

# Active Exploration using Trajectory Optimization for Robotic Grasping in the Presence of Occlusions

Gregory Kahn<sup>1</sup>, Peter Sujan<sup>1</sup>, Sachin Patil<sup>1</sup>, Shaunak Bopardikar<sup>3</sup>, Julian Ryde<sup>3</sup>, Ken Goldberg<sup>2</sup>, Pieter Abbeel<sup>1</sup>

**Abstract**— We consider the task of actively exploring unstructured environments to facilitate robotic grasping of occluded objects. Typically, the geometry and locations of these objects are not known a priori. We mount an RGB-D sensor on the robot gripper to maintain a 3D voxel map of the environment during exploration. The objective is to plan the motion of the sensor in order to search for feasible grasp handles that lie within occluded regions of the map. In contrast to prior work that generates exploration trajectories by sampling, we directly optimize the exploration trajectory to find grasp handles. Since it is challenging to optimize over the discrete voxel map, we encode the uncertainty of the positions of the occluded grasp handles as a mixture of Gaussians, one per occluded region. Our trajectory optimization approach encourages exploration by penalizing a measure of the uncertainty. We then plan a collision-free trajectory for the robot arm to the detected grasp handle. We evaluated our approach by actively exploring and attempting 300 grasps. Our experiments suggest that compared to the baseline method of sampling 10 trajectories, which successfully grasped 58% of the objects, our active exploration formulation with trajectory optimization successfully grasped 93% of the objects, was 1.3× faster, and had 3.2× fewer failed grasp attempts.

## I. INTRODUCTION

Our work is motivated by the desire to facilitate autonomous grasping of objects behind clutter, inside containers, on high shelves, and other such unstructured environments. Refer to Fig. 1 for such a situation where the robot needs to grasp items commonly found in a kitchen that are occluded by a grocery bag and a box.

The presence of visual occlusions prevents commonly used sensors such as cameras and laser range finders from observing the geometry of the objects in the environment, thus impeding grasp planning and execution. The limited range of sensing and uncertainty in sensing due to noise and calibration errors pose additional challenges. Moreover, creating detailed geometric models of these objects in advance may be infeasible since cluttered and occluded environments may consist of objects that have not been previously encountered. In such situations, the robot needs to sense the world to gain more information about the objects and their surroundings before attempting to grasp them.

To facilitate exploration in this setting, we mount an RGB-D sensor on the robot gripper of a 7-DOF manipulator,

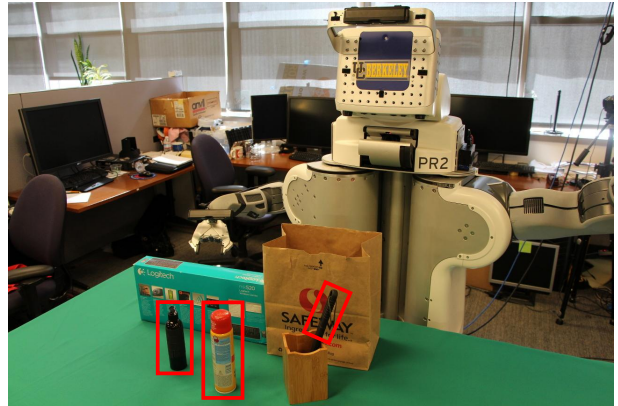


Fig. 1: **Active exploration and grasping:** Occlusions pose a key challenge for robotic grasping in unstructured environments. For instance, the soap dispenser, spray can and spatula are not visible from both the head and gripper-mounted RGB-D sensors and therefore cannot be grasped directly.

providing more degrees of freedom than the existing 2-DOF head-mounted RGB-D sensor. We use an off-the-shelf implementation of KinectFusion [28], [35] to fuse the 3D point clouds from the RGB-D sensor mounted on the gripper into a single 3D voxel map of the environment. With the recent trend of miniaturization of depth sensors [13], [14], [27], one can envision mounting these miniature sensors on the palms and wrists of end-effectors to gain a better understanding of the 3D world for reliable grasping and manipulation.

The main contributions of this work are three-fold: (i) Since reducing uncertainty by directly optimizing over the entire 3D voxel map includes superfluous regions and is computationally expensive, we encode only the uncertainty of the positions of the occluded grasp handles using a mixture of Gaussians; (ii) In contrast to the traditional approaches of using a greedy next-best-view (NBV) approach or exploration along randomly sampled trajectories, we use trajectory optimization to compute locally optimal trajectories for the robot gripper by penalizing a measure of the uncertainty to encourage exploration for grasp handles; and (iii) We validate our approach by combining our method of active exploration using trajectory optimization with off-the-shelf components to autonomously explore and grasp in an unknown environment. Fig. 3 provides an overview of our approach for active exploration for grasping in the presence of occlusions.

We evaluated our approach by actively exploring and attempting 300 grasps using an RGB-D sensor mounted on

<sup>1</sup>Department of Electrical Engineering and Computer Sciences

<sup>2</sup>Department of Industrial Engineering and Operations Research and Department of Electrical Engineering and Computer Sciences

<sup>1-2</sup> University of California, Berkeley; Berkeley, CA 94720, USA

<sup>3</sup> United Technologies Research Center Inc., Berkeley, CA, USA

the gripper of a 7-DOF PR2 manipulator. Our experiments suggest our active exploration formulation can sufficiently explore the environment to localize grasp handles and that active exploration strategies based on trajectory optimization outperform a baseline approach of randomly sampling trajectories and selecting a trajectory for execution that minimizes the objective function.

## II. RELATED WORK

Robotic grasping has been extensively studied over the past four decades [3], [34]. Most prior work on grasping assumes complete knowledge of the geometry of the target object and focuses on grasp quality metrics and grasp planning in this context. However, the geometry and properties of objects are often not known a priori when operating in unstructured environments and information about objects has to be acquired through sensors [21]. Recent work [9] has proposed heuristics to grasp unknown or unrecognized objects based on both the overall shape of the object and local features obtained from RGB-D sensor data. Machine learning methods have been used to grasp novel objects [23], [36]. Eye-in-hand systems have also been used for grasping unknown objects using visual servoing [25]. However, these methods do not consider the scenario where objects might be completely occluded from the robot’s sensors to begin with, which requires exploration of the environment to find the target objects.

