here we will use a relatively new sensor, a ToF (Time of Flight) camera. This camera delivers 3D images at 25fps, potentially allowing fast perception algorithms and, contrarily to stereo systems, it does not rely on computing depth on texture or other object surface characteristics. Depth information will be used to identify the position of the robot hand in the space, and to easily separate objects from background.

This article is structured as follows. POMDP background is introduced in Section 2. The planification strategy is introduced in Section 3, and in particular the planification of the symbolic steps that are involved in the container-content manipulation (Section 3.1). In Section 4 some experiments are presented, validating our approach in a general stacking case, in the case of different object sizes, and with occlusions between objects. Finally, Section 5 is devoted to the conclusions and future work.

2 POMDP BACKGROUND

A POMDP models a sequence of events in discrete states and time where the agent chooses actions to perform. It is represented by the tuple (S, A, T, R, O) where S is the finite set of *states* and A is a discrete set of *actions*. The *transition model* $T(s, a, s')$ describes the probability of a transition from a state s to s' when the action a is performed. The *reward model* $R(s, a)$ defines the numeric reward given to the agent when it executes an action a being in state s . *Observation model* $O(z, a, s)$ describes the probability of an observation z when the action a is performed, and the state is s . A POMDP handles partially observable environments, there is only an indirect representation of the state of the world. The *belief state* b is the probability distribution over all states in the model. At each time step the belief state is updated by Bayesian forward-filtering.

A decision about which action is most applicable is given by the policy function which contains the information about the best action to perform for any possible belief distribution. The policy balances the probabilities of a future sequence of events with the expected accumulated reward which has to maximize. Computing a policy is highly intractable with classic exact methods like *value iteration* or *policy iteration*. However, some recent work has been devoted to find approximated solutions (Hsu et al., 2007), as *point based value iteration* (Hsu et al., 2008), *discrete Perseus* or HSVI are quite fast and yield good results.

3 PLANNING AND EXECUTING THE MANIPULATION OPERATIONS

We divide the high level task of cleaning a table of several objects in two different levels. First, it is important to decide which object to first manipulate. Then, the next issue is to know the exact position of the object and effectively manipulate it. Here we will present our development for the planning algorithm which deals symbolically with the problem and performs high level task.

The effective manipulation of the objects can be solved by means of a classical control algorithms. Alternatively, we have recently proposed to solve the low level task defining also a POMDP in the space of discretized hand positions (Trilla, 2009). With this approach we are able to deal with robots with low precision or repeatability, and also with mobile robot manipulators that naturally are not exactly placed equally

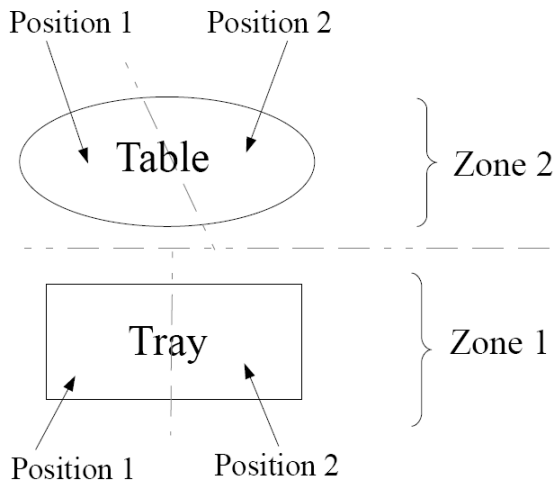


Figure 2: General schema of the working space, detailing the different labeled areas. Table can be divided in several regions in general. Here is divided in two, like in the experiments.

in front of a table.

3.1 Planning the strategy

We define the problem as follows. The working area is divided in zones: one for the tray and the rest for the table (see Fig 2). Each zone is divided in positions where the objects can be placed. Space state is defined as the set (zo, p, sn, t) , where zo (zone) and p (position) identify the object location symbolically, sn indicates the number of objects stacked in the same place, and t is the type of the objects in that place.

The actions are defined to move the objects in the following order:

1. move from the table to the first position on the tray
2. move from the table to the second position of the tray
3. stack the objects on the table freeing up a position.

Using only these three actions is possible to successfully complete the cleaning table task. When two objects are of the same size the system not needs to specify which one of the objects on the table is going to be stacked first. The transition model can be designed including any previous criteria so can be moved first the larger objects or those which are in a certain zone (the closest to the tray for instance).

We want to consider an unknown number of objects into the table. This is appropriate for modelling two different situations: when the point of view of the observer doesn't allow to view completely the whole scene, and when occlusions shadow the correct number of objects.

