Nonmyopic View Planning for Active Object Classification and Pose Estimation

Nikolay Atanasov, *Student Member, IEEE*, Bharath Sankaran, *Student Member, IEEE*, Jerome Le Ny, *Member, IEEE*, George J. Pappas, *Fellow, IEEE*, and Kostas Daniilidis, *Fellow, IEEE*

***Abstract*—One of the central problems in computer vision is the detection of semantically important objects and the estimation of their pose. Most of the work in object detection has been based on single image processing, and its performance is limited by occlusions and ambiguity in appearance and geometry. This paper proposes an active approach to object detection in which the point of view of a mobile depth camera is controlled. When an initial static detection phase identifies an object of interest, several hypotheses are made about its class and orientation. Then, a sequence of views, which balances the amount of energy used to move the sensor with the chance of identifying the correct hypothesis, is planned. We formulate an active hypothesis testing problem, which includes sensor mobility, and solve it using a point-based approximate partially observable Markov decision process algorithm. The validity of our approach is verified through simulation and real-world experiments with the PR2 robot. The results suggest that the approach outperforms the widely used greedy viewpoint selection and provides a significant improvement over static object detection.**

***Index Terms*—Active object classification and pose estimation, hypothesis testing, motion control, planning and control for mobile sensors, recognition, robotics, vocabulary tree.**

I. Introduction

With the rapid progress of robotics research, the utility of autonomous robots is no longer restricted to controlled industrial environments. The focus has shifted to high-level interactive tasks in complex environments. The effective execution of such tasks requires the addition of semantic information to the traditional traversability representation of the environment. For example, household robots need to identify objects of interest and estimate their pose accurately in order to perform manipulation tasks. One of the central problems in computer vision, i.e., object classification and pose estimation, historically has been addressed with the assumption that the sensing device is static [1]–[3]. However, occlusions, variations in lighting, and imperfect object models in complex environments decrease the accuracy of single-view detectors. Active perception approaches circumvent these issues by utilizing appropriate sensing settings to gain more information about the scene. Research into *sensor management* [4] presents a structured approach to controlling the degrees of freedom in sensor systems in order to improve the information acquisition process. However, most of the work either assumes a simplified model for the detection process [5], [6] or avoids the problem altogether and concentrates on estimating a target's state after its detection [7], [8].

This paper is a step toward bridging the gap between the research in sensor management and the recent advances in 3-D object detection enabled by the advent of low-cost RGB-D cameras and open-source point cloud libraries [9]. Rather than placing the burden of providing perfect results on a static detector, the sensor can move to increase the confidence in its decision. We consider the following problem. A mobile sensor has access to the models of several objects of interest. Its task is to determine which, if any, of the objects are present in a cluttered scene and estimate their poses. The sensor has to balance the detection accuracy with the time spent observing the objects. The problem can be split into a static detection stage followed by a planning stage, which selects a sequence of views minimizing the mistakes made by the observer.

A preliminary version of this paper appeared at the 2013 IEEE International Conference of Robotics and Automation [10]. It included the problem formulation, theoretical development, and an initial evaluation of our approach in simulation. This version clarifies the theoretical development, provides extensive simulation results, and demonstrates the performance in real-world experiments using an Asus Xtion RGB-D sensor attached to the wrist of a PR2 robot.

The rest of this paper is organized as follows. The next section presents related approaches to active perception and summarizes our contribution. In Section III, we set up hypotheses about the class and orientation of an unknown object and formulate the active recognition problem precisely. A new approach for static detection with a depth camera is presented in Section IV. An observation model that assigns a confidence measure to the static detections and in turn to the class-pose hypotheses is

Manuscript received June 20, 2013; revised October 25, 2013 and April 14, 2014; accepted April 20, 2014. Date of publication May 29, 2014; date of current version September 30, 2014. This paper was recommended for publication by Associate Editor P. Jensfelt and Editor D. Fox upon evaluation of the reviewers' comments. This work was supported in part by TerraSwarm (one of the six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA). This work was also supported by Grant NSF-IIP-0742304, Grant NSF-OIA-1028009, Grant ARL MAST-CTA W911NF-08-2-0004, Grant ARL RCTA W911NF-10-2-0016, and Grant NSF-DGE-0966142.

N. Atanasov and G. J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: atanasov@seas.upenn.edu; pappasg@seas.upenn.edu).

B. Sankaran is with the CLMC Laboratory, University of Southern California, Los Angeles, CA 90089 USA (e-mail: bsankara@usc.edu).

J. L. Ny is with the Department of Electrical Engineering, Polytechnique Montreal, Montreal, QC H3T1J4, Canada, and also with GERAD, Montreal, QC H3T1J4, Canada (e-mail: jerome.le-ny@polymtl.ca).

K. Daniilidis is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: kostas@cis.upenn.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2014.2320795

discussed in Section V. Section VI describes how to choose a sequence of views for the sensor, which tests the hypotheses and balances sensing time with decision accuracy. Implementation details are in Section VII. Finally, Section VIII presents results from simulation and real-world experiments and compares the performance of our active approach with those of static detection and the widely used greedy view selection.

II. Related Work

Some of the earliest work in active perception is by Bajscy *et al.* [11], [12]. It is focused on improving the estimates of objects' 3-D positions by controlling the intrinsic parameters of a stereo camera. Similarly, Pito's paper [13] addresses the problem of selecting the resolution of a camera to improve surface reconstruction by maximizing information gain. Since then, many active vision works have utilized information-theoretic criteria for viewpoint and sensor parameter selection. The approaches in sensor management [4], [14] can be classified, according to sensor type, into *mobile* (sensors have dynamic states) and *stationary* (sensors have fixed states). The objective may be to identify a target and estimate its state or simply to improve the state estimate of a detected target. The targets might be mobile or stationary as well. The process of choosing sensor configurations may be *myopic* (quantifies the utility of the next configuration only) or *nonmyopic* (optimizes over a sequence of future configurations).

A large body of work assumes fixed sensor positions and focuses on choosing effective sensor parameters. For example, Sommerlade and Reid [8] control a pan-zoom-tilt camera with a fixed position to track mobile targets based on myopic minimization of conditional entropy. This simplifies the problem considerably because the tradeoff between minimizing the sensor movement energy and maximizing viewpoint informativeness is avoided. Methods that deal with active selection of views for realistic sensor models typically resort to myopic planning [15]–[17]. Denzler and Brown [18] select the focal length, pan and tilt angles, and the viewpoint of a camera on a hemisphere around an object of interest via greedy maximization of entropy. Borotschnig *et al.* [16] represent object appearance via parametric eigenspaces and use probability distributions in the eigenspace to greedily select discriminative views. Greedy approaches for active object recognition have been used in semantic mapping and localization too [19].

