Unifying Perception, Estimation and Action for Mobile Manipulation via Belief Space Planning

Leslie Pack Kaelbling and Tomás Lozano-Pérez

Abstract— **In this paper, we describe an integrated strategy for planning, perception, state-estimation and action in complex mobile manipulation domains. The strategy is based on planning in the *belief space* of probability distribution over states. Our planning approach is based on hierarchical symbolic regression (pre-image back-chaining). We develop a vocabulary of fluents that describe sets of belief states, which are goals and subgoals in the planning process. We show that a relatively small set of symbolic operators lead to task-oriented perception in support of the manipulation goals.**

I. INTRODUCTION

A mobile robot in a complex environment can never be completely certain about the state of the environment. This is not a problem that can be corrected by improved sensing; parts of the environment are simply not visible — spaces behind walls, doors and other objects. Thus a robot will have to take sensing actions, including pointing its cameras, moving to new poses in order to get a better view, or moving objects out of the way in support of doing high-level tasks, such as putting objects away in a kitchen. In this paper, we describe an integrated strategy for planning, perception, state-estimation and action in complex mobile manipulation domains.

We have developed an approach to combined task and motion planning that integrates geometric and symbolic representations in an aggressively hierarchical planning architecture, called HPN [1]. The hierarchical decomposition allows efficient solution of problems with very long horizons and the symbolic representations support abstraction in complex domains with large numbers of objects and are integrated effectively with the detailed geometric models that support motion planning. We extended this approach to handle uncertainty by changing the space in which we plan from the space of underlying configurations of the robot and objects to the *belief space* of probability distributions over configurations of the robot and objects [2]. In this paper, we develop a fully probabilistic representation of relative uncertainty between objects in the world and an explicit representation of observed space. This much richer representations of belief space supports both state estimation using a very high-fidelity model and planning using an abstract symbolic representation.

This work was supported in part by the NSF under Grant No. 019868, in part by ONR MURI grant N00014-09-1-1051, in part by AFOSR grant AOARD-104135 and in part by Singapore Ministry of Education under a grant to the Singapore-MIT International Design Center.

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139, USA lpk@mit.edu, tlp@mit.edu

What should the robot represent about its environment? In order to manipulate objects, it needs to know their poses, and the poses of other nearby objects fairly accurately. In order to move through space (both with base and arms), it needs to know whether that space is free. Early work in mapping explicitly represented knowledge about free space in occupancy grids, and scan-based SLAM methods implicitly contain information about where the robot has looked. In mapping methods based on object pose estimation, however, the focus has been on explicit representation of object poses, with an implicit assumption that all space not explicitly marked as occupied is in fact free.

We propose a two-part state representation, with explicit pose estimation of the robot and objects in the world, together with a variable-resolution grid representation of the parts of space that have been recently observed. Thus, if a region of space has been recently observed and doesn't overlap any of the known objects, the robot believes with high probability that the space is free and can be traversed.

Our planning approach is based on hierarchical symbolic planning, using a technique that is called *regression* in the AI planning literature and *pre-image back-chaining* in the robotics literature. We develop a vocabulary of logical fluents that describe sets of belief states; these sets serve as goals and subgoals in the planning process. These fluents can never jointly represent a belief state in complete detail, but are grounded in procedures that test their truth in the current detailed belief state. They are just sufficiently detailed to support planning while maintaining a tractably small representation of complex subgoals.

We briefly touch on related work, then describe the underlying representation for belief states. We go on to describe the formalism used by the planner and give example planning operator descriptions that illustrate interactions between sensing and acting. Finally, we demonstrate the proposed methods on a task with partial information and noisy sensing using a simulated Willow Garage PR2 robot.

II. RELATED WORK

Advances in 3D perception, navigation and motion planning have enabled sophisticated manipulation systems that interleave perception, planning and acting in realistic domains (e.g., [3], [4]). In most such systems, perception tends to be loosely coupled to the task, usually by assigning the perception subsystem the job of constructing some sort of map of the environment, in which planning will be done. There is also a body of work in which perception is actively controlled to achieve some goal. For example, in active vision (e.g., [5]), cameras are controlled to detect objects more effectively. In robot exploration (e.g., [6], [7], [8]), the target locations of the robots are chosen to so as to perceive the most uncertain locations. However, in this existing work, the planning for perception is not driven by or tightly integrated into planning for some larger task, such as manipulation; in most cases, perception itself is the task.

We use planning in belief space to achieve a tight integration of perception and action; perception is used to achieve desired belief states that enable accomplishing a manipulation goal. Belief space, either in the form of logical representation of sets of states [9], [10] or in the form of probability distributions over an underlying state space [11] provides the key representation for integrated planning of perception and action. Recent research [12], [13], [14] has established the value of control in belief space using simplified models and replanning. Our approach to belief space planning builds directly on this work.

III. Example domain

The methods described in this paper are designed to apply broadly to robotics applications involving uncertainty in complicated domains. To make the discussion concrete, we will use a problem domain of picking and placing objects using a Willow Garage PR2 robot. The robot can move its left arm, head, and base. There is significant error in the base odometry, much less error in the arm positioning relative to the base, and negligible error in the head pose. For simplicity in grasp selection and efficient trajectory planning, we limit the robot to grasps that keep its hand parallel to the floor; this is merely expedient and is not essential.