To take into account an unknown number of objects on the table we propose to only consider two of them at each time. The approach can be extended to take into account more objects/positions, but then the complexity of the solution also increases. Once a position on the table is empty that position is assigned to the next object. This assignment could be done by any criteria, proximity, size of the object (first the larger ones then the smaller ones), or randomly as in our experiments. This represents the action uncertainty: when an action is performed the next state can have an empty position or the transition can go to a different state with objects in that position. The objects can be found already stacked on the table, but the probability of success in such a situation decreases, reflecting that is easier to manipulate a unique object that a stack of two or more objects. In this way planning can be done with no previous knowledge about the number of the objects on the table, and lets the planner to adapt to different situations.

Observations indicate the number and type of objects stacked. The uncertainty of the observations is the number of objects stacked in each position and is modeled as a Gaussian probability distribution over the belief state. The container-content problem (which objects can be or not stacked) is related to the type of object and the symbolic POMDP has to chose the proper stacking actions maximizing the available space on the tray.

Finally, the rewards are defined as positive when the objects are placed on the tray, move objects have a cost and stacking movements have a higher cost. The aim is reached when the table is clean or the tray is full. A great advantage of this model is the versatility of its policy so it can adapt to different situations in order to reach the aim for many scenarios.

Observe that our proposal is not assuming neither implicitly nor explicitly the success in the motion operation. As usual, after the execution of each transfer action the state is observed and the appropriate policy is chosen in accordance. Failed grasping operations are naturally handled, as the state will not change. However, if the object accidentally falls down in the middle of the operation and its position is outside the table area this particular object will not be recovered with the presented approach.

4 EXPERIMENTS

4.1 The observation model

For the observations we want to model a SR3000 SwissRanger camera. This sensor delivers images of

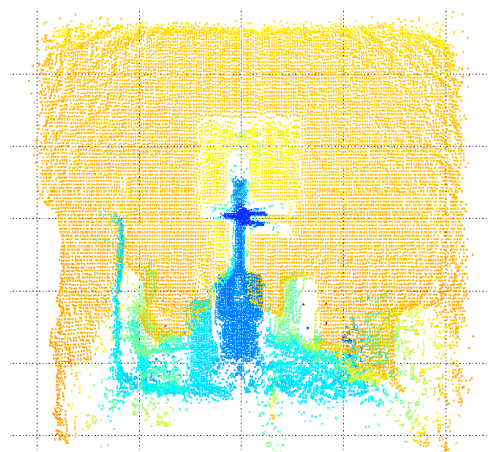


Figure 3: Depth image provided by the ToF camera and used to compute the perceptions. Colors correspond to different depths. Observe the robot horizontally centered in the image, and the hand with its three fingers in the center.

160×120 pixels with depth information for each pixel at 25fps using ToF (time-of-flight) principle. ToF cameras emit modulated light, and depth computation is based on measures on the reflected light using the well known time-of-flight principle (Kolb et al., 2008). Fig. 3 shows our testing scenario as perceived by the ToF camera. We use depth to easily segment the robot and the object from the background, but depth information is an important cue that allows to compute also several of the required grasping parameters (Kuehnle et al., 2008).

Depth measures from a ToF camera are noisy, and systematic errors are present and depend on the computed depth and on the position of the pixel onto the sensor. To obtain precise measures some calibration procedures have to be performed (Kahlmann et al., 2006; Lindner and Kolb, 2006). Also, some other non-systematic errors affect the camera readings, i.e. surface orientation, irregular illuminated zones. ToF camera calibration is still an active field of research and a lot of work has been done to completely understand the source of errors and compensate them.

This sensor is very interesting as offers the capability of measuring the 3D position of the robot hand as well as the position of the different objects on the table (Fig. 3). It is possible without calibrating the camera-robot system to control the motion of the robot (Trilla, 2009). However, the resolution of the sensor is quite small, so it is justifiable to state that it is difficult to determine in our scenario if one glass is alone or it is stacked with other glasses.

The required uncertainties are measured using this scenario.

4.2 Symbolic planification

Once it is defined with proper values, the symbolic POMDP has been tested with simulations facing it with several scenarios and checking its decisions and the sequence of actions. This experiment is defined with two zones and two positions in each zone. There are two types of glasses: big ones and little ones. The transitions are restricted to stack big glasses with big glasses, the little glasses can be stacked inside the big or the little glasses indifferently. The number of glasses stacked in the same position is restricted to four and only is defined the type of the top and bottom of the glass because they are involved in the container-content problem. The states are determined by all the possible combinations of this variables.