A key challenge in autonomous operation of robots in unknown environments is active exploration, which involves positioning the robot’s sensors to obtain measurements that reduce uncertainty in the map of the environment. We refer the reader to an extensive survey by Chen et al. [8].

Active exploration methods can be primarily grouped into two categories: (i) frontier-based, and (ii) information-theoretic methods. Frontier-based methods [42] are primarily geometric in nature and compute frontiers as the discrete boundary between the free and unknown regions in the environment map. Extensions of this strategy have been successfully used for building maps of unknown environments in 2D [5], [16] and 3D [11], [37], [38]. Information-theoretic methods optimize an information-theoretic measure such as Shannon’s entropy or mutual information computed over the environment map [7], [15], [19], [24], [39]. However, the frontiers and information-theoretic measures are computed over a discretized 2D or 3D map of the environment. Prior work considers next best view (NBV) planning in which the robot’s sensors are positioned by sampling a set of candidate sensor poses or trajectories and selecting the optimal sensor pose or trajectory. In contrast, we propose a trajectory optimization method to explore unexplored and occluded regions and empirically show that this is computationally efficient when compared to sampling and has a higher probability of success when applied to robotic grasping applications. Trajectory optimization methods also directly incorporate kinematic constraints on the sensor motion.

Active exploration using an eye-in-hand range sensor has been used for 3D scene reconstruction [8], [24], [33]

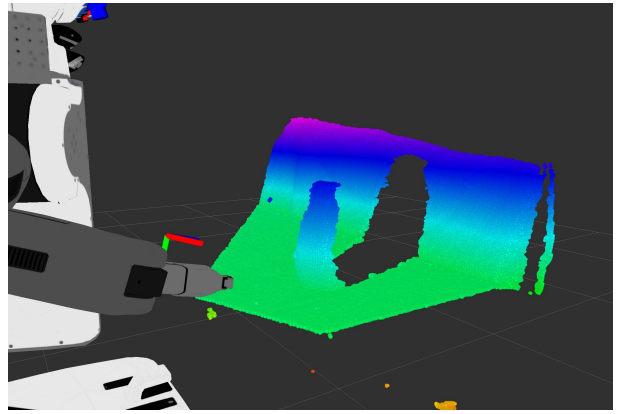


Fig. 2: **Occluded regions and graspable objects:** The current map of a single object on a tabletop surface. The occluded region (i.e. shadow) behind the object may contain graspable objects of interest.

and object detection in cluttered environments [2]. Active exploration for robotic grasping has been explored in prior work [17], [20], [26], [32]. However, these methods assume that the object geometry is known a priori, implying that a grasp can be planned for in advance, and the problem involves identifying the object in the presence of occlusions or in clutter. Methods for active exploration for grasping unknown objects [1], [4] use active exploration to reconstruct the 3D geometry of the object before planning a grasp. This is computationally expensive and might not always be feasible, e.g., when the object is in a cluttered environment or a constrained environment such as a container. In contrast, we combine our strategy for active exploration using trajectory optimization with a grasp handle identification method developed for unknown objects [40]. We note that our method can also be combined with other methods for grasping novel objects based on visual/shape features [9], [23], [36].

Our work uses recent advances in trajectory optimization under uncertainty [30] and extends the modelling of discontinuous sensor domains in 2D simulations [29] to 3D real-world experiments.

## III. SYSTEM OVERVIEW

We consider the problem of exploring and grasping in an unknown environment with an RGB-D sensor rigidly mounted to the gripper of a 7-DOF manipulator. We assume the environment is static, the robot base is fixed, and all objects are kinematically reachable and can be grasped.

We detect frontiers in the environment to enable exploration. Consider Fig. 2 in which the RGB-D sensor field-of-view is obstructed by an object, creating an occluded region. If the RGB-D sensor can be maneuvered to view the occluded region, grasp handles may be found that the gripper can then grasp. The desired behavior of the exploration is to reduce the uncertainty of these occluded regions with respect to the environment frontier.

Fig. 3 illustrates the active exploration and grasping system. We formalize our approach for active exploration using trajectory optimization in Sec. IV. We then combine our

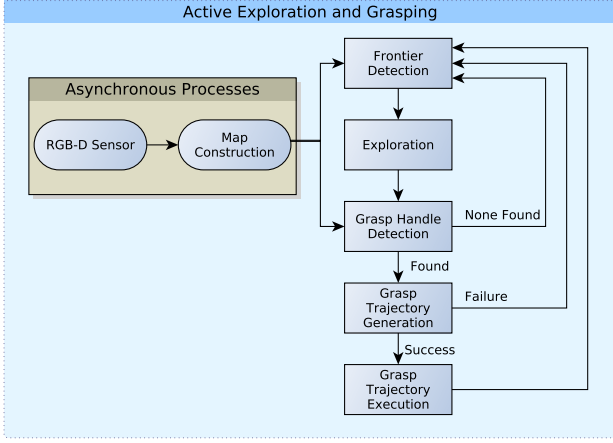


Fig. 3: **Active exploration and grasping system:** A map of the environment is continuously updated using the streaming point clouds from the RGB-D sensor. Frontiers and occluded regions are then extracted from the map. An exploration trajectory is generated, which is executed on the system. If a grasp handle is detected and a collision-free grasp trajectory is found, the system executes the grasp, releases the object in a predetermined area and resumes exploration. Otherwise, the system continues exploring.

exploration approach with off-the-shelf components for 3D map construction and grasp handle detection to form a fully autonomous active exploration and grasping system in Sec. V. Real-world experimental results are presented in Sec. VI.

#### IV. TRAJECTORY OPTIMIZATION FOR ACTIVE EXPLORATION

Given the current frontiers and occluded regions, we formulate a discrete-time trajectory optimization problem over time horizon  $T$  to localize occluded grasp handles.

##### A. Objective Formulation

**Map uncertainty parameterization:** During the exploration for occluded grasp handles, we maintain a map of the environment in a 3D voxel grid. Since reducing uncertainty by directly optimizing over the entire voxel grid includes superfluous regions and is computationally expensive, we concisely represent only the occluded grasp handle position uncertainties by parameterizing the occluded regions as a mixture of  $M$  Gaussians, where  $M$  is a user chosen parameter. Let  $(\mathbf{x}^m, \Sigma_0^m)$  be the mean position and initial covariance of the  $m$ th occluded region. The occluded region covariances  $\Sigma_t^m$  evolve over the time horizon  $T$  due to new observations. The occluded region means  $\mathbf{x}^m$  do not evolve because the environment is assumed to be static.