The robot can observe the environment with the stereo sensors on its head, which can be panned and tilted, as well as a scanning laser on its torso. The sensors yield three-dimensional point clouds; a perception service attempts to detect instances of the known shape models in any point clouds that become available. There is error in the detected poses of the objects. Such detection and recognition systems are also susceptible to misclassification (seeing a can where a box is), resulting in data-association problems. We are, for the purposes of this paper, operating under the assumption that any misclassification issues can be handled using robust-statistics strategies and outlier rejection (e.g., [15]) and that unimodal spatial distributions will suffice for all objects.

In this demonstration problem, the robot picks and places objects, relying primarily on visual sensing, but using a simple reactive grasping procedure that uses information from touch sensors on the hand to adjust the fine details of a grasping operation to pick up the object.

IV. Belief state representation

At the heart of any planner is a representation of its knowledge of the state of the world (the belief state), which must be updated after every action (physical or perceptual). The belief state for mobile manipulation under uncertainty needs to contain enough information to support queries both about the nominal (mean or mode) state of the world and about the system's confidence in those estimates. The confidence estimates must be accurate enough to support decision-theoretic trade-offs between, for example, attempting to pick up an object or taking another look to localize it more accurately. It also needs to be a sufficient statistic for the history of observations so that recursive updates may be made effectively. We do not believe there is a good uniform representational strategy for all aspects of information about the domain, so we have separate representations for poses of known objects and for observed space.

A. *Object poses*

When execution is initiated, the robot knows what objects exist in the domain, and knows what their shapes are; but it has a very highly uncertain estimate of their poses. Because our domains currently involve relatively few objects, we simply represent the distribution over their poses, together with the base pose of the robot, using a full joint Gaussian distribution. We assume that the shapes of objects are completely known and that they are always resting stably on a known face and are placed on a known horizontal surface, and thus have four degrees of pose freedom: x, y, z, and θ.

When the robot moves, an odometry error model is used to do an unscented Kalman filter (UKF) update of the belief, with a control input computed as the difference between the uncorrected raw odometry of the robot at the current time and at the time of the previous update. When an object is detected, it is rejected as an outlier if it is highly unlikely in the current model; otherwise, the detection, reported in robot-relative coordinates, is also used in a UKF update, which affects the robot and object poses, as well as other poses through entries in the covariance matrix. Observation noise is currently assumed to be Gaussian and there is no handling of false positive or false negative observations. An additional concern during estimation is the incorporation of physical constraints: objects may not interpenetrate one another and must be supported by other objects (not floating in the air). In parallel work, we have developed an approach to solving this problem [16].

The belief state is augmented with a point estimate of the arm and head configuration of the robot, the object that is currently being grasped by the robot, if any, and the grasp used.

The first row of figure 1 shows four states of the UKF during the course of a planning and execution run. The robot is drawn in its mean pose. The other objects are drawn using their distribution relative to the mean robot pose. The mean pose of the object is drawn in its regular color, and then a "shadow" consisting of the object drawn at poses in which each individual dimension is extended to its positive or negative 95% confidence limit is drawn in gray. In the leftmost frame, there are three known objects: a table, a cupboard, and a cup, and there is substantial uncertainty about their poses. The next frame shows the situation after the robot has made an observation and detected the table and cupboard. They still have moderate uncertainty, but it is dificult to see in this figure; the cup has not yet been