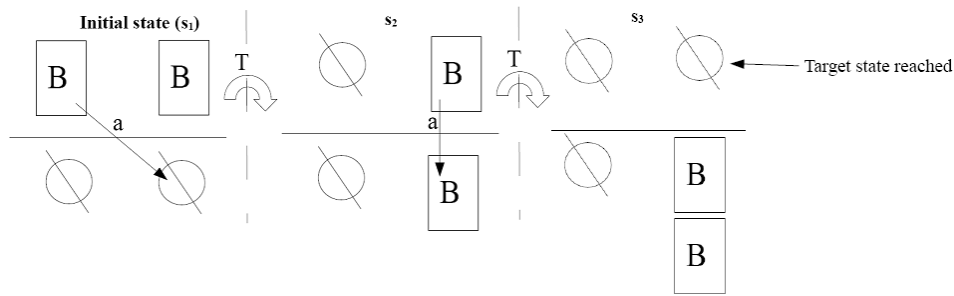
As explained before, choosing this state space the planner becomes able to deal with any quantity of objects on the table. It will not perform the optimal sequence of actions because it has not previous information about the total number of the objects. However, using this method is not necessary to recalculate the policy for different amount of objects and is not limited to a maximum number of them.

The probability of reaching any of the possible states after performing an action is defined as uniformly distributed in the transition model. Consequently, after moving a glass there is the same probability of reaching a state with an empty position or a full one. This transition is decided randomly and it gives the uncertainty over the actions results. Observations give a probability distribution for the belief state over the quantity of objects stacked in the same position, there is uncertainty about the number of glasses.

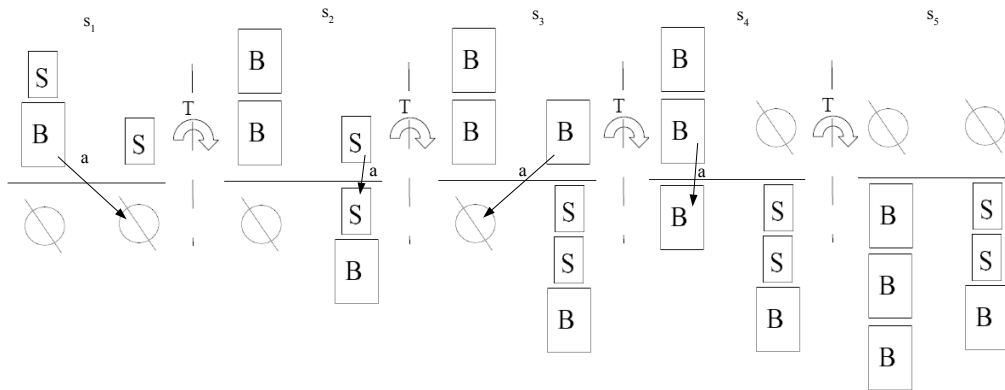
The rewards are defined as follows, stacking operations have a cost because the pile becomes more unstable, place the glasses on the tray is rewarded positively and reach the aim has a positive reward as well. Stack four or more glasses has a penalty (more than four glasses is considered the same state) because it is a too unstable pile and it has a high risk of falling over.

Some results of the carried out experiments are shown in Figs. 4(a) and 4(b), that describe how POMDP-S solves the proposed task. In this diagrams, boxes marked with B represent big glasses and boxes with S are the small ones; empty positions are represented by a circle and the action to perform with an arrow; the top side is the table and the bottom side is the tray; states are indicated as s_i ; between two states there is a transition T; objects placed together mean that they are stacked vertically.

The first diagram (Fig. 4(a)) represents a simple



(a) Simple example, where two big objects are cleaned from the table. Observe that planner decides to create a stack. This is right solution considering that only one object is observed in each zone, as new objects can be observed, i.e. due to occlusions.



(b) A complex example to demonstrate our algorithm capabilities. Regions with several objects are discovered in states s_2 and s_3 , before the translations of objects to the tray. The restriction on the size of the object for stacking is correctly handled in transitions s_3 to s_4 and also s_4 to s_5 where big objects are not stacked onto a small one

Figure 4: Diagram of two experiments results. Boxes marked with B represent big glasses and boxes with S are the small ones. Empty positions are represented by a circle and the action to perform with an arrow. The top side is the table and the bottom side is the tray. States are indicated as s_i . Between two states there is a transition T. Objects placed together mean that they are stacked vertically.

situation where there are only two big glasses on the table. It is worth noting how the planner decides, on the last action s_3 , to stack them on the tray instead of placing them separately, even knowing that this action has a higher cost. The explanation is that, as the number of objects is unknown, exists the possibility that more objects appear after the last movement (due to an occlusion), so planner decides that it is safer to keep a free space available on the tray.

In the second diagram (Fig. 4(b)) there is a more complex situation, with several objects on the table, several of them already stacked and occluded. An example of this occlusion fact can be seen in the transition from s_1 to s_2 . The agent is moving two stacked objects to an empty position on the tray, but in the position that represents this table zone two new stacked objects appear. Note that this happens also in transition from s_2 to s_3 , with a big objects appearing when

a small one is moved.