**Sensor state parameterization:** To represent the current state of the RGB-D sensor, we maintain the current state of the 7-DOF manipulator joints because the pose of the rigidly attached RGB-D sensor is fully determined by the kinematics of the manipulator. Note that because the RGB-D sensor state is fully determined by the system to which it is attached, our approach generalizes beyond 7-DOF manipulators to systems such as quadcopters.

We denote the current state and uncertainty of the 7-DOF manipulator as  $(\mathbf{x}_0^R, \Sigma_0^R)$  where  $\mathbf{x}_0^R$  is the current joint state

and  $\Sigma_0^R$  is the covariance. Let  $\mathbf{u}$  be the control input applied to the joint state  $\mathbf{x}^R$ .

The dynamics and measurement models for the system and RGB-D sensor are given by the stochastic, differentiable functions  $\mathbf{f}$  and  $\mathbf{h}$ :

$$\mathbf{x}_{t+1}^R = \mathbf{f}(\mathbf{x}_t^R, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim N(\mathbf{0}, I) \quad (1)$$

$$\mathbf{z}_t^m = \mathbf{h}(\mathbf{x}_t^R, \mathbf{x}^m, \mathbf{r}_t), \quad \mathbf{r}_t \sim N(\mathbf{0}, I) \quad (2)$$

where  $\mathbf{q}_t$  is the dynamics noise and  $\mathbf{r}_t$  is the measurement noise that is assumed to be drawn, without loss of generality, from a standard normal Gaussian distribution and can be scaled appropriately to be state and control dependent within the functions  $\mathbf{f}$  and  $\mathbf{h}$ , respectively.

For this work, we assume the system state  $\mathbf{x}^R$  is fully observable, effectively eliminating the stochasticity of the dynamics.

**Objective function:** We seek a set of control inputs  $\mathbf{u}_{0:T-1}$  that minimizes the uncertainty of the occluded regions and penalizes the control effort. One such formulation is to penalize the trace of the occluded region covariances and the magnitude of the control input [30], encoded in the cost functions:

$$c_t(\mathbf{x}_t^R, \Sigma_t^{1:M}, \mathbf{u}_t) = \alpha \|\mathbf{u}_t\|_2^2 + \sum_{m=1}^M \beta_t \text{tr}(\Sigma_t^m) \quad (3)$$

$$c_T(\mathbf{x}_T^R, \Sigma_T^{1:M}) = \sum_{m=1}^M \beta_T \text{tr}(\Sigma_T^m) \quad (4)$$

where  $\alpha$  and  $\beta_t$  are user-defined scalar weighting parameters.

To eliminate the stochasticity of the cost functions, we follow Platt et al. [31] and assume the maximum likelihood observation is obtained at each time step. The objective function is:

$$\min_{\mathbf{x}_{0:T}^R, \mathbf{u}_{0:T-1}} E[c_T(\mathbf{x}_T^R, \Sigma_T^{1:M}) + \sum_{t=0}^{T-1} c_t(\mathbf{x}_t^R, \Sigma_t^{1:M}, \mathbf{u}_t)] \quad (5)$$

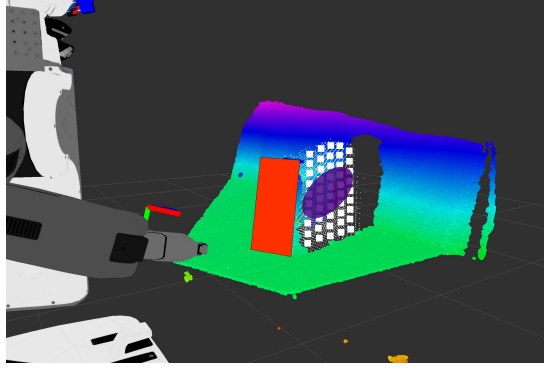
$$\begin{aligned} \text{s. t. } & \mathbf{x}_{t+1}^R = \mathbf{f}(\mathbf{x}_t^R, \mathbf{u}_t, \mathbf{0}) \\ & \mathbf{x}_t^R \in \mathcal{X}_{feasible}, \mathbf{u}_t \in \mathcal{U}_{feasible} \\ & \mathbf{x}_0^R = \mathbf{x}_{init}^R, \Sigma_0^m = \Sigma_{init}^m \end{aligned}$$

##### B. Uncertainty and System Modeling

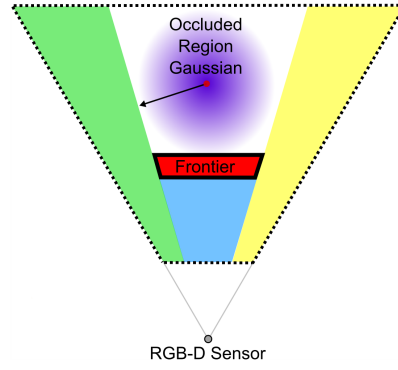
**Uncertainty model:** Given the current system belief  $(\mathbf{x}_t^R, \Sigma_t^R)$ , occluded region beliefs  $(\mathbf{x}^m, \Sigma_t^m)$ , control input  $\mathbf{u}_t$  and measurements  $\mathbf{z}_{t+1}^m$ , the occluded region beliefs evolve using an extended Kalman filter (EKF). We assume the occluded regions are independent and thus their covariances evolve separately.

Due to discontinuities in the RGB-D sensor domain stemming from environment occlusions and limited field-of-view, some of the dimensions of  $\mathbf{z}_t^m$  may not be measured. Following the approach of Patil et al. [29], we define a binary vector  $\delta_t^m \in \mathbb{R}^{\dim[\mathbf{z}]}$ , where the  $i$ th entry in the vector  $\delta_t^m$  takes the value 1 if the  $i$ th dimension of  $\mathbf{z}_t^m$  and a value of 0 if no measurement is obtained due to discontinuities in the sensing domain. Let  $\Delta_t^m = \text{diag}[\delta_t^m]$ .

For each occluded region, the EKF update equations are:



(a) Frontier detection



(b) Signed-distance computation

Fig. 4: The signed-distance computation (b) is computed with respect to the frontiers and occluded region Gaussians (a).