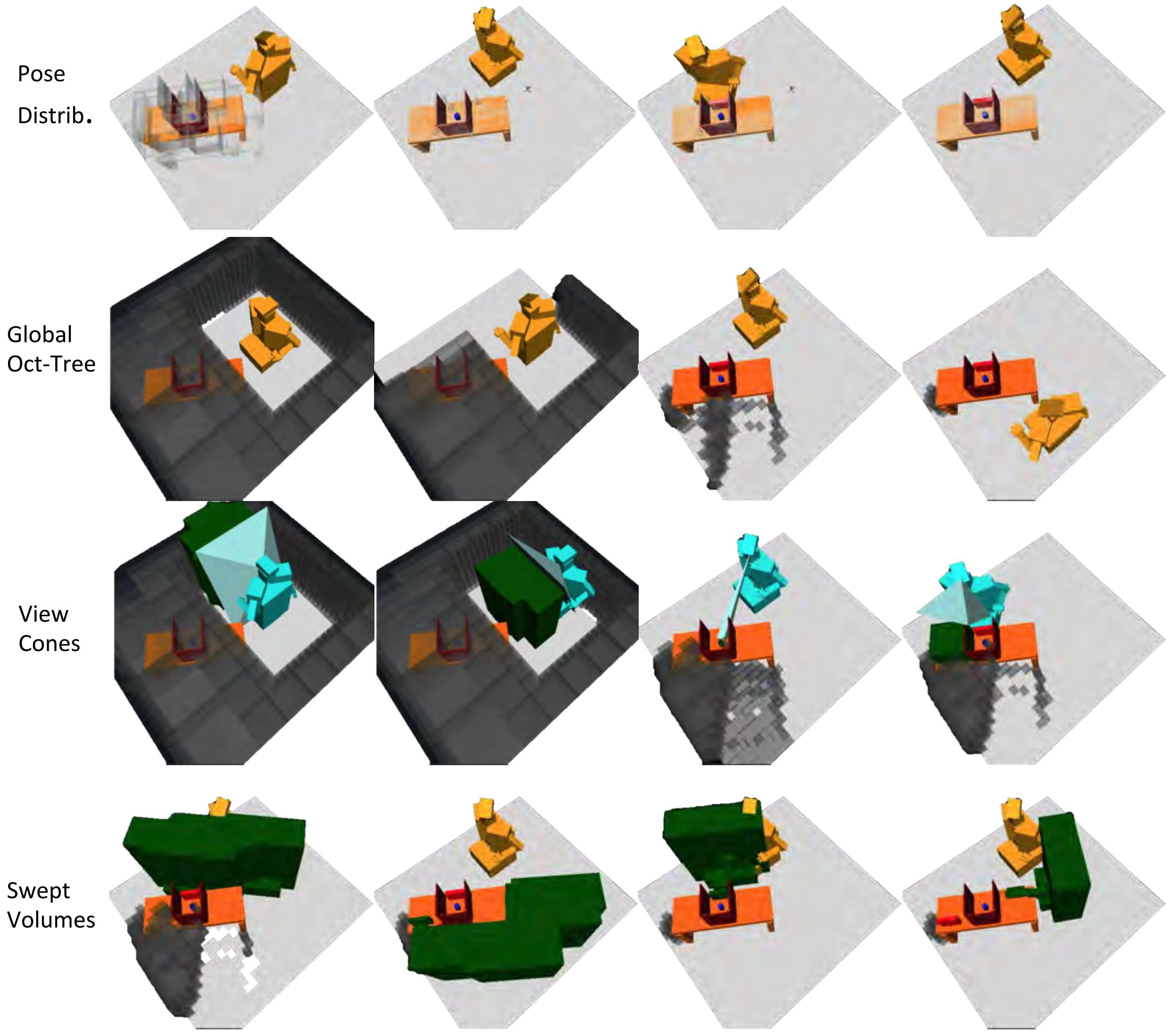

Fig. 1. The first row shows the distribution of the objects relative to the mean robot pose. The second row shows the unobserved space (as gray boxes). The third row shows generated robot poses (cyan), view cones (light blue) and target regions (green) for looking operations. The fourth row shows swept volumes for generated robot motions.

observed. In the third frame, a new object is discovered and added to the state of the filter; but it occluded the view of the cup, so it remains significantly uncertain. In the last frame the cup has been observed and its uncertainty reduced.

B. *Observed space*

Another important query the robot needs to make of the belief state is whether a region of space is known to be clear of obstacles and therefore safe to traverse. To answer such queries, we represent the parts of the space that the robot has recently observed with its depth sensors.

Keeping an accurate probabilistic model of known-clear space is quite difficult: the typical approach in two-dimensional problems is an occupancy grid [17] (recently extended to three-dimensional problems [18]). It requires a detailed decomposition of the space into grid cells and although there are some attempts to handle odometry error in the robot [19], this remains challenging. A more principled strategy would be to maintain the history of robot poses in the UKF, rather than just the current one, and combine the depth maps sensed at each of those poses into a global map of observed space.

We take a much simpler approach, operating under two assumptions: first, that the observed-space maps we construct will only be used in the short term; second, that the mechanisms for detecting objects and tracking their poses will provide a high-accuracy estimate of the poses of material objects. Looking is not too expensive, and objects may be dynamic, so we expect, for instance, when the robot re-enters a room, that it will need to reconstruct the observed-space map. Thus, handling long-distance relative odometry errors is not crucial. For this reason, we simply attach each depth scan to the most likely pose estimate for the robot in the Kalman filter (this is much more accurate than the raw odometry). We integrate the observed-space information from the sequence of scans into an oct-tree representation of the space that has been observed by the robot. This representation of known space need not be as high-resolution as an occupancy grid, which must also represent the boundaries between free and occupied regions of the environment; in our approach, those boundaries are represented by the continuous object-pose distributions in the Kalman filter.

In the following sections, we will denote space that has been observed as $\mathcal{S}_{obs}$. The second row of figure 1 shows the observed-space oct-tree at four different points during execution; space that is filled with dark-grey cells has not yet been observed by the robot. At initialization time, the robot knows the contents of the region of space right around it. As it moves and scans (in this case, using both the scanning laser on the torso as well as the narrow-field stereo cameras on the head), it clears out more of the space, until in the final frame, it has observed most of the observable space in the room. One very important role that the observed-space map plays is to constrain the robot motion planner when it is determining a trajectory for final execution of a motion primitive; any part of the space that has not yet been observed is marked as an obstacle and cannot be traversed.

V. Belief set estimation for planning

When planning in belief space, goals must be described in belief space. Example goals are "With probability greater than 0.95, the cup is in the cupboard." or "The probability that more than 1% of the floor is dirty is less than 0.01." These goals describe *sets* of belief states. The process of planning with pre-image backchaining computes pre-images of goals, which are themselves sets of belief states [2]. Our representational problem is to find a compact yet sufficiently accurate way of describing goals and their pre-images.