Golovin and Krause [20] showed that myopic planning for an adaptively submodular objective function is worse than the optimal strategy by a constant factor. Mutual information is submodular but not adaptively submodular, and as a result, the performance guarantees for greedy policies hold in the *nonadaptive setting only*, i.e., if the sequence of sensor views were selected offline and applied regardless of the observations received online. Such open-loop planning performs well only with linear Gaussian sensor models [21]. Instead of mutual information, we use a criterion that quantifies the tradeoff between the sensor movement energy expenditure and the expected cost of an incorrect object classification. The cost function is called the *value of information* [22, ch. 15] or the expected Bayesian risk [23]. Unfortunately, Krause [22, Prop. 15.1] shows that it is not adaptively submodular. Even with a fixed sensor state, a myopic strategy can perform arbitrarily worse than the optimal one in an adaptive setting [24]. In light of these observations, the goal of this paper is to devise a nonmyopic view planning (NVP) strategy and compare its performance with the myopic information-theoretic approaches. We represent the active object classification and pose estimation problem as a partially observable Markov decision process (POMDP) and use a point-based approximate algorithm [25], [26] to solve it.

The work that is closest to ours [27] uses a mobile sensor to classify stationary objects and estimate their poses. Static detection is performed using SIFT matching, and the object pose distributions are represented with Gaussian mixtures. Similar to our approach, the problem is encoded by a POMDP, but instead of an approximate nonmyopic policy, the authors resort to a myopic approach to reduce the differential entropy in the pose and class distributions. Karasev *et al.* [23] plan the path of a mobile sensor for visual search of an object in an otherwise known and static scene. The authors hypothesize about the object's pose and apply greedy maximization of the conditional entropy of the next measurement. Paletta and Pinz [28] describe a reinforcement learning approach for obtaining a sequence of views that maximally discriminate objects of various classes at different orientations. An approximate policy that maps a sequence of received measurements to a discriminative viewpoint is obtained offline.

Velez *et al.* [29] consider detecting doorways along the path of a mobile sensor traveling toward a fixed goal. The unknown state of a candidate detection is binary: "present" or "not present." Deformable part models [30] are used for static detection; stereo disparity and plane fitting are used for pose estimation. The pose estimates are not optimized during the planning stage. An entropy field is computed empirically offline for all viewpoints in the workspace and is used to nonmyopically select locations with high information gain. The planning is open loop because the object state distributions change online, but only the precomputed entropy field is used. We use a depth sensor, which validates the assumption that position estimates need not be optimized. However, the orientation estimates can be improved through active planning. Inspired by the work on hypothesis testing [24], [31], we introduce a rough discretization of the space of orientations so that the hidden object state takes on several values: one for "object not present" and the rest for "object present" with a specific orientation. Using several hypotheses necessitates closed-loop planning because the ranking of the viewpoints in terms of informativeness changes, depending on which hypothesis is more likely. In the POMDP formulation, we assume that the sensor observations are independent across different viewpoints, which is frequently violated in practice. An interesting aspect in the work of Velez *et al.* is that the correlating influence of the environment is approximated with a convex combination of a fully uncorrelated and a fully correlated sensor models. In future work, we would like to incorporate this idea in our observation model.

Another approach based on hypothesis testing is given in [32]. To disambiguate competing hypotheses, the authors myopically

select views to maximize the dissimilarity between the distributions of the expected measurements. In recent years, the active vision problem has received significant attention in the robotics community as well. Hanheide *et al.* [33] present an approach for object search and place categorization in large indoor environments. A novel probabilistic model is used to encode structural relations among objects and places (e.g., cereal boxes are often located in kitchens). An object search task is then represented by pairing the probabilistic conceptual map with the visual appearance of the object of interest. A sequence of views is planned using a POMDP abstraction with conditional entropy as the reward function [34]. Other recent work that does active visual search in a similar spirit is given in [35]. Probabilistic spatial relations and static properties of rooms are used to pose the problem as a fully observable Markov decision process (MDP). A greedy next-best-view approach is used to determine if an object is present or not at a specific location, while the MDP is used to synthesize a sequence of good locations to search. Sridharan *et al.* [36] plan visual sensing actions for scene understanding and disambiguation. Similar to our approach, a POMDP captures the tradeoff between plan reliability and execution time and enables a robot to simultaneously decide which region in the scene to focus on and what processing to perform.

Contributions: We introduce a new 3-D object detector, i.e., the viewpoint-pose tree (VP-Tree), which uses point cloud data from a depth sensor. The VP-Tree provides a *pose estimate* in addition to detecting an object's class. This is achieved via partial view matching and helps in cases when the object is partially occluded or in contact with another object. Second, we propose an NVP approach to improve the static detection and pose estimation by moving the sensor to more informative viewpoints. Our optimization criterion captures the tradeoff between gaining more certainty about the correct object class and pose and the cost of moving the sensor. This encodes the requirements of an object recognition task more precisely than the mutual information criterion.

III. Problem Formulation

A. Sensing

Consider a mobile depth sensor, which observes a *static* scene, containing unknown objects. The sensor has access to a finite database $\mathcal{D}$ of object models (see Fig. 1), and a subset $\mathcal{I}$ of them is designated as objects of interest. We assume that an object class has a single model associated with it and use the words model and class interchangeably. This is necessary because our static detector, i.e., the VP-Tree, works with instances. However, the view planning approach is independent of the static detector and can be used with class-based detectors.

The task of the sensor is to detect all objects from $\mathcal{I}$, which are present in the scene and to estimate their poses. Note that the detection is against not only known objects from the database but also clutter and background. At each time step, the sensor obtains a point cloud from the scene, splits it into separate surfaces (*segmentation*), and associates them with either new or previously seen objects (*data association*). These procedures are not the focus of this paper, but we mention how we perform them in Section VII-A. We assume that they estimate the object positions accurately.

Hypotheses are formulated about the class and orientation of an unknown object by choosing a *small* finite set of discrete orientations $\mathcal{R}(c) \subset SO(3)$[1] for each object class $c \in \mathcal{I}$. To denote the possibility that an object is not of interest, we introduce a dummy class $c_\emptyset$ and a dummy orientation $\mathcal{R}(c_\emptyset) = \{r_\emptyset\}$. The sensor needs to decide among the following hypotheses:

$H(c_\emptyset, r_\emptyset)$: the object does *not* belong to $\mathcal{I}$

$H(c, r)$: the objects class is $c \in \mathcal{I}$ with orientation $r \in \mathcal{R}(c)$.