(a) The output of the frontier detection algorithm given a single object in the environment. The frontier is represented by the red rectangle and the voxels in the occluded region are shown in white. The purple ellipsoid is the Gaussian fit to the occluded voxels.

(b) A simplified 2D illustration of the signed-distance computation for a single occluded region Gaussian. The RGB-D sensor view frustum is outlined by the dotted line. The frontier detection algorithm has extracted the red frontier and the purple occluded region Gaussian above it. The view frustum is first truncated geometrically against the frontier, creating the green, blue and yellow convex regions. The signed-distance is negative if the occluded region mean is inside one of the convex regions and positive otherwise. The magnitude of the signed-distance is the distance to the nearest convex region border. The signed-distance is shown by the black arrow.

$$\Sigma_{t+1} = (I - K_t H_t) \Sigma_t^- \quad (6a)$$

$$K_t = \Sigma_t^- H_t^\top \Delta_t [\Delta_t H_t \Sigma_t^- H_t^\top \Delta_t + R_t R_t^\top]^{-1} \Delta_t \quad (6b)$$

$$\Sigma_t^- = A_t \Sigma_{t-1} A_t^\top + Q_t Q_t^\top \quad (6c)$$

$$A_t = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}^R}(\mathbf{x}_t^R, \mathbf{u}_t, \mathbf{0}) & 0 \\ 0 & I \end{bmatrix}, \quad Q_t = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{q}}(\mathbf{x}_t^R, \mathbf{u}_t, \mathbf{0}) & 0 \\ 0 & 0 \end{bmatrix} \quad (6d)$$

$$H_t = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{x}^R}(\mathbf{x}_{t+1}^R, \mathbf{x}^m, \mathbf{0}) & \frac{\partial \mathbf{h}}{\partial \mathbf{x}^m}(\mathbf{x}_{t+1}^R, \mathbf{x}^m, \mathbf{0}) \end{bmatrix} \quad (6e)$$

$$R_t = \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\mathbf{x}_{t+1}^R, \mathbf{x}^m, \mathbf{0}), \quad \Delta_t = \begin{bmatrix} I & 0 \\ 0 & \Delta_t^m \end{bmatrix} \quad (6f)$$

$$\Sigma_0 = \begin{bmatrix} \Sigma_0^R & 0 \\ 0 & \Sigma_0^m \end{bmatrix} \quad (6g)$$

In Eq. 6g the initial covariances are created by diagonally concatenating the initial system uncertainty with the occluded region covariance. Eq. 6c is the dynamics update for the covariances, where in Eq. 6d the matrices  $A_t$  and  $Q_t$  are modified based on the assumption that the environment is static. The Kalman gain update in Eq. 6b includes the binary matrix  $\Delta_t$  to account for discontinuities in the RGB-D sensing domain, where  $\Delta_t$  in Eq. 6f contains the identity matrix because we assume the joint state is fully observable.

**Dynamics model:** The dynamics of the 7-DOF manipulator are  $\mathbf{f}(\mathbf{x}^R, \mathbf{u}, \mathbf{q}) = \mathbf{x}^R + \mathbf{u} + \mathbf{q}$  with  $\mathbf{x}^R \in \mathbb{R}^7$  and  $\mathbf{u} \in \mathbb{R}^7$ . The joint state  $\mathbf{x}^R$  is constrained by the lower and upper joint limits such that  $\underline{\mathbf{x}}^R \leq \mathbf{x}^R \leq \bar{\mathbf{x}}^R$ .

**Observation model:** The observation function  $\mathbf{h}$  is the joint state  $\mathbf{x}^R$  of the 7-DOF manipulator and the relative position of the occluded region  $\mathbf{x}^m$  to the position of the RGB-D sensor:

$$\mathbf{h}(\mathbf{x}^R, \mathbf{x}^m, \mathbf{r}) = \begin{bmatrix} \mathbf{x}^R \\ \mathbf{x}^m - \mathbf{t} \end{bmatrix} + \mathbf{r} \quad (7)$$

$$\begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \text{forward-kinematics}(\mathbf{x}^R) \quad (8)$$

The occluded region mean  $\mathbf{x}^m$  may not be observable due to occlusions and the limited field-of-view of the RGB-D

sensor. We model this discontinuity with the binary variable  $\delta^m$  [29] because the Kalman gain in Eq. 6b should not update the occluded region covariance  $\Sigma^m$  if the occluded region mean  $\mathbf{x}^m$  is not visible. However, we approximate  $\delta^m$  with a sigmoid to ensure the objective function, which is defined by the uncertainty model, is differentiable.

Given the current frontiers  $\Pi$ , let  $\text{sd}(\mathbf{x}^R, \mathbf{x}^m, \Pi)$  be the signed-distance of the occluded region mean to the RGB-D sensor field-of-view. Let the parameter  $\alpha$  be the slope of the sigmoidal approximation and  $\rho(\mathbf{x}^m)$  be the distance in the image plane to the principal axis [22]. The sigmoidal approximation of the measurement availability  $\delta^m$  is:

$$\delta^m = \frac{1}{1 + \exp[-\alpha(\text{sd}(\mathbf{x}^R, \mathbf{x}^m, \Pi) + \rho(\mathbf{x}^m))]} \quad (9)$$

The function  $\rho$  increases the signed-distance, which encourages the optimization to center the occluded region means in the image plane.

To calculate the signed-distance of the occluded region mean to the RGB-D sensor field-of-view, we represent the field-of-view as a set of convex regions by truncating the view frustum against the frontiers  $\Pi$  (Fig. 4b). This truncation process is done geometrically and cannot be accomplished by discretizing the view frustum because we require the signed-distance function to be differentiable. The signed-distance is negative if the occluded region mean is inside one of the convex regions and positive otherwise. The magnitude of the signed-distance is the distance to the nearest convex region border.

**Trajectory Sampling and Trajectory Optimization:** Two methods of finding control inputs that minimize the objective function in Eq. 5 are trajectory sampling and trajectory optimization.

For trajectory sampling, one randomly samples  $N$  sets of control inputs  $\mathbf{u}_{0:T-1}$  and chooses the control inputs that minimize the objective function. Sampling trajectories implicitly satisfies the dynamics constraints and collision

avoidance is straightforward to implement. However, as the dimension of the sample space  $\dim[\mathbf{x}^R]T$  increases,  $N$  must increase in order to sufficiently cover the sample space. Thus trajectory sampling suffers from the curse of dimensionality.