In traditional symbolic planning, *fluents* are logical assertions used to represent aspects of the state of the external physical world; conjunctions of fluents are used to describe sets of world states, to specify goals, and to represent pre-images. States in a completely symbolic domain can be represented in complete detail by an assignment of values to all possible fluents in a domain. Real world states in robotics problems, however, are highly complex geometric arrangements of objects and robot configurations which cannot be completely captured in terms of logical fluents. However, logical fluents can be used to characterize the domain at an abstract level for use in the upper levels of hierarchical planning.

We will take a step further and use fluents to characterize aspects of the robot's *belief state*, for specifying goals and pre-images. For example, the condition "With probability greater than 0.95, the cup is in the cupboard," can be written using a fluent such as $BIn(cup, cupboard, 0.95)$, and might serve as a goal for planning. For any fluent, we need to be able to test whether or not it holds in the current belief state, and we must be able to compute the pre-image of a set of belief states described by a conjunction of fluents under each of the robot's actions. Thus, our description of operators will not be in terms of their effect on the state of the external world but in terms of their effect on the fluents that characterize the robot's belief. Our work is informed by related work in partially observed or probabilistic regression (back-chaining) planning [20], [21], [22]. In general, it will be very difficult to characterize the exact pre-image of an operation in belief space; we will strive to provide an approximation that supports the construction of reasonable plans and relies on execution monitoring and replanning to handle errors due to approximation.

We will represent *belief sets* as conjunctions of fluents. Each fluent is a test on an actual belief state: the belief state is in the belief set if all of the fluents test to true.

A. *Fluents for mobile manipulation*

We demonstrate the use of logical fluents for describing belief sets in the mobile manipulation domain. In previous work, we explored this strategy, but with an ad hoc representation of pose uncertainty [2]. In this section, we describe the most important fluents in our formulation, and their definitions in terms of tests on belief states.

We specify conditions on continuous belief distributions, by requiring, for instance, that the mean of the distribution be within some value of the target and the variance be below some threshold. Generally, we would like to derive requirements on beliefs from requirements for action in the physical world. So, in order for a robot to grasp an object, the estimated position of the object needs to be within a tolerance equal to the difference between the width of the open hand and the width of the object (or possibly a different tolerance depending on the robustness of the grasping routine). The variance of the robot's estimate of the object position is not the best measure of how likely the robot is to succeed: instead we will use the concept of the *probability near mode* (PNM) of the distribution. It measures the amount of probability mass within some δ of the mode of the distribution. So, the robot's prediction of its success in grasping the object would be the PNM with δ equal to half of the hand width minus the object width.

The first set of fluents characterize belief about the pose of an object. We start by asserting that the mode of the distribution on the pose of object i, which for a Gaussian corresponds to its mean μ_i, is within δ of a desired pose ϕ:

$$PoseModeNear(i, \phi, \delta) \equiv \|\mu_i - \phi\| < \delta \ .$$

Any appropriate norm can be used here; in our implementation of this domain, the test is actually computed componentwise.

To characterize certainty about a pose, we must specify a frame of reference. Although all poses are expressed in a global coordinate frame, we are typically interested in the variance in the estimate of the pose of one object i, ϕ_i, relative to the pose of another object j, ϕ_j. For example, it may frequently be the case that two objects have very uncertain pose relative to the global coordinate frame, but if they are observed in the same image, they have very certain poses relative to one another. We use the fluent

$$BVRelPose(i, j, \epsilon, \delta) \equiv PNM(\phi_i - \phi_j, \delta) > 1 - \epsilon \ .$$

This fluent is intended to indicate belief states in which we know the "Value" of the relative pose, without specifying what the value will be.

In order to specify conditions on the configuration of the robot, including the base pose, as well as the hand (in this work, we only use one arm of the robot), we use the fluent $ConfModeNear(c, \delta, \delta_g)$. It is true if the mode of the distribution of the robot's configuration is within δ of the configuration c. In this case, it tests to see that both the 4D Cartesian pose of the base and the 4D Cartesian pose of the hand are within δ of those specified in c (in the same manner as for *PoseModeNear*) and that the actual gripper opening is withing δ_g of the gripper opening in c.

The next set of fluents characterize beliefs about the contents of regions in space. The regions under consideration are not pre-discretized, but are computed dynamically during planning as part of the pre-image backchaining process.

The $BContents(r)$ fluent simply asserts that the region r has been completely observed.

$$BContents(r) \equiv r \subseteq \mathcal{S}_{obs} \ .$$

The $BClearX(r, x, \epsilon)$ fluent asserts that the region r is believed to be clear with high confidence except for objects in the set x.

$$\begin{aligned} BClearX(r, x, p) \quad &\equiv \quad BContents(r) \ \& \\ &\quad\ \neg\exists i \notin x. \Pr(overlaps(i, r)) > p \ . \end{aligned}$$

where i is an object, r is a region, and p is a probability. To test this in practice, we construct a union of the volumes obtained by placing the object at the p-percentile pose in each direction of each dimension, then test to see if that p-shadow region overlaps r. A related fluent asserts that a region r is believed to contain an object i.

$$BIn(i, r, p) \equiv \Pr(contains(r, i)) > p \ .$$

It is tested by seeing whether the p-shadow of i relative to r is contained in r.