To measure the correctness of the sensor's decisions, we introduce a cost for choosing $H(\hat{c}, \hat{r})$ when $H(c, r)$ is correct:

$$J_D(\hat{c}, \hat{r}, c, r) := \begin{cases} K(\hat{r}, r), & \hat{c} = c \\ K_+, & \hat{c} \in \mathcal{I}, c \notin \mathcal{I} \\ K_-, & \hat{c} \neq c, c \in \mathcal{I} \end{cases}$$

where K_+ and K_- are costs for making false positive and false negative mistakes, respectively, and $K(\cdot, \cdot)$ is a cost for an incorrect orientation estimate when the class is correct.

Example: Suppose that the task is to look for chairs (c_1) and tables (c_2), regardless of orientation ($K(\hat{r}, r) := 0$). The decision cost can be represented by the matrix

$J_D(\hat{c}, \hat{r}, c, r)$:

$\hat{c}$ \ c	$c_\emptyset$	c_1	c_2
$c_\emptyset$	0	K_-	K_-
c_1	K_+	0	K_-
c_2	K_+	K_-	0

In static detection, it is customary to run a chair detector first to distinguish between $c_\emptyset$ and c_1 and then a table detector to distinguish between $c_\emptyset$ and c_2. Our framework requires moving the sensor around the object to distinguish among the hypotheses, and it is necessary to process them concurrently.

B. Mobility

We are interested in choosing a sequence of views for the sensor, which has an optimal tradeoff between the energy used to move and the expected cost of incorrect decisions. Doing this with respect to all objects in the scene simultaneously results in a complex joint optimization problem. Instead, we treat the objects independently and process them sequentially, which simplifies the task to choosing a sequence of sensor poses to observe a single object. Further, we restrict the motion of the sensor to a sphere of radius ρ, centered at the location of the object. The sensor's orientation is fixed so that it points at the centroid of the object. We denote this space of sensor poses by $V(\rho)$ and refer to it as a *viewsphere*. A sensor pose $x \in V(\rho)$ is called a *viewpoint*. See Fig. 2 for an example. At a high-level planning stage, we assume that we can work with a fully actuated model of the sensor dynamics. The viewsphere is discretized into a set of viewpoints $\mathcal{X}(\rho)$, described by the nodes of a graph. The edges connect nodes that are reachable within a

[1] $SO(3)$ denotes the 3-D rotation Lie group.

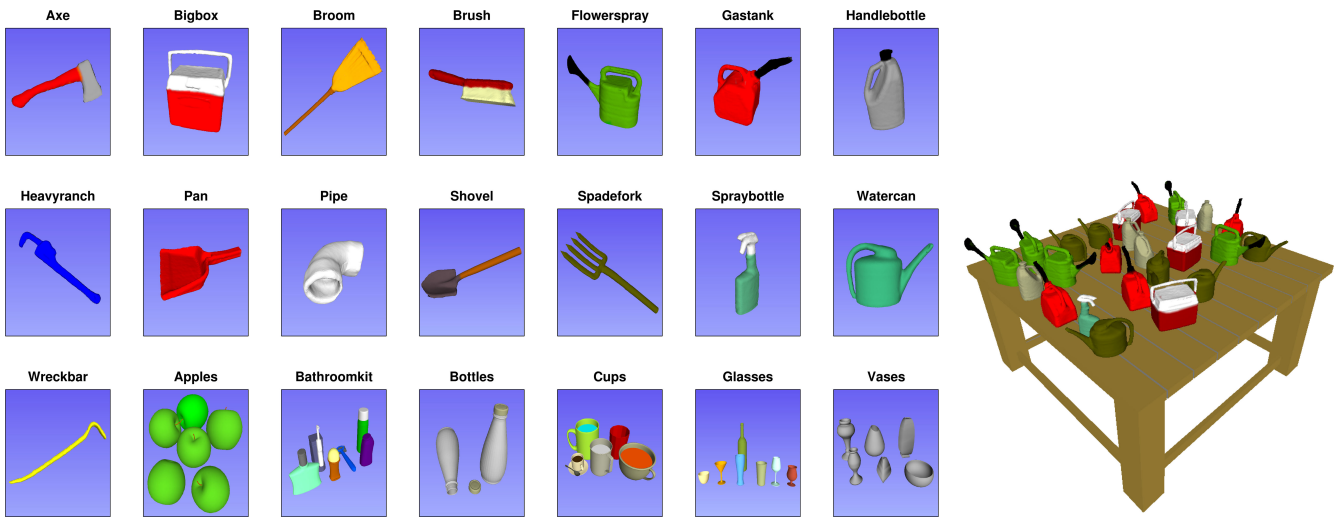

Fig. 1. (Left) Database of object models constructed using kinect fusion [9]. (Right) Example of a scene used to evaluate our framework in simulation.

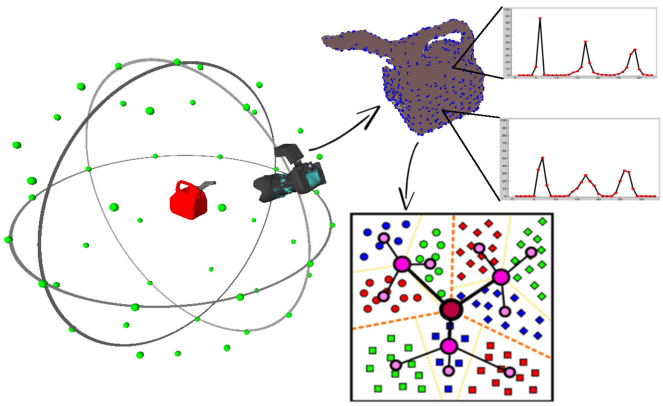

Fig. 2. Sensor position is restricted to a set of points on a sphere centered at the location of the object. Its orientation is fixed so that it points at the centroid of the object. A point cloud is obtained at each viewpoint, key points are selected, and local features are extracted (top right). The features are used to construct a VP-Tree (bottom right).

single time step from the current location based on the kinematic restrictions of the sensor. Since the motion graph is known *a priori*, the Floyd–Warshall algorithm can be used to precompute the all-pair movement cost between viewpoints:

$$g(x, x') = g_M(x, x') + g_0 = \text{ cost of moving from } x \text{ to } x' \text{ on the viewsphere } \mathcal{X}(\rho) \text{ and taking another observation}$$

where $g_0 > 0$ is a fixed measurement cost, which prevents the sensor from obtaining an infinite number of measurements without moving. As a result, a motion plan of length T for the sensor consists of a sequence of viewpoints $x_1, \ldots, x_T \in \mathcal{X}(\rho)$ on the graph, and its cost is

$$J_M(T) := \sum_{t=2}^{T} g(x_{t-1}, x_t).$$

C. Active Object Classification and Pose Estimation

Problem: Let $x_1 \in \mathcal{X}(\rho)$ be the initial pose of the mobile sensor. Given an object with unknown class c and orientation r, choose a stopping time τ, a sequence of viewpoints $x_2, \ldots, x_\tau \in \mathcal{X}(\rho)$, and a hypothesis $H(\hat{c}, \hat{r})$, which minimize the total cost:

$$\mathbb{E}\{J_M(\tau) + \lambda J_D(\hat{c}, \hat{r}, c, r)\} \quad (1)$$

where $\lambda \geq 0$ determines the relative importance of a correct decision versus cost of movement. The expectation is over the correct hypothesis and the observations collected by the sensor.

