Next Best View Estimation With Eye In Hand Camera

Christian Potthast and Gaurav S. Sukhatme

I. Introduction

For many robotic systems, automated data acquisition is an important part of detecting and recognizing objects. This is true especially for robots like the PR2 which are supposed to operate in human environments like our homes. Human living environments are usually highly cluttered. Thus, oftentimes, a scene can not be fully observed from one position only. For scenes like this, data acquisition is treated as a sensor placement problem, also known as the *Next-Best-View* (NBV) problem. The NBV approach seeks a single additional sensor placement in order to improve the knowledge of the environment. However, the set of possible NBV positions is constrained by the robots abilities. For instance, if the sensor is mounted on the head of the robot, the position of the sensor can only be changed by driving the robot around. In environments like kitchens this can make information gathering difficult, as countertops are only accessible from one side, making it difficult for the robot to observe the objects on the countertop from a different angle. We address automatic data acquisition in scenarios like this in this work.

To enable the PR2 to handle constrained positioning settings like this, we equip it with a camera system mounted on the high dimensional manipulator. The camera system we are using for our system is the Microsoft Kinect, which provides us a point cloud representation of the scene. The increase in degrees of freedom allows the robot to choose from a bigger set of possible viewing positions allowing it to observe an area which would otherwise be inaccessible. Additionally, we consider the possibility of planning the trajectory to the NBV position such that the movement can be used to collect even more data.

In the following section we introduce the main ideas of our approach. Then we illustrate and discuss first results obtained through this approach.

II. Approach

Our approach consists of two main phases that are iterated until a scene has been sufficiently observed. The first phase is the next best view estimation, where a pose is sought that delivers maximum expected information gain. The second phase then plans a trajectory to this pose with intermediate waypoints which are utilized to take additional scans. These waypoints are selected such that the expected information gain is maximally. Before describing the details of each phase, we first describe our representation of the environment.

The Authors are with the Robotic Embedded Systems Laboratory, University of Southern California (USC), Los Angeles, CA 90089, USA {potthast, gaurav}@usc.edu

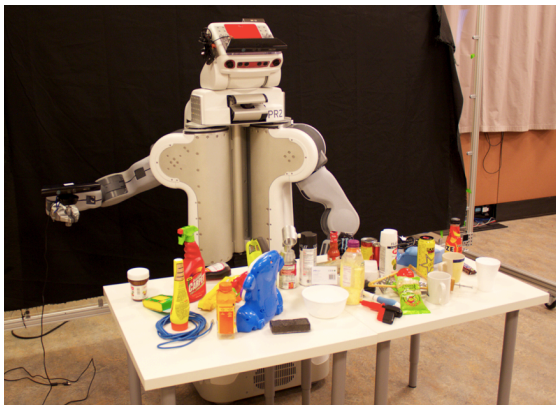


Fig. 1. In the experimental setup the PR2 is positioned in front of a highly cluttered environment. The goal is to maximally observe the scene.

Occupancy grid

The point cloud data produced from the Kinect sensor gives us no information about the space where no measurements have been returned. There is no distinction in point clouds between space which is free and space which could not be measured due to occlusion or range limitations. We need to distinguish between the two types of space, because the unknown space arising from occlusion is the space we are interested in. This unknown space can give us new information about the scene once it is observed. By using a ray traversal algorithm we can distinguish between space we have seen and is actually free and space which we have not seen and is therefore unknown space. We represent this knowledge of the environment with the help of an occupancy grid. Each voxel has a probability distribution over the two possible states free and occupied, where a probability of 0.5 indicates that whether this voxel is free or occupied is unknown. Given new measurements the voxels are updated using a sequential Bayesian update formulation.

Next Best View Estimation

To estimate the pose which maxims the knowledge about the scene we have to solve the NBV problem. In the NBV problem we are given a partially observed scene, represented as an occupancy grid, were we want to find the global optimization pose which maximizes the information gain, by reducing unknown space. From an information theoretical point of view observing a voxel to be occupied is equally valuable as detecting it as as free space. Voxel state changes are therefore a measure of information gain in a new scan.

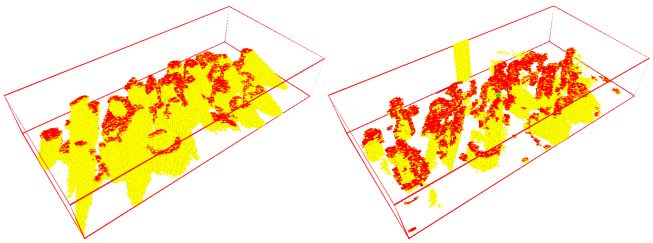
Thus, minimizing the unknown space is equivalent to maximizing the expected number of observed voxels. Intuitively, we aim to find a new viewing position which has the largest information gain compared to all other possible poses. To find this new position we have to make a guess about what we can observe from all possible poses and predict how the occupancy grid would change if we were to take a scan from each pose. This is achieved by taking 'virtual scans' from every possible pose and choosing the pose that promises the most information gain. The pose with the highest information gain is our NBV pose.