B. Operator descriptions

Operator descriptions for planning characterize the belief pre-image of an action: the conditions that must be true of a belief state, so that the resulting belief state will satisfy the result condition of the operator. Because our domain dynamics are stochastic, even in the belief space, we cannot guarantee a particular outcome; these operator descriptions characterize the most likely outcome, and we will re-plan whenever that outcome fails to occur.

The following operator descriptions constitute part of the planning domain description that is used to generate the examples in section VI. They illustrate two important ideas: (1) That motion and perception actions need to be tightly integrated in order to achieve goals in complex environments; and (2) that that the general-purpose planning mechanism of logical regression-based planning, applied in belief space, can be used to achieve this integration.

Each operator description has a name, then several components. The tag `pre` indicates a precondition fluent, `let` indicates a quantity that is being computed and named, `exists` indicates a call to a heuristic *generator* procedure that may generate one or more bindings of variables that have large or infinite domains, and `result` indicates a result fluent. These operators are applied backward during planning: the goal is a conjunction of fluents. A step in the planning search is to match a fluent in the current goal with the result fluent of an operator, to remove that result from the goal, then to add the preconditions to the goal; if the new goal is not a contradiction, then it becomes a node in the search tree. Whenever a goal is satisfied in the current belief state (this happens when all of the fluents in the goal are true) then a legal plan has been found.

Following is a description of the `Pick` operator, which results in the robot believing that it is holding the objct `O`, with high probability. One thing to note is that the preconditions are divided into two sets. The first precondition is that the pose of the object `O` be known, with large tolerances, with respect to the robot. Given that, the planner considers places from which the object might be picked up: typically, from the current mode of the distribution of the pose of that object in the `b0`, which is the belief state that holds at planning time, and given that pose, it generates one or more paths `P` that the robot might need to move through in order to pick up the object at that pose. Given these generated values, we establish another set of preconditions: that the swept volume of that path be known to be clear, that the robot not be holding anything, that the robot be known to be close to the pre-grasp configuration that we expected it to be in when we computed the path `P`, and that the robot's configuration, relative to the object's pose, be known highly accurately. If all of these conditions hold, then the primitive procedure that picks up the object will succeed with high probability, resulting in the object being held. The domain description also contains a similar operator description for placing the object. The fact that there are some belief preconditions that must be satisfied before even computing the rest of the preconditions fits naturally into the hierarchical planning framework.

```
Pick(O, ObjPose):
  pre: BVRelPose(O, robot, bigEps, planDelta)
  exists: ObjPose in {modeObjPose(b0)} U generateParking(O)
          P in generatePickPaths(ObjPose)
  pre: PoseModeNear(O, Objpose, planDelta)
       BClearX(sweptVol(P), [O], clearEps)
       BHolding(None, holdingEps)
       ConfModeNear(preGrasp(P), graspDelta)
       BVRelPose(O, robot, eps, graspDelta)

  result: BHolding(O, eps)
```

In our planning domain, the primitive operation of looking at an object (by pointing the robot's head so that the object will be centered in the stereo camera frame) can be used to achieve several belief conditions. The operator description below characterizes how looking can be used to increase certainty of the pose of object O relative to the robot. We generate a configuration of the robot, RobotConf, which includes the configuration of the head, that has the property that if the robot is in that pose and O is at its mean pose, then it is possible to view object O.

Additionally, ViewCone is a volume of space between the robot's camera and the object that must be free in order for the view not to be occluded. The first precondition is interesting: it is that in the most likely state, the view cone is not known to be occluded (note that because of the huge epsilon the "shadows" will simply be the objects at their mean poses). This means that if it is most likely that something is in the way, we will be required to move it out. We rely on the replanning mechanisms of HPN: if, upon looking, it is revealed that there is an object occluding the view, then a new plan will be made that achieves the BClearX condition by removing the occluding object(s).

Next, we require that the robot's configuration be near the one generated for viewing the object. The requirement is currently only on the mean of the robot's pose distribution, so that it is in roughly the right place.

Finally, in order to have as a likely outcome BVRelPose with probability 1 - Eps, we determine the pre-image of that fluent by finding a value PNMRegress(eps, Delta, obsVar), which is larger than Eps, such that if an observation is made with variance obsVar and that starting degree of uncertainty, the desired result will hold (see section V-C). This operation can chain backwards, determining a sequence of several Look operations in order to guarantee the desired resulting confidence.

```
Look(O):
  exists: (ViewConf, ViewCone) in generateViewPose(O)
  pre: BClearX(ViewCone, [O], hugeEps)
       ConfModeNear(ViewConf, lookDelta)
       BVRelPose(O, robot,
                 PNMRegress(eps, Delta, obsVar), Delta)
  result: BVRelPose(O, robot, Eps, Delta)
```

The next two operators don't have actual primitives associated with them: they are essentially definitional, and compute a set of conditions under which the resulting condition will be true; applying the operator during planning simply replaces the target condition with the preconditions at the appropriate level of abstraction.