Our approach to solving the active object classification and pose estimation problem consists of two stages. First, we use the VP-Tree to perform static detection in 3-D, as described in the next section. Since the detection scores are affected by noise and occlusions, they are not used directly. Instead, the hypotheses about the detection outcome are maintained in a probabilistic framework. In the second stage, we use nonmyopic planning to select better views for the static detector and update the probabilities of the hypotheses.

IV. Static Object Detection

In this section, we introduce a novel 3-D object detector, the VP-Tree, which provides coarse pose estimates in addition to recognizing an object's class. The VP-Tree is built on the principles of the vocabulary tree, introduced by Nistér and Stewénius [37]. A vocabulary tree is primarily used for large-scale image retrieval where the number of semantic classes is in the order of a few thousand. The VP-Tree extends the utility of the vocabulary tree to joint recognition and pose estimation in 3-D by using point cloud templates extracted from views on a sphere around the models in the database $\mathcal{D}$. The templates serve to discretize the orientation of an object and make it implicit in the detection. Given a query point cloud, the best matching template carries information about both the class and the pose of the object relative to the sensor.

A simulated depth sensor is used to extract templates from a model by observing it from a discrete set $\{v_1(\rho), \ldots, v_G(\rho)\} \subset$

$V(\rho)$ of viewpoints (see Fig. 2), which need not be the same as the set of planning viewpoints $\mathcal{X}(\rho)$. Here, G stands for the number of viewpoints (48 were used in the experiments). The obtained point clouds are collected in a training set $\mathcal{T} := \{\mathcal{P}_{g,l} \mid g = 1, \ldots, G, l = 1, \ldots, |\mathcal{D}|\}$. Features, which describe the local surface curvature, are extracted for each template, as described below, and are used to train the VP-Tree. Given a query point cloud at test time, features are extracted, and the VP-Tree is used to find the template from $\mathcal{T}$, whose features match those of the query the closest.

A. Feature Extraction

It is necessary to identify a set of keypoints $\mathcal{K}_\mathcal{P}$ for each template $\mathcal{P} \in \mathcal{T}$ at which to compute local surface features. Most 3-D features are some variations of surface normal estimation and are very sensitive to noise. Using a unique keypoint estimator would be prone to errors. Instead, the keypoints are obtained by sampling the point cloud uniformly (see Fig. 2), which accounts for global appearance and reduces noise sensitivity. Neighboring points within a fixed radius of every keypoint are used to compute Fast Point Feature Histograms [38]. The features are filtered using a pass-through filter and are assembled in the set $\{f\}_{kp}$ associated with $kp \in \mathcal{K}_\mathcal{P}$.

B. Training the Viewpoint-Pose Tree

The features $\bigcup_{\mathcal{P}\in\mathcal{T}} \bigcup_{kp\in\mathcal{K}_\mathcal{P}} \{f\}_{kp}$ obtained from the training set are quantized hierarchically into visual words, which are defined by k-means clustering (see [37] for more details). Instead of performing unsupervised clustering, the initial cluster centers are associated with one feature from each of the models in $\mathcal{D}$. The training set $\mathcal{T}$ is partitioned into $|\mathcal{D}|$ groups, each consisting of the features closest to a particular cluster center. The same process is applied to each group of features, recursively defining quantization cells by splitting each cell into $|\mathcal{D}|$ new parts. The tree is determined level by level, up to a prespecified maximum number of levels.

Given a query point cloud $\mathcal{Q}$ at test time, we determine its similarity to a template $\mathcal{P}$ by comparing the paths of their features down the VP-Tree. The relevance of a feature at node i is determined by a weight $w_i := \ln(|\mathcal{T}|/\eta_i)$, where η_i is the number of templates from $\mathcal{T}$ with at least one feature path through node i. The weights are used to define a query descriptor q and a template descriptor $d_\mathcal{P}$, with ith components $q_i := n_i w_i$ and $d_i := m_i w_i$, respectively, where n_i and m_i are the number of features of the query and the template with a path through node i. The templates from $\mathcal{T}$ are ranked according to a relevance score:

$$s(q, d_\mathcal{P}) := \left\| \frac{d_\mathcal{P}}{\|d_\mathcal{P}\|_1} - \frac{q}{\|q\|_1} \right\|_1.$$

The template with the lowest relevance score is the best matching one to $\mathcal{Q}$.

C. Performance of the Viewpoint-Pose Tree

The performance of the static detector was evaluated by using the templates from $\mathcal{T}$ as queries to construct a confusion matrix (see Fig. 3). If the retrieved template matched the class of the query, it was considered correct regardless of the viewpoint. To analyze the noise sensitivity of the VP-Tree, we gradually increased the noise added to the test set. Gaussian noise with standard deviation varying from 0.05 to 5 cm on a log scale was added along the direction of the ray cast from the observer's viewpoint. The resulting class retrieval accuracy is shown in Fig. 4. As expected, the performance starts to degrade as the amount of noise is increased, but the detector behaves well at the typical depth camera noise levels.

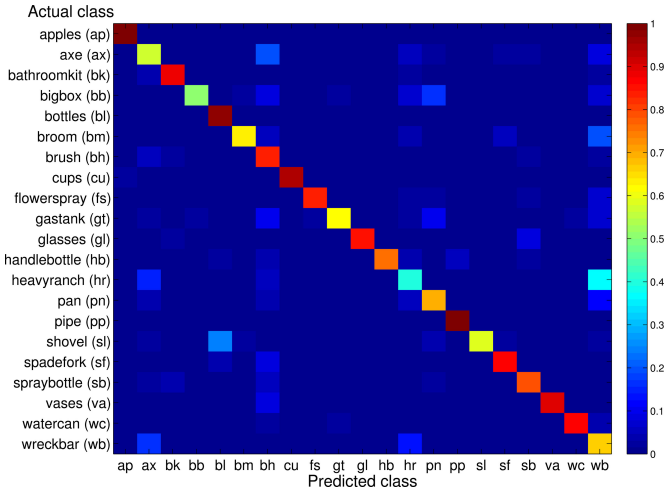

Fig. 3. Confusion matrix for all classes in the VP-Tree. A class is formed from all views associated with an object.