The correct planification using the size of the objects in the stacking strategy can be seen in the transition from s_3 to s_4 . Only big objects are present, but in the top of the stack there is a small object, so the empty tray position is preferred. The situation is similar in transition from s_4 to s_5 .

In our approach, if the agent reaches a state with no possible action (i.e. there exists no transition) the algorithm ends. This could happen prematurely if the objects are not stacked properly. In the next section we will envisage some possible improvements to handle this behavior.

5 CONCLUSIONS AND FUTURE WORK

We have presented a planning framework, combining two different POMDP approaches, that effectively solve the task of cleaning a table and stack an unknown number of objects of different size on a tray

The symbolic POMDP chooses the adequate combination of actions to clean the table. Here the state space design is crucial as we have some restrictions. First, we assumed explicitly that the number of objects on the table was not observable, i.e. due to occlusions between objects, because it is hard to estimate the number of objects when they are stacked, or purely by noise in the observations. Second, the solution to the problem implies stacking objects of different size with some restrictions, i.e. a big object cannot be stacked onto a small one.

Thanks to the proposed space state and action space, the codification of the different positions and sizes of the glasses lets the planner generate a policy able to deal with the container-contents problem stacking the glasses in the best way, maximizing the free space on the tray. The solution we propose to be able to handle the unknown number of objects is based on focus the attention on only one of the objects present in the each one of the different regions on the table, and it has turned out to be particularly adequate and powerful.

5.1 Future work

Here we have considered that the objects can be already stacked on the table, and that the the robot can perform stacking actions on the table. However this condition has hardly raised in our experiments. One challenge we are facing now is to incentive object stacking actions on the table, before putting them in the tray. A new variable is needed here to balance the cost between two transportation actions and one stacking plus a transportation operation. In the computation of this cost the trajectory from the table to the tray in the transportation action becomes important, as we want to stack closer objects, or more important, stack objects that are in the transportation trajectory to the tray. Our formulation is general in the number defined zones on the table, so we face this new condition as a natural extent of the presented algorithm.

We have considered to add an additional action when observations are not enough to decide for one action, i.e. in order to gather information about the number of glasses stacked on each position. A promising option is an asking action to an operator where the answer could modify the agent's belief

state (Armstrong-Crews and Veloso, 2007).

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation under project DPI2008-06022 and the Generalitat de Catalunya under the consolidated Robotics Group. G. Alenyà was supported by the CSIC under a JAE-Doc Fellowship.

REFERENCES

- Armstrong-Crews, N. and Veloso, M. (2007). Oracular partially observable markov decision processes: A very special case. In *Proc. IEEE Int. Conf. Robot. Automat.*, Rome., pages 2477–2482.
- Glashan, R., Hsiao, K., Kaelbling, L. P., and Lozano-Pérez, T. (2007). Grasping POMDPs: Theory and experiments. In *RSS Workshop: manip. for human env.*
- Hsiao, K., Kaelbling, L. P., and Lozano-Pérez, T. (2007). Grasping POMDPs. In *Proc. IEEE Int. Conf. Robot. Automat.*, Rome., pages 4685–4692.
- Hsu, D., Lee, W., and Rong, N. (2007). What makes some POMDP problems easy to approximate? In *Advances in Neural Information Processing Systems (NIPS)*.
- Hsu, D., Lee, W., and Rong, N. (2008). A point-based pomdp planner for target tracking. In *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena.
- Kahlmann, T., Remondino, F., and Ingensand, H. (2006). Calibration for increased accuracy of the range imaging camera SwissrangerTM. In *ISPRS Commission V Symposium*, pp. 136–141, Dresden.
- Kemp, C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments. *IEEE Robot. Automat. Mag.*, 14(1):20–2.
- Kolb, A., Barth, E., and Koch, R. (2008). ToF-sensors: New dimensions for realism and interactivity. In *Proc. IEEE CVPR Workshops*, vol. 1-3, pp. 1518–1523.
- Kuehnle, J. U., Xue, Z., Stotz, M., Zoellner, J. M., Verl, A., and Dillmann, R. (2008). Grasping in depth maps of time-of-flight cameras. In *Proc. Int. Workshop Robotic Sensors Environments*, pp. 132–137.
- LaValle, S.M. (2004). *Planning Algorithms*. Cambridge UP
- Lindner, M. and Kolb, A. (2006). Lateral and depth calibration of PMD-distance Sensors. In *Proc. 2nd Int. Sym. Visual Computing*, vol. 4292, pp. 524–533.
- Saxena, A., Driemeyer, J., and Ng, A. Y. (2008). Robotic grasping of novel objects using vision. *Int. J. Robot. Res.*, 27:157–173.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge.
- Trilla, L. (2009). Planificació de moviments en entorns amb incertesa per a manipulació d'objectes. Master's thesis, Universitat Politècnica de Catalunya.