For trajectory optimization, one can solve a nonlinear optimization problem to find a locally optimal set of controls [30]. In this work, we used sequential quadratic programming (SQP) to locally optimize the non-convex, constrained optimization problem. The innermost QP solver was generated by a numerical optimization code framework called FORCES [10]. FORCES generates code for solving QPs that is specialized for convex multistage problems such as trajectory optimization.

Trajectory optimization does not suffer from the curse of dimensionality. However, the objective function must be differentiable and calculating the gradients required by the underlying QP can be computationally expensive. Collision avoidance can be addressed by adding a cost term as in Van den Berg et al. [41], but we do not include collision avoidance in this work.

**Executing Controls and Replanning:** Given the output controls  $\mathbf{u}_{0:T-1}^*$  of either trajectory sampling or trajectory optimization, we follow the model predictive control paradigm [6] by executing a subset of the optimized controls and then replanning.

## V. ACTIVE EXPLORATION AND GRASPING

We combine our active exploration formulation with off-the-shelf components to form a fully autonomous active exploration and grasping system.

**Map Construction:** We use the Point Cloud Library [35] implementation of KinectFusion (KinFu) [28] to fuse the streaming point clouds into a single map.

KinFu represents the environment map using a truncated signed-distance function (TSDF) [28] in which each voxel contains a confidence weight  $w_i$  and the truncated signed-distance  $d_i$  to the nearest surface. Voxels with  $w_i = 0$  are unknown voxels and voxels with  $w_i > 0$  are either in free-space ( $d_i > 0$ ) or near a surface ( $d_i \approx 0$ ). Fig. 5 illustrates these different voxel regions in a simplified 2D illustration.

**Frontier Detection:** We want to extract the frontiers and occluded regions (e.g. green rectangle and blue region in Fig. 5) from the TSDF for the objective function in Eq. 5 to be fully defined.

We first extract the surface voxels and zero-weight voxels from the TSDF. Under the assumption that the graspable objects are on a tabletop surface, we find the largest plane using RANSAC and filter out voxels beneath this plane. We then extract occluding objects by finding clusters in the surface voxels and fitting an oriented bounding box (OBB) to each cluster. The frontiers are the faces of the OBBs closest to the current RGB-D sensor position. The occluded regions are parameterized by fitting Gaussians to the zero-weight voxels that are occluded by each OBB. The algorithm is outlined in Alg. 1 and an example output is shown in Fig. 4a.

---

### Algorithm 1: Extract Frontiers and Occluded Regions

---

**Input:** TSDF  
**Output:** Frontiers  $\Pi$ , Occluded regions  $(\mathbf{x}^m, \Sigma_0^m)$

- 1  $S \leftarrow$  extract surface voxels from TSDF
- 2  $Z \leftarrow$  extract zero-weight voxels from TSDF
- 3 table-plane  $\leftarrow$  extract largest plane from  $S$
- 4 filter( $S$ , table-plane)
- 5  $C \leftarrow$  find clusters in  $S$
- 6 **foreach** cluster  $c^m$  in  $C$  **do**
- 7      $OBB^m \leftarrow$  oriented bounding box of  $c^m$
- 8      $\Pi^m \leftarrow$  front face of  $OBB^m$
- 9      $occ^m \leftarrow$  points in  $Z$  occluded by  $OBB^m$
- 10     $(\mathbf{x}^m, \Sigma_0^m) \leftarrow$  fit Gaussian to  $occ^m$

---

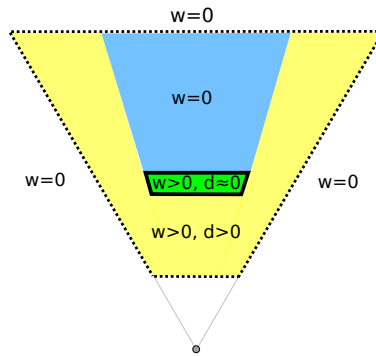


Fig. 5: **Truncated signed-distance function:** A simplified 2D illustration of the weights  $w$  and truncated signed-distance values  $d$  of the KinectFusion (KinFu) truncated signed-distance function (TSDF). The RGB-D sensor view frustum is outlined by the dotted line and the single object in the environment is shown in green. The voxels in the yellow region are known free space. The voxels in green are a known surface. The voxels in blue are an occluded region inside the view frustum. The remaining voxels are outside of the view frustum.

The simplification of the occlusions to simple geometric primitives is necessary for the observation model. The representation of the occluded regions as a Gaussian mixture is necessary to fit into the EKF framework.

**Exploration:** The exploration trajectory is either generated by sampling trajectories or trajectory optimization (Sec. IV) and is then executed on the robot.

**Grasp Handle Detection:** To generate candidate grasp handles from the KinFu point cloud, we use the ROS package handle\_detector [40], which identifies grasp handles in a point cloud by searching for regions in the point cloud that satisfy a set of geometric conditions sufficient for grasping.

**Grasp Trajectory Generation and Execution:** We attempt to plan a collision-free trajectory from the current gripper pose to each grasp handle. Given a grasp handle and the current map, we use an off-the-shelf motion planner with collision avoidance to generate a collision-free grasp trajectory. This is a different optimization than the exploration trajectory optimization detailed in Sec. IV. If a valid grasp trajectory is found, exploration is halted and the motion planner is called again to plan a post-grasp collision-free trajectory to the release point. We do not model the grasped object when planning the post-grasp trajectory, but instead add a cost to