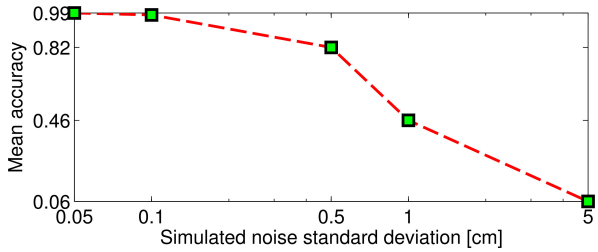

Fig. 4. Effect of signal noise on the classification accuracy of the VP-Tree.

V. Observation Model

Statistics about the operation of the VP-Tree detector for different viewpoints and object classes are needed to maintain a probability distribution over the object hypotheses. Using the VP-Tree output as the sensor observation reduces the observation space from all possible point clouds to the space of VP-Tree outputs and includes the operation of the vision algorithm in the statistics. Given a query point cloud, suppose that the VP-Tree returns template $\mathcal{P}_{g,l}$ as the top match. Assume that the templates in $\mathcal{T}$ are indexed so that those obtained from models in $\mathcal{I}$ have a lower l index than the rest. We take the linear index of $\mathcal{P}_{g,l}$ as the observation if the match is an object of interest. Otherwise, we record only the model index l, ignoring the

viewpoint g:

$$Z = \begin{cases} (l-1)G + g, & \text{if } l \leq |\mathcal{I}| \\ G|\mathcal{I}| + (l - |\mathcal{I}|), & \text{if } l > |\mathcal{I}|. \end{cases}$$

This makes the observation space 1-D.

In order to compute the likelihood of an observation offline, we introduce an occlusion state ψ for a point cloud. Suppose that the z-axis in the sensor frame measures depth and the xy plane is the image plane. Given parameters ϵ and $\mathcal{E}$, we say that a point cloud is occluded from left if it has less than $\mathcal{E}$ points in the image plane to the left of the line $x = -\epsilon$. If it has less than $\mathcal{E}$ points in the image plane above the line $y = \epsilon$, it is occluded from top. Similarly, we define occluded from bottom, occluded from right, and combinations of them (left-right, left-top, etc.). Let Ψ denote the set of occlusion states, including the nonoccluded ($\psi_\emptyset$) and the fully occluded cases. Then, the likelihood of a VP-Tree observation z for a given sensor pose $x \in \mathcal{X}(\rho)$, hypothesis $H(c, r)$, and occlusion $\psi \in \Psi$ is

$$h_z(x, c, r, \psi) := \mathbb{P}(Z = z \mid x, H(c, r), \psi).$$

The function h is called the *observation model* of the static detector. It can be obtained offline because, for a given occlusion state, it only depends on the characteristics of the sensor and the vision algorithm. Since all variables are discrete, h can be represented with a histogram, which we compute from the training set $\mathcal{T}$. Note, however, that the observation model depends on the choice of planning viewpoints and hypotheses, which means that it needs to be recomputed if they change. Ideally, it should be computed once for a given training set and then be able to handle scenarios with different sets of hypotheses and planning viewpoints.

To make the computation of the observation model independent of the choice of hypotheses and planning views, we discretize the viewsphere $V(\rho)$ very finely into a new set of viewpoints $V^o(\rho)$ with coordinates in the object frame. A nominal observation model

$$h^o_z(v, c, \psi) := \mathbb{P}(Z = z \mid v, c, \psi), \quad v \in V^o(\rho), c \in \mathcal{D}, \psi \in \Psi$$

is computed and used to obtain $h_z(x, c, r, \psi)$ as follows.

1) Determine the sensor's pose $w(x, r)$ in the object frame.
2) Find the closest viewpoint $v \in V^o(\rho)$ to $w(x, r)$ (the fine discretization avoids a large error).
3) Rotate the lines associated with ψ to the object frame of c to get the new occlusion region. Obtain a point cloud from v, remove the points within the occlusion region, and determine the occlusion state ψ^o in the object frame.
4) Copy the values from the nominal observation model:

$$h_z(x, c, r, \psi) = h^o_z(v, c, \psi^o).$$

As a result, it is necessary to compute only the nominal observation model $h^o_z(v, c, \psi^o)$. The histogram representing h^o was obtained in simulation. A viewsphere with radius $\rho = 1$ m was discretized uniformly into 128 viewpoints [the set $V^o(\rho)$]. A simulated depth sensor was used to obtain 20 independent scores from the VP-Tree for every viewpoint $v \in V^o(\rho)$, every model $c \in \mathcal{D}$, and every occlusion state $\psi \in \Psi$. Fig. 5 shows an example of the final observation model obtained from the nominal one with the planning viewpoints and hypotheses used in some of our experiments.

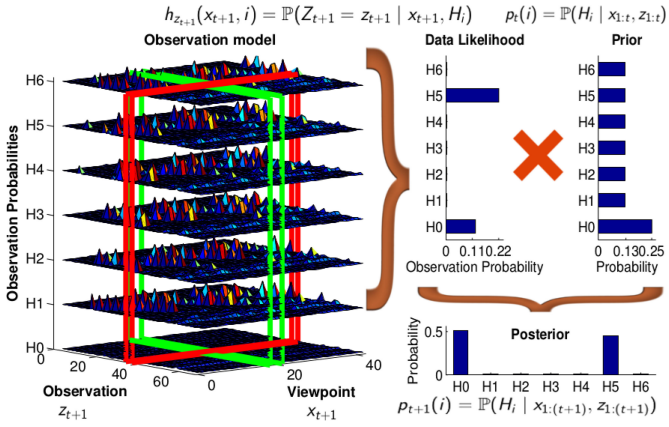

Fig. 5. Observation model obtained with seven hypotheses for the Handlebottle model and the planning viewpoints used in the simulation experiments (see Section VIII-A). Given a new VP-Tree observation z_{t+1} from the viewpoint x_{t+1}, the observation model is used to determine the data likelihood of the observation and to update the hypotheses' prior by applying Bayes rule.