Formally, given a set $S = \{s_1, s_2, ..., s_n\}$ of possible viewing position we want to find the next best viewing pose $\hat{s} \in S$. To estimate $\hat{s}$ we define a function $F(s_n)$ which is a measure of how many unknown voxels can be observed from s_n, conducting 'virtual scans' from each pose S. Simply assuming that every voxel can be seen unless the ray to the voxel under question is blocked by an occupied voxel leads to inaccurate information gain estimations. Objects which have not yet been observed could potentially block the view to unknown voxels and therefore limit the actual information gain of a viewing position s_n. This will lead to wrong estimates over the unknown space and the computed $\hat{s}$ is not necessarily the optimal choice. The better we can predict the outcome of a potential new scanning pose s_n the more accurate we can choose $\hat{s}$. In our approach, we propose to compute the next best view pose $\hat{s}$ by choosing the pose s_n with the maximum expected number of observed voxels, where each voxel is associated with a probability of being observed given a pose s.

The observability probability of any unknown voxel is computed by using a decaying kernel function depending on the number of unknown penetrated voxels. The more unknown voxels the ray needs to penetrate, the smaller is the observability probability of that target voxel. Intuitively this expresses the fact that the more unknown space a ray penetrates, the more likely it is that the ray will be blocked by an object which has not yet been seen. The NBV pose $\hat{s}$ is estimated by choosing the pose s_n with the maximum expected number of observed voxels, evaluated by $\mathbb{E}\big[F(s_n)\big]$.

Motion Planning

A sensor motion can be represented as a number of waypoints connected by edges forming a path. The selection of these waypoints can be optimized with respect to the expected information gain of a scan at a given waypoint. In a geometric approach to finding these optimal waypoints we would simply solve backwards for the sensor pose given the center of an unknown region. However this approach is not applicable because unknown space in the region of interest prevents us from predicting whether a found backward solution is usable. This makes it necessary to randomly sample sensor poses in the high dimensional sensor space and evaluating the poses for their information gain. To efficiently select waypoints to form a sensor motion from a given start pose to a goal pose in a high dimensional space we are using a PRM planner. A PRM planner is a sampling based approach, divided into two phases, a planning and a query phase.



(a) (b)

Fig. 2. Illustration of the observed environment represented as an occupancy grid (yellow unknown space, red occupied space) after a number of measurements have been taken. The scene was observed using the head mounted kinect in (a) and using the kinect placed in the manipulator in (b).

In the planning phase of the PRM we build a graph with nodes representing camera poses and edges as connections between nodes. To create the PRM we randomly sample a 6D pose $s = \mathbb{R}^6$, with $S = \{s_1, s_2, \ldots, s_n\}$ in the cartesian manipulator workspace and check their feasibility in terms of reachability and joint constraints of the manipulator using inverse kinematics. Each new pose is inserted into the graph by connecting the new node to its k-nearest neighbors, determined with a metric (euclidean distance).

In the query phase of the PRM we want to estimate a path given a start pose P_s and a goal pose P_g. The start pose P_s is the current camera frame pose and the goal pose P_g is the estimated next best view pose $\hat{s}$ hence $P_g = \hat{s}$. We want to find a path in the graph which maximizes the expected number of observed voxels.

To find the best path we have to evaluate all paths $r = \{r_1, r_2, \ldots, r_n\}$ which connect the start pose P_s and a goal pose P_g for their potential information gain. However, depending on the number of samples and edges between the nodes, the number of possible paths can be quite large, making an exhaustive breadth-first search expensive. To reduce the number of possible paths and with that the search time we constrain paths to be strictly reducing the goal distance on every node. A path is evaluated by calculating the information gain of each waypoint with respect to the previous waypoint in the path r. The best path r_b is defined as a path which has the maximum information gain combined over all waypoints in the path.

III. Results and Conclusion

We observed the same scene with a head mounted kinect as well as a manipulator equipped with a kinect sensor and compared the remaining unknown space to each other. As a result we can see in Fig. 1(a) the head mounted kinect is able to observe 87% of the scene. In comparison, the in hand sensor is able to observe 97%. We can see that in a scenario where the robot previously has been unable to sufficiently observe the scene, it now can almost fully observe the environment. This is critical in places like kitchens where movability constraints and clutter are often present. Additionally motion planning reduces the computation time because fewer NBV pose need to be found.