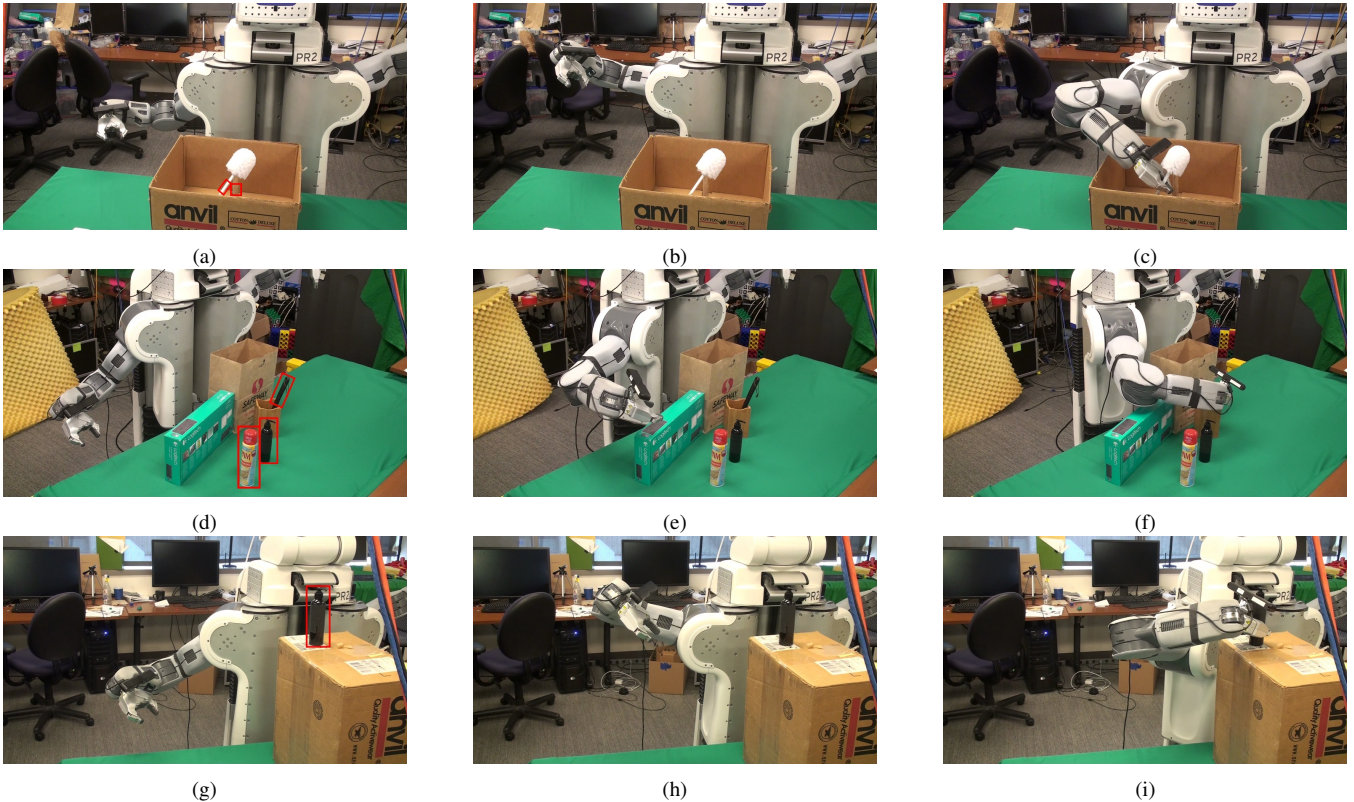


Fig. 6: **Experiment scenarios:** Experiment setups for bathroom (a)-(c), kitchen (d)-(f) and shelf (g)-(i). The bathroom scene consists of a toilet brush and a plunger in a box. The kitchen scene has a spray can, soap dispenser and spatula occluded by a grocery bag and a box. The shelf scene has a soap dispenser on a high shelf. Figures (a), (d) and (g) show initial experiment configurations and outline graspable objects in red. Figures (b), (e) and (h) show the moment a grasp handle is identified. Figures (c), (f) and (i) show successful grasps.

the optimization that maximizes the distance of the gripper to the tabletop surface.

Once the object is grasped and released, the KinFu map is reset because the PCL implementation of KinFu assumes a static environment. The active exploration and grasping process then begins anew.

## VI. EXPERIMENTS

**Hardware Setup:** The RGB-D sensor was rigidly mounted on the right 7-DOF arm of a PR2 robot. The transform of the RGB-D sensor was registered to the kinematic chain using the registered RGB-D sensor on the PR2 head and an AR marker. The RGB-D sensor used was a PrimeSense Carmine 1.09, which has a range of 0.35m-1.4m.

All code was implemented in C++ and run on an 8-core 3.4 GHz Intel i7 processor with a 1280 MB GeForce GTX 570 graphics card. Interprocess communication was through ROS.

**Experiment Scenarios:** We evaluated our active exploration and grasping system on a bathroom, kitchen and shelf scene. In all scenes, the graspable objects were positioned in reach of the 7-DOF manipulator. The initial RGB-D sensor pose was chosen so no grasp handles were visible. Between trials, the initial RGB-D sensor and object poses were perturbed to demonstrate robustness. The non-graspable occluding objects were not moved between trials. In all scenes, no grasp handle

could have been seen by the head RGB-D sensor because of occlusions or its vantage point.

*Bathroom scene:* In the bathroom scene (Fig. 6(a)-(c)), a toilet brush and plunger were placed in a box. The RGB-D sensor was positioned to be level with and pointed at the box.

*Kitchen scene:* In the kitchen scene (Fig. 6(d)-(f)), a spray can, soap dispenser, and spatula were placed behind a grocery bag and a box. The RGB-D sensor was positioned to be level with and pointed at the box.

*Shelf scene:* In the shelf scene (Fig. 6(g)-(i)), a soap dispenser was placed on a high shelf. The RGB-D sensor was positioned to look at the base of the shelf.

**Experiments:** We compared trajectory optimization with trajectory sampling for 10, 50, 100 and 200 sampled trajectories. We chose these sampling sizes because the time for trajectory optimization was comparable to sampling 100 trajectories. Both trajectory optimization and trajectory sampling had trajectory lengths of  $T = 5$ , which we chose based on the range of motion of the 7-DOF manipulator. From the occluded region Gaussians produced by the frontier detection algorithm, we input the  $M = 2$  occluded region Gaussians with the largest uncertainty for all exploration methods. We chose  $M$  based on the average number of objects across all scenes. The execution of the exploration trajectories was limited to a speed of  $5 \frac{\text{cm}}{\text{s}}$  to ensure consistency in the map construction. The execution of the grasp trajectories was