VI. Active Hypothesis Testing

In this section, we provide a dynamic programming formulation for the single object optimization problem in (1). Let $\bar{\mathcal{I}} := \mathcal{I} \cup \{c_\emptyset\}$ denote the set of all hypothesized classes and $M := \sum_{c \in \bar{\mathcal{I}}} |\mathcal{R}(c)|$ the total number of hypotheses. The state of the problem at time t consists of the sensor pose $x_t \in \mathcal{X}(\rho)$ and the *information state* $p_t \in [0, 1]^M$, containing the probabilities for each hypothesis $H(c, r)$:

$$p_t(c, r) := \mathbb{P}(H(c, r) \mid x_{1:t}, z_{1:t}, \psi_{1:t}) \in [0, 1]$$

conditioned on the past sensor trajectory, $x_{1:t}$; the past VP-Tree observations, $z_{1:t}$; and the occlusion states of the observed point clouds, $\psi_{1:t}$. Suppose that the sensor decides to continue observing by moving to a new viewpoint $x_{t+1} \in \mathcal{X}(\rho)$. The newly observed point cloud is used to determine the VP-Tree score z_{t+1} and the occlusion state ψ_{t+1}. Then, the probabilities in p_t are updated according to Bayes' rule:

$$\begin{aligned} p_{t+1}(c, r) &= \mathbb{P}(H(c, r) \mid x_{1:(t+1)}, z_{1:(t+1)}, \psi_{1:(t+1)}) \\ &= \frac{\mathbb{P}(Z_{t+1} = z_{t+1} \mid x_{t+1}, H(c, r), \psi_{t+1}) p_t(c, r)}{\mathbb{P}(Z_{t+1} = z_{t+1} \mid x_{t+1}, \psi_{t+1})} \\ &= \frac{h_{z_{t+1}}(x_{t+1}, c, r, \psi_{t+1}) p_t(c, r)}{\sum_{c' \in \bar{\mathcal{I}}} \sum_{r' \in \mathcal{R}(c)} h_{z_{t+1}}(x_{t+1}, c', r', \psi_{t+1}) p_t(c', r')} \end{aligned}$$

using the observation model obtained in Section V and the assumption of independent successive observations. Fig. 5 illustrates the update. Let $T(p_t, x_{t+1}, z_{t+1}, \psi_{t+1})$ denote the Bayesian operator above, which maps p_t to p_{t+1} given a view x_{t+1}, a VP-Tree score z_{t+1}, and an occlusion state ψ_{t+1}.

The future sequence of views is planned with the assumption that there are no occlusions, i.e., $\psi_s = \psi_\emptyset$ for $s > t + 1$. This choice makes the planned sequence independent of the observed scene and allows it to be computed offline. Supposing for a

moment that the stopping time, τ, is known, the sensor is not allowed to move and has to make a decision at time τ. After the last observation z_τ has been incorporated in the posterior p_τ the terminal cost of the problem is

$$V_\tau(x_\tau, p_\tau) = \min_{\hat{c}\in\bar{\mathcal{I}}, \hat{r}\in\mathcal{R}(\hat{c})} \mathbb{E}_{c,r}\{\lambda J_D(\hat{c}, \hat{r}, c, r)\} = \min_{\hat{c}\in\bar{\mathcal{I}}, \hat{r}\in\mathcal{R}(\hat{c})} \sum_{c\in\bar{\mathcal{I}}} \sum_{r\in\mathcal{R}(c)} \lambda J_D(\hat{c}, \hat{r}, c, r) p_\tau(c, r).$$

The intermediate stage costs for $t = (\tau - 1), \ldots, 0$ are

$$V_t(x_t, p_t) = \min_{v\in\mathcal{X}(\rho)} \{g(x_t, v) + \mathbb{E}_{Z_{t+1}} V_{t+1}(v, T(p_t, v, Z_{t+1}, \psi_\emptyset))\}.$$

Letting τ be random again and t go to infinity, we get the following infinite-horizon dynamic programming equation:

$$V(x, p) = \min\left\{ \min_{\hat{c}\in\bar{\mathcal{I}}, \hat{r}\in\mathcal{R}(\hat{c})} \sum_{c\in\bar{\mathcal{I}}} \sum_{r\in\mathcal{R}(c)} \lambda J_D(\hat{c}, \hat{r}, c, r) p_\tau(c, r), \min_{v\in\mathcal{X}(\rho)} g(x, v) + \mathbb{E}_Z\{V(v, T(p, v, Z, \psi_\emptyset))\} \right\} \quad (2)$$

which is well posed by [39, Prop. 9.8 and 9.10]. Equation (2) gives an intuition about the relationship between the cost functions $g(\cdot,\cdot)$, J_D, and the stopping time τ. If at time t the expected cost of making a mistake, given by $\min_{\hat{c}\in\bar{\mathcal{I}}, \hat{r}\in\mathcal{R}(\hat{c})} \sum_{c\in\bar{\mathcal{I}}} \sum_{r\in\mathcal{R}(c)} \lambda J_D(\hat{c}, \hat{r}, c, r) p_t(c, r)$, is smaller than the cost of taking one more measurement, then the sensor stops and chooses the minimizing hypothesis; otherwise, it continues observing the scene.

To determine the value function $V(x, p)$, we resort to numerical approximation techniques, which work well when the state space of the problem is sufficiently small. Define the set $A := \{(c, r) \mid c \in \mathcal{I},\ r \in \mathcal{R}(c)\} \cup \{(c_\emptyset, r_\emptyset)\}$ of all hypothesized class-orientation pairs. Then, for $s_1, s_2 \in \mathcal{X}(\rho) \cup A$, redefine the cost of movement and the state transition function:

$$g'(s_1, p, s_2) = \begin{cases} g(s_1, s_2), & s_1, s_2 \in \mathcal{X}(\rho) \\ \displaystyle\sum_{c\in\bar{\mathcal{I}}} \sum_{r\in\mathcal{R}(c)} & \lambda J_D(c', r', c, r) p(c, r), \\ & s_1 \in \mathcal{X}(\rho), s_2 = (c', r') \in A \\ 0, & s_1 = s_2 \in A \\ \infty, & \text{otherwise} \end{cases}$$

$$T'(p, s, z, \psi_\emptyset) = \begin{cases} T(p, s, z, \psi_\emptyset), & s \in \mathcal{X}(\rho) \\ p, & s \in A. \end{cases}$$

Using the new definitions, we can rewrite (2) into the usual Bellman optimality equation for a POMDP:

$$V(s, p) = \min_{s'\in\mathcal{X}(\rho)\cup A} \{g'(s, p, s') + \mathbb{E}_Z\{V(s', T'(p, s', Z, \psi_\emptyset))\}\}.$$