In order to achieve the condition that region R is known to be clear with the exception of a list of objects, we must first know the contents of R. Then, we require that each of the objects that is overlapping R in the current belief state be moved, with high probability, to a part of the domain called the warehouse; in addition, we establish the requirement that the region be clear of all other objects, as well. This condition will cause the region to be cleared out again if some exogenous process puts objects into it and will prevent other object placements from being made into that region.

```
BClearX:
    pre: BContents(R)
    let: occluders = objectsOverlapping(R, b0) - Exceptions
    pre: BClearX(R, Exceptions + occluders, Eps)
         for o in occluders: BIn(o, 'warehouse', placeEps)
    result: BClearX(R, Exceptions, Eps):
```

Finally, to achieve the condition that the contents of a region R are known, we depend on a recursive decomposition of the region. If the region can be viewed in one look operation, then we generate a configuration for the robot and associated view cone for viewing the region and require the view cone not to be known to be occluded and require the robot configuration to be near the generated one; if the region is too big for a single view, then we split it into subregions, driven partly by the desire to aggregate space that has already been viewed into the same subregion and space that has not been viewed into different subregions. For each of the subregions, we assert that the contents must be known.

```
BContents:
if not viewable(R):
    let: subRegions = split(R, b0)
    pre: for s in subRegions: BContents(s)
else:
    exists: (ViewConf, ViewCone) in generateViewPose(R)
    pre: BClearX(ViewCone, [R], hugeEps)
         ConfModeNear(ViewConf, lookDelta)
```

C. *Regression of fluents*

We defined fluents characterizing aspects of continuous probability distributions, and we use them in operator descriptions. It is necessary to be able to compute the pre-image of a fluent in belief space. We will begin with a simple one-dimensional case, and the describe how it is done for the robot and object pose fluents described in section V-B.

For a planning goal of $BV(X, \epsilon, \delta)$, that is, knowing the amount of probability mass of random variable X that is within δ of the mode is greater than $1 - \epsilon$, we need to know expressions for the regression of that condition under the actions and observations in our domain. In the following, we determine such expressions for the case where the underlying belief distribution on state variable X is Gaussian and the dynamics of X are stationary.

For a one-dimensional random variable $X \sim \mathcal{N}(\mu, \sigma^2)$,

$$P(|X - \mu| < \delta) = \Phi\left(\frac{\delta}{\sigma}\right) - \Phi\left(-\frac{\delta}{\sigma}\right) = \text{erf}\left(\frac{\delta}{\sqrt{2}\sigma}\right) \quad ,$$

where Φ is the Gaussian CDF.

Assume the action is to make an observation, and the observation o is drawn from a Gaussian distribution with mean X and variance σ_o^2. To guarantee that

$$BV(X, \epsilon, \delta) = P(|X - \mu| < \delta) > 1 - \epsilon$$

holds after observing o, we must guarantee that $BV(X, 1 - \mathit{PNMObsRegress}(1 - \epsilon, \delta, \sigma_o^2), \delta)$ holds on the previous step, where $\mathit{PNMObsRegress}(\theta_{t+1}, \delta, \sigma_o^2)$ is

$$\text{erf}\left(\sqrt{\text{erf}^{-1}(\theta_{t+1})^2 - \frac{\delta^2}{2\sigma_o^2}}\right) \quad .$$

Assume that, instead of an observation, the action is to change the random quantity X by an amount Y, then generally there is some loss of certainty about the value of X, characterized by process noise σ_Y, which may be dependent on the value of Y. To guarantee that $BV(X, \epsilon, \delta)$ holds after changing X, we must guarantee that $BV(X, 1 - \mathit{PNMChangeRegress}(1-\epsilon, \delta, \sigma_Y^2), \delta)$ holds on the previous step, where $\mathit{PNMChangeRegress}(\epsilon, \delta, \sigma_Y^2)$ is

$$1 - \operatorname{erf}\left(\frac{\delta \operatorname{erf}^{-1}(1-\epsilon)}{\sqrt{\delta^2 - 2\sigma_Y^2 \operatorname{erf}^{-1}(1-\epsilon)^2}}\right) .$$

If $\epsilon < 1 - \operatorname{erf}\left(\frac{\delta}{\sqrt{2}\sigma_Y}\right)$, then there is no degree of prior certainty that will guarantee that, after the action, the *BV* condition will be satisfied.

To compute regression conditions for *BVRelPose* for example, we might need to compute the preimage of a set of distributions on an entire pose. We are using four-dimensional vectors of ϵ and δ values, however, so we simply regress the epsilons individually for each dimension.

D. Generators

Because the symbolic planner is working in what is essentially an infinite domain (the space of poses, paths, volumes, etc.) it is impossible to create an *a priori* discretization or enumeration of the state space. Instead, we employ heuristic *generator* procedures to generate candidate values of variables, such as paths and poses, that have infinite domains. We have previously described generators (suggesters) for grasps and paths [1]; we now describe some additional considerations for using generators in belief-space planning.