	Bathroom					Kitchen					Shelf				
	Sampling				Traj	Sampling				Traj	Sampling				Traj
	10	50	100	200	Opt	10	50	100	200	Opt	10	50	100	200	Opt
Total number of objects	20	20	20	20	20	30	30	30	30	30	10	10	10	10	10
Objects grasped	14	19	18	20	20	15	21	23	28	27	6	8	7	8	9
Objects missed	7	6	6	4	0	14	21	12	9	7	5	6	5	8	1
Avg. time per object (s)	130	133	122	141	112	159	146	139	132	115	165	129	140	131	122
	$\pm 72$	$\pm 49$	$\pm 30$	$\pm 61$	$\pm 25$	$\pm 74$	$\pm 56$	$\pm 38$	$\pm 37$	$\pm 24$	$\pm 83$	$\pm 31$	$\pm 42$	$\pm 15$	$\pm 33$
Avg. time to plan (s)	0.09	0.51	1.05	2.41	0.96	0.13	0.70	0.73	1.80	0.82	0.14	0.68	0.93	1.92	0.86
	$\pm 0.1$	$\pm 0.4$	$\pm 0.7$	$\pm 3.2$	$\pm 0.4$	$\pm 0.2$	$\pm 1.4$	$\pm 0.8$	$\pm 1.7$	$\pm 0.3$	$\pm 0.1$	$\pm 0.6$	$\pm 0.8$	$\pm 1.2$	$\pm 0.4$
Plans per grasp	4.8	4.5	4.0	4.6	3.7	6.2	5.2	4.9	4.4	4.0	6.6	4.8	5.2	4.6	4.6
	$\pm 3.0$	$\pm 2.2$	$\pm 1.5$	$\pm 2.5$	$\pm 0.8$	$\pm 3.7$	$\pm 2.8$	$\pm 1.9$	$\pm 1.6$	$\pm 1.5$	$\pm 3.7$	$\pm 1.5$	$\pm 2.0$	$\pm 1.0$	$\pm 1.4$
Avg. distance travelled (m)	3.38	3.70	3.48	3.68	3.16	3.77	3.70	3.63	3.40	3.29	3.71	3.25	3.23	3.29	3.15
Frontier detection (%)	39.3	36.3	33.5	34.8	32.7	43.8	37.8	37.0	34.2	32.6	50.9	45.0	45.4	39.2	43.7
Planning (%)	0.3	1.7	3.5	7.9	3.2	0.5	2.5	2.5	5.9	2.8	0.5	2.5	3.5	6.8	3.3
Exploration movement (%)	33.1	33.0	29.9	30.7	30.5	33.8	31.6	33.0	31.0	30.5	36.1	32.7	35.3	32.9	35.9
Grasping (%)	27.3	29.0	33.0	26.6	33.6	21.9	28.1	27.5	28.8	34.1	12.5	19.8	15.9	21.1	17.2

TABLE I: **Experiment results:** Experiment results for the bathroom, kitchen and shelf scenes comparing the exploration methods of 10, 50, 100 and 200 sampled trajectories with trajectory optimization. Ten trials were conducted for each scene and exploration method combination. Of the 300 grasp attempts, exploration with trajectory optimization grasped as many objects as exploration with 200 sampled trajectories, while trajectory optimization had the fewest grasp misses. Trajectory optimization attempted grasps the quickest and the planning time for trajectory optimization was similar to sampling 100 trajectories. Exploring with trajectory optimization required the least plans per grasp attempt and travelled the least distance per grasp attempt. The majority of the time in the active exploration and grasping system was split between frontier detection, executing exploration trajectories and grasping.

limited to a speed of  $8 \frac{\text{cm}}{\text{s}}$  for safety reasons. We ran 10 trials for each scene and exploration method. The experimental results are listed in Table I.

**Analysis:** We analyze the results in terms of the number of objects grasped, missed grasps and execution time.

**Grasps:** Across all scenes, exploration using sampled trajectories had more successful grasps as the number of sampled trajectories increased. The number of successful grasps when exploring with trajectory optimization was similar to sampling 200 trajectories. More objects were successfully grasped when exploring with trajectory optimization or a high number of sampled trajectories because these methods provided better initializations for the grasp trajectory optimization. With a poor initialization (e.g. upside down), the grasp trajectory was more likely to not reach the grasp handle position or push the object out of reach (Fig. 7). Collisions with the environment did not occur for trajectory optimization and 200 sampled trajectories because the observation function encourages distance between the RGB-D sensor and the frontiers and occluded regions.

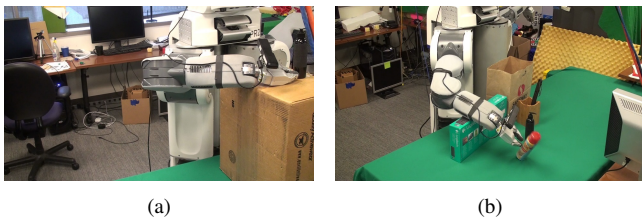


Fig. 7: **Failure cases:** Common reasons for missing a grasp: (a) inaccurate grasp handle detection and (b) unintended collisions with the grasp object.

Across all scenes, exploration with trajectory optimization had fewer misses than all explorations with sampled trajectories. Sampled trajectories often led to partial views of grasp

handles, which led to riskier grasp attempts and more misses.

**Execution time:** Across all scenes, trajectory optimization averaged the shortest time to a grasp attempt. The average time to plan an exploration trajectory with trajectory optimization was similar to the time to sample 100 trajectories. Exploration with trajectory optimization on average planned fewer times per grasp attempt and travelled the least distance per grasp attempt.

## VII. FUTURE WORK

Our approach separates the task of exploration and grasping, which may be suboptimal. Future work could consider exploration and grasping simultaneously by representing each occluded region as a Gaussian process implicit surface (GPIS) and minimizing the uncertainty of these GPISs [12].

We chose the task of finding and grasping all feasible objects in the environment. A more common scenario is to find and grasp a specific object. Future work could investigate targeted exploration and grasping by incorporating prior knowledge and using hypothesis pruning methods to select which occluded regions to explore [2], [18].

The underlying discrete map representation is abstracted away using a Gaussian mixture. We plan to investigate using gradient-based trajectory optimization over information-theoretic objectives defined directly on the discrete map representation [7], [19], [39].

## VIII. CONCLUSION

We have addressed the problem of exploring and grasping in an unknown environment. Our key contributions include (i) encoding the uncertainty of the positions of the occluded grasp handles as a mixture of Gaussians, (ii) using trajectory optimization to compute locally optimal trajectories for the

robot gripper by penalizing a measure of the uncertainty to encourage exploration for grasp handles, and (iii) combining our active exploration formulation with off-the-shelf components to autonomously explore and grasp in an unknown environment. Our experiments suggest this formulation can sufficiently explore the environment to discover grasp handles and that given equivalent time constraints, trajectory optimization outperforms trajectory sampling.