The state space of the POMDP is the discrete space of sensor poses $\mathcal{X}(\rho)$ and the continuous space $\mathcal{B} := [0, 1]^M$ of distributions over the M hypotheses. Since the viewpoints are chosen locally around the object, the space $\mathcal{X}(\rho)$ is very small in practice (only 42 views were used in our experiments). The main computational challenge comes from the exponential growth of the size of $\mathcal{B}$ with the number of hypotheses M. To alleviate this difficulty, we apply a point-based POMDP algorithm [25], [26], which uses samples to compute successive approximations to the optimally reachable part of $\mathcal{B}$. The algorithm computes an approximate stationary policy $\hat{\mu} : (\mathcal{X}(\rho) \cup A) \times \mathcal{B} \to \mathcal{X}(\rho) \cup A$, which maps the current sensor viewpoint x_t and the hypotheses' probabilities p_t to a future viewpoint or a guess of the correct hypothesis. In practice, there is some control over the size of M. In most applications, the number of objects of interest is small, and we show in Section VIII-B that a very sparse discretization of the orientation space is sufficient to obtain accurate orientation estimates.

VII. Implementation Details

A. Segmentation and Data Association

Our experiments were carried out in a tabletop setting, which simplifies the problems of segmentation and data association. Point clouds obtained from the scene were clustered according to Euclidean distance by a Kd-tree. An occupancy grid representing the 2-D table surface was maintained in order to associate the clustered surfaces with new or previously seen objects. Each cell of the grid could be unoccupied or associated with the ID of an existing object. The centroid of a newly obtained object surface was projected to the table and compared with the occupied cells (if any). If the cell corresponding to the new centroid was close enough to a cell associated with an existing object, the new surface was associated with that object and its cell was indexed by the existing object's ID. Otherwise, a new object with a unique ID was instantiated. Since segmentation was not the focus of this paper, we did not explicitly address the case when objects were touching. In such situations, the detection outcome would be inconsistent and dependent on the chosen viewpoints.

B. Coupling Among Objects

The optimization in problem (1) is with respect to a single object, but while executing it, the sensor obtains surfaces from other objects within its field of view. To utilize these observations, we have the sensor turn toward the centroid of every visible object and update the probabilities of the hypotheses associated with the object. The turning is required because the observation model was trained only for a sensor facing the centroid of the object. Removing this assumption requires more training data and complicates the observation model computation. The energy used for these turns is not included in the optimization in (1).

The scores obtained from the VP-Tree are not affected significantly by scaling. This allows us to vary the radius ρ of the viewsphere in order to ease the sensor movement and to update hypotheses for other objects within the field of view. The radius is set to 1 m by default, but if the next viewpoint is not reachable, it can be adapted to accommodate for obstacles and the sensor dynamics. Algorithm 1 summarizes the complete view planning framework.

Algorithm 1 View Planning for Active Object Recognition

Input: Initial sensor pose $x_1 = (x_1^p, x_1^r) \in \mathbb{R}^3 \ltimes SO(3)$, object models of interest $\mathcal{I}$, vector of priors $p_0 \in [0,1]^M$
Output: Decision $\hat{c}^i \in \bar{\mathcal{I}}, \hat{r}^i \in \mathcal{R}(\hat{c}^i)$ for every object i in the scene
Priority queue $pq \leftarrow \emptyset$; Current object ID $i \leftarrow$ unassigned
for $t = 1$ to ∞ **do**
  Obtain a point cloud $\mathcal{Q}_t$ from x_t
  Cluster $\mathcal{Q}_t$ and update the table occupancy grid
  for every undecided object j seen in $\mathcal{Q}_t$ **do**
    Rotate the sensor so that x_t^r faces the centroid of j
    Get viewsphere radius: $\rho \leftarrow \|x_t^p - centroid(j)\|$
    Get closest viewpoint: $v^j \leftarrow \arg\min_{v \in \mathcal{X}(\rho)} \|x_t^p - v\|$
    Obtain a point cloud $\mathcal{Q}^j$
    Get VP-Tree score z^j and occlusion state ψ^j from $\mathcal{Q}^j$
    Update probabilities for object j: $p_t^j \leftarrow T(p_{t-1}^j, v^j, z^j, \psi^j)$
    if $j \notin pq$ **then**
      Insert j in pq according to probability $j \in \mathcal{I}$: $1 - p_t^j(c_\emptyset, r_\emptyset)$
  if i is unassigned **then**
    if pq is not empty **then**
      $i \leftarrow pq.pop()$
    else ▷ All objects seen so far have been processed.
      if whole scene explored **then**
        break
      else
        Move sensor to an unexplored area and start over
  $x_{t+1} \leftarrow \hat{\mu}(v^i, p_t^i)$
  if $x_{t+1} == (c, r) \in A$ **then**
    $\hat{c}^i \leftarrow c$, $\hat{r}^i \leftarrow r$, $i \leftarrow$ unassigned, Go to line 19
  Move sensor to x_{t+1}

VIII. Performance Evaluation

The VP-Tree was trained on templates extracted using a simulated depth sensor from 48 viewpoints, uniformly distributed on a viewsphere of radius $\rho = 1$ m (see Fig. 2). The observation model was trained as described in the last paragraph of Section V. Since the VP-Tree scores are not significantly affected by scaling, the score likelihoods remain similar as the viewsphere radius varies. We simplify the training process by using a fixed viewsphere radius, which limits the number of sensor poses at which we train the observation model. The reason for using a simulated sensor is also simply pragmatic. A lot of point clouds, each accompanied with a ground truth sensor pose, are needed for every model in the database in order to train the observation model. This extensive training is simpler to do in simulation but affects the recognition results in real scenes adversely.

We used $|\mathcal{X}(\rho)| = 42$ planning viewpoints in the upper hemisphere of the viewsphere to avoid placing the sensor under the table. The following costs were used in all experiments:

$$\lambda = 75, \quad J_D(\hat{c}, \hat{r}, c, r) = \begin{cases} 0, & \hat{c} = c \text{ and } \hat{r} = r \\ 1, & \text{otherwise} \end{cases}$$

$$g(x, x') = gcd(x, x') + g_0 \tag{3}$$

where $gcd(\cdot,\cdot)$ is the great-circle distance between two viewpoints $x, x' \in \mathcal{X}(\rho)$, and $g_0 = 1$ is the measurement cost. The parameter λ was set high (heuristically) in order to favor correct decisions over speed and to emphasize the advantage of active view planning over static detection. If necessary, a more principled approach to choosing λ, such as cross validation, can be used.