Because we are planning to gain visual information, one important question is where to place the robot in order to get a good view of an object or a region of space that is of interest to the system. The generator works by sampling a set of collision-free robot placements and testing them to see if they satisfy the visibility requirements for the object or region to be viewed and if they can be reached safely by the robot.

The third row of figure 1 shows example results of the view generator. The robot is shown in cyan at the generated pose. The cone emenating from the eye is the view cone, which must be unoccluded in order for this pose to result in a good view of the object. The generator prefers to find poses with view cones that are unoccluded in the belief state at the time of planning; however, if necessary, it will plan to look 'through' movable objects. In that case, the planning operators will construct additional steps in the plan to move the occluding object out of the way before looking. The third frame shows the pose and view cone for looking at a small object in the back of the cupboard; the other frames show view cones for looking at regions of space (shown in green) that had not yet been entirely observed.

Similarly, when the robot needs to move to a new base pose or to pick up or place an object, the high-level planner needs to guarantee that space will be available for that operation. The actual primitive robot motion operations, when they are executed, are planned using an RRT on a high-fidelity model. However, while we are considering multiple different high-level plans, it is not efficient to plan robot motions completely accurately. So, the path generator also uses a visibility-graph planner for an approximate robot, and it only tries to guarantee a path to a 'home' region. If, every time the robot moves, there is guaranteed to be a free path to the 'home' region, then it can never block itself in. The fourth row of figure 1 shows, in green, highly approximate, but conservative swept volumes of the robot moving through the generated paths. It is these volumes that must be determined to be clear before the robot can execute the associated motion or manipulation actions.

VI. Example planning and execution

Figure 2 shows a sequence of images depicting the planning and execution process for an initial goal of placing the small blue cup at one end of the table.

The robot starts with a known area around it, and the rest of the room is unknown—as shown in the first oct-tree in figure 1. To determine the contents of the swept regions of generated motions (the large green regions in the bottom row of figure 1), a series of look motions and view cones are generated (the cyan robot and light blue cones in the third row of figure 1). When these scans are executed (steps 2–7 in figure 2), new areas of the oct-tree become known as illustrated in subsequent oct-trees in figure 1.

After the first two scans (steps 2 and 3 in figure 2), the table has not been observed and so its pose distribution is diffuse – as shown in the first pose distribution in figure 1. Also, the big red object has not been seen; note that it is not part of the initial model. After the table is scanned (in step 4 of figure 2), its pose distribution becomes tight but the blue cup is still not visible (since it is occluded by the big red object), so its pose distribution is still diffuse, as shown in the second pose distribution of figure 1. In the scan of step 4, the red object is also seen and added to the model (as seen in the third pose distribution of figure 1). When going to look at the warehouse region (in step 5 of figure 2), where the red object is to be moved, the robot serendipitously "sees" the blue cup and narrows its distribution, as seen in the fourth pose distribution of of figure 1. If the blue cup had not been seen at this point, a plan would have been constructed to move the red object out of the way so as to enable looking at the blue cup. Steps 6 and 7 of figure 2 are undertaken to ensure that the space that the robots needs to traverse while moving the red block and the blue cup are free of obstructions.

After the required regions are known, the planning and execution proceeds as usual, resulting in a sequence of operations to move the red block to the warehouse, pick up the blue block and take it to its goal location (steps 8-12 of figure 2).

Conclusions. This paper has provided methods for extending belief-space planning techniques to mobile manipulation problems, seamlessly integrating perception and manipulation actions, performing goal-directed perception in service of manipulation and goal-directed manipulation in service of perception.