#### ACKNOWLEDGEMENTS

This research has been funded in part by AFOSR-YIP Award #FA9550-12-1-0345, by NSF under award IIS-1227536, by a DARPA Young Faculty Award #D13AP00046, and by UTRC.

#### REFERENCES

- [1] J. Aleotti, D. L. Rizzini, and S. Caselli, "Perception and Grasping of Object Parts from Active Robot Exploration," *Journal of Intelligent & Robotic Systems*, pp. 1–25, 2014.
- [2] N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. J. Pappas, and K. Daniilidis, "Hypothesis Testing Framework for Active Object Detection," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013, pp. 4216–4222.
- [3] A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," in *ICRA*, 2000, pp. 348–353.
- [4] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008, pp. 292–298.
- [5] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated Multi-Robot Exploration," *IEEE Trans. on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [6] E. F. Camacho and C. Bordons, *Model Predictive Control*. London, UK: Springer Verlag, 2004.
- [7] B. Charrow, V. Kumar, and N. Michael, "Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information," in *Robotics: Science and Systems (RSS)*, 2013.
- [8] S. Chen, Y. Li, and N. M. Kwok, "Active Vision in Robotic Systems: A Survey of Recent Developments," *Int. Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [9] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan, "Towards Reliable Grasping and Manipulation in Household Environments," in *Experimental Robotics*, 2014, pp. 241–252.
- [10] A. Domahidi, "FORCES: Fast optimization for real-time control on embedded systems," <http://forces.ethz.ch>, Oct. 2012.
- [11] C. Dornhege and A. Kleiner, "A Frontier-Void-based Approach for Autonomous Exploration in 3D," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.
- [12] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty Aware Grasping and Tactile Exploration," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013.
- [13] Engadget, "Tiny integrated 3D Depth Cameras," Jun 2013. [Online]. Available: <http://www.engadget.com/2013/06/06/pmd-infineon-camboard-pico-s-3d-depth-camera/>
- [14] Google, "Google: Project Tango," Feb 2014. [Online]. Available: <https://www.google.com/atap/projecttango/>
- [15] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active Planning for Underwater Inspection and the Benefit of Adaptivity," *Int. Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.
- [16] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies," in *ISR/ROBOTIK 2010*, 2010, pp. 1–7.
- [17] D. Holz, M. Nieuwenhuisen, D. Droschel, J. Stuckler, A. Berner, J. Li, R. Klein, and S. Behnke, "Active Recognition and Manipulation for Mobile Robot Bin Picking," in *Gearing up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe*, 2014, pp. 133–153.
- [18] S. Javdani, M. Klingensmith, J. A. Bagnell, N. Pollard, and S. Srinivasa, "Efficient Touch Based Localization through Submodularity," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013.
- [19] B. J. Julian, S. Karaman, and D. Rus, "On Mutual Information-based Control of Range Sensing Robots for Mapping Applications," *Int. Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.
- [20] L. Kaelbling and T. Lozano-Perez, "Integrated Task and Motion Planning in Belief Space," *Int. Journal of Robotics Research*, vol. 32, no. 9–10, pp. 1194–1227, 2013.
- [21] D. Katz, J. Kenney, and O. Brock, "How can Robots Succeed in Unstructured Environments?" in *RSS 2008 Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008.
- [22] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.
- [23] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, "Grasping with Application to an Autonomous Checkout Robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 2837–2844.
- [24] M. Krainin, B. Curless, and D. Fox, "Autonomous Generation of Complete 3D Object Models using Next Best View Manipulation Planning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 5031–5037.
- [25] A. Leeper, K. Hsiao, E. Chu, and J. K. Salisbury, "Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments," in *Int. Symp. on Experimental Robotics (ISER)*, 2014, pp. 253–267.
- [26] Z. W. Lim, D. Hsu, and W. S. Lee, "Adaptive Informative Path Planning in Metric Spaces," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2014.
- [27] MIT Technology Review, "Depth sensing cameras head to mobile devices," Oct 2013. [Online]. Available: <http://m.technologyreview.com/news/519546/depth-sensing-cameras-head-to-mobile-devices/>
- [28] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Int. Symp. on Mixed and Augmented Reality (ISMAR)*. IEEE, 2011, pp. 127–136.
- [29] S. Patil, Y. Duan, J. Schulman, K. Goldberg, and P. Abbeel, "Gaussian Belief Space Planning with Discontinuities in Sensing Domains," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.
- [30] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, "Scaling up Gaussian Belief Space Planning through Covariance-Free Trajectory Optimization and Automatic Differentiation," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2014.
- [31] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief Space Planning assuming Maximum Likelihood Observations," in *Robotics: Science and Systems (RSS)*, 2010.
- [32] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Efficient Planning in Non-Gaussian Belief Spaces and its Application to Robot Grasping," in *Int. Symp. on Robotics Research (ISRR)*, 2011.
- [33] C. Pothast and G. S. Sukhatme, "A Probabilistic Framework for Next Best View Estimation in a Cluttered Environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148–164, 2014.
- [34] D. Prattichizzo and J. C. Trinkle, "Grasping," *Springer Handbook of Robotics*, pp. 671–700, 2008.
- [35] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.
- [36] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *Int. Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.
- [37] R. Shade and P. Newman, "Choosing Where to Go: Complete 3D Exploration with Stereo," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 2806–2811.
- [38] S. Shen, N. Michael, and V. Kumar, "Stochastic Differential Equation-based Exploration Algorithm for Autonomous Indoor 3D Exploration with a Micro-Aerial Vehicle," *Int. Journal of Robotics Research*, vol. 31, no. 12, pp. 1431–1444, 2012.
- [39] C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-based Exploration Using Rao-Blackwellized Particle Filters," in *Robotics: Science and Systems (RSS)*, vol. 2, 2005.
- [40] A. ten Pas and R. Platt, "Localizing Handle-like Grasp Affordances in 3D Point Clouds," in *International Symposium on Experimental Robotics*, 2014.
- [41] J. van den Berg, S. Patil, and R. Alterovitz, "Motion Planning under Uncertainty using Iterative Local Optimization in Belief Space," *Int. Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [42] B. Yamauchi, "A Frontier-based Approach for Autonomous Exploration," in *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1997, pp. 146–151.