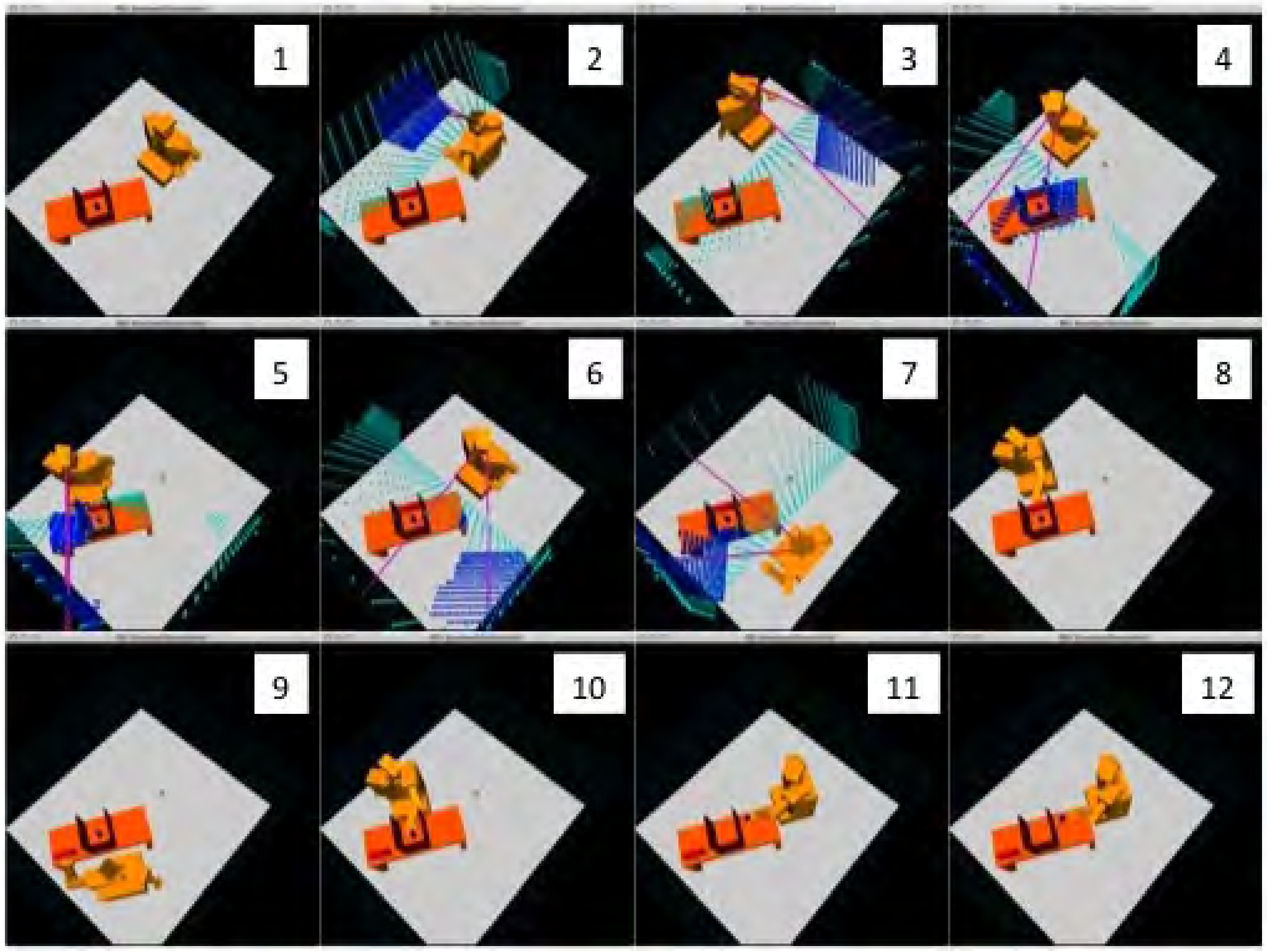

Fig. 2. The key steps in the execution of a plan to place the small blue cup in a target region at one end of the table. The red object is initially not in the object's model of the world. Scans with the head-mounted sensor are shown as dark blue points. Scans with the scanning laser are shown in cyan.

References

[1] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *ICRA*, 2011.

[2] ——, "Pre-image backchaining in the belief space for mobile manipulation," in *ISRR*, 2011.

[3] R. B. Rusu, I. A. Sucan, B. P. Gerkey, S. Chitta, M. Beetz, and L. E. Kavraki, "Real-time perception-guided motion planning for a personal robot," in *IROS*, St. Louis, MO, 2009.

[4] D. Pangercic, M. Tenorth, D. Jain, and M. Beetz, "Combining perception and knowledge for everyday manipulation," in *IROS*, 2010.

[5] E. Sommerlade and I. Reid, "Probabilistic surveillance with multiple active cameras," in *ICRA*, May 2010.

[6] C. Stachniss, *Robotic Mapping and Exploration*, ser. Springer Tracts in Advanced Robotics. Springer, 2009, vol. 55.

[7] P. Wang and K. Gupta, "View planning for exploration via maximal c-space entropy reduction for robot mounted range sensors," *Advanced Robotics*, vol. 21, no. 7, pp. 771–792, 2007.

[8] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *JAIR*, vol. 34, pp. 707–755, 2009.

[9] R. P. A. Petrick and F. Bacchus, "Extending the knowledge-based approach to planning with incomplete information and sensing," in *ICAPS*, 2004, pp. 2–11.

[10] D. Bryce, S. Kambhampati, and D. E. Smith, "Planning graph heuristics for belief space search," *JAIR*, vol. 26, pp. 35–99, 2006.

[11] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, 1998.

[12] T. Erez and W. Smart, "A scalable method for solving high-d continuous POMDPs using local approximation," in *UAI*, 2010.

[13] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *RSS*, 2010.

[14] N. E. D. Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *ICRA*, 2010.

[15] J.-A. Ting, E. Theodorou, and S. Schaal, "Learning an outlier-robust kalman filter," in *ECML*, 2007.

[16] L. L. S. Wong, L. P. Kaelbling, and T. Lozano-Perez, "Collision-free state estimation," in *ICRA*, 2012.

[17] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[18] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[19] A. Souza, A. Santana, R. Britto, L. Gonalves, and A. Medeiros, "Representation of odometry errors on occupancy grids." in *ICINCO-RA (2)*, 2008.

[20] C. Boutilier, "Correlated action effects in decision theoretic regression," in *UAI*, 1997.

[21] C. Fritz and S. A. McIlraith, "Generating optimal plans in highly-dynamic domains," in *UAI*, 2009.

[22] R. B. Scherl, T. C. Son, and C. Baral, "State-based regression with sensing and knowledge," *Intl. J. of Software and Informatics*, vol. 3, no. 1, pp. 3–30, 2009.