A. Performance Evaluation in Simulation

A single object of interest (Handlebottle) was used: $\mathcal{I} = \{c_H\}$. Keeping the pitch and roll zero, the space of object yaws was discretized into six bins to formulate the hypotheses following:

$$H(\emptyset) := H(c_\emptyset, r_\emptyset) = \text{ The object is } \textit{not} \text{ a Handlebottle}$$

$$H(r) := H(c_H, r) = \text{ The object is a Handlebottle with yaw } r \in \{0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ\}.$$

Seventy synthetic scenes were generated with ten true positives for each of the seven hypotheses. The true positive object was placed in the middle of the table, while the rest of the objects served as occluders. Fig. 1 shows an example of the scenes used in the simulation.

Four approaches for selecting sequences of views from $\mathcal{X}(\rho)$ were compared. The static approach takes a single measurement from the starting viewpoint and makes a decision based on the output from the VP-Tree. This is the traditional approach in machine perception. The second approach is our NVP. Note that the NVP policy is computed offline and used as a lookup table, which makes the planning decisions online instantaneous.

The third approach (random) is a random walk on the viewsphere, which avoids revisiting viewpoints. It ranks the viewpoints, which have yet to be visited, according to the great-circle distance from the current viewpoint. Then, it selects a viewpoint at random among the closest ones. The observation model is used to update the hypotheses' probabilities over time. The experiment is terminated when the probability of one hypothesis is above 60%, i.e., $\tau = \inf\{t \geq 0 \mid \exists (c, r) \in A \text{ such that } p_t(c, r) \geq 0.6\}$, and that hypothesis is chosen as the sensor's decision. This stopping rule was chosen empirically so that the random approach makes about the same number of measurements as NVP with the costs given in (3). This allows us to compare the informativeness of the chosen sensor views.

The widely used greedy mutual information (GMI) approach is last. Specialized to our setting, the GMI policy takes the following form:

$$\begin{aligned} &\mu_{\mathrm{GMI}}(x, p) \\ &\quad = \arg\max_{x' \in \mathcal{N}V} \frac{\mathbf{I}(H(c,r); Z)}{g(x, x')} \\ &\quad = \arg\min_{x' \in \mathcal{N}V} \frac{\mathbf{H}(H(c,r) \mid Z)}{g(x, x')} \\ &\quad = \arg\min_{x' \in \mathcal{N}V} \frac{1}{g(x, x')} \sum_{z \in \mathcal{Z}} \sum_{c \in \bar{\mathcal{I}}} \sum_{r \in \mathcal{R}(c)} p(c, r) h_z(x', c, r, \psi_\emptyset) \\ &\qquad \times \log_2 \left(\frac{\sum_{c' \in \bar{\mathcal{I}}} \sum_{r' \in \mathcal{R}(c')} p(c', r') h_z(x', c', r', \psi_\emptyset)}{p(c, r) h_z(x', c, r, \psi_\emptyset)} \right) \end{aligned}$$

where $\mathcal{N}V := \{x \in \mathcal{X}(\rho) \mid x \text{ has not been visited}\}$, $H(c, r)$ is the true hypothesis, $\mathbf{I}(\cdot;\cdot)$ is mutual information, $\mathbf{H}(\cdot \mid \cdot)$ is conditional entropy, and $\mathcal{Z}$ is the space of observations as defined in Section V. The same stopping rule as for the random approach was used so that the number of measurements made by GMI is roughly the same as those for random and NVP.

TABLE I
SIMULATION RESULTS FOR A BOTTLE DETECTION EXPERIMENT

			True Hypothesis							Avg Number of Measurements	Avg Movement Cost	Avg Decision Cost	Avg Total Cost
			H(0°)	H(60°)	H(120°)	H(180°)	H(240°)	H(300°)	H(∅)				
Predicted Hypothesis (%)	Static	H(0°)	60.35	3.86	1.00	2.19	1.48	2.19	28.92	1.00	0.00	29.74	30.74
		H(60°)	5.53	53.90	2.19	1.00	1.48	1.95	33.94	1.00	0.00	34.57	35.57
		H(120°)	4.86	4.62	51.49	3.90	2.21	1.24	31.68	1.00	0.00	36.38	37.38
		H(180°)	4.34	4.34	6.01	49.13	1.95	1.24	32.98	1.00	0.00	38.15	39.15
		H(240°)	3.88	1.96	1.24	2.20	56.11	1.24	33.37	1.00	0.00	32.92	33.92
		H(300°)	5.07	1.24	2.44	2.44	1.72	54.29	32.82	1.00	0.00	34.28	35.28
		H(∅)	0.56	1.09	3.11	1.93	0.32	3.13	89.87	1.00	0.00	7.60	8.60
												Overall Average Total Cost:	**31.52**
	Random	H(0°)	73.78	3.17	1.24	2.21	1.48	1.24	16.87	2.00	1.26	19.66	22.93
		H(60°)	1.96	70.34	2.20	1.72	1.00	1.48	21.31	2.36	1.71	22.25	26.31
		H(120°)	1.00	1.49	70.75	3.43	1.00	1.24	21.09	2.30	1.64	21.94	25.87
		H(180°)	1.48	1.73	3.66	66.97	1.97	1.48	22.71	2.71	2.16	24.78	29.64
		H(240°)	1.48	1.24	1.48	2.45	68.76	1.72	22.87	2.41	1.77	23.43	27.62
		H(300°)	1.72	1.97	1.00	1.24	1.97	71.85	20.25	2.60	2.02	21.11	25.74
		H(∅)	0.07	2.11	2.00	1.53	1.59	0.37	92.33	4.95	4.93	5.76	15.64
												Overall Average Total Cost:	**24.82**
	Greedy MI	H(0°)	82.63	2.93	0.76	1.61	0.83	0.40	10.85	1.96	1.20	13.03	16.19
		H(60°)	0.80	80.14	1.05	1.07	0.14	1.16	15.64	2.26	1.58	14.89	18.73
		H(120°)	1.09	1.05	76.93	2.64	0.83	0.82	16.66	2.30	1.64	17.31	21.25
		H(180°)	1.47	1.25	3.62	75.60	0.71	0.50	16.84	2.79	2.25	18.30	23.34
		H(240°)	0.49	1.15	0.82	2.58	75.29	1.71	17.96	2.37	1.72	18.53	22.62
		H(300°)	1.79	0.50	0.12	0.86	1.21	81.78	13.74	2.59	2.00	13.66	18.25
		H(∅)	0.72	1.35	2.23	0.39	0.25	0.41	94.65	5.29	5.37	4.01	14.67
												Overall Average Total Cost:	**19.29**
	NVP	H(0°)	87.98	0.48	0.24	0.24	0.24	0.48	10.34	2.06	1.45	9.01	12.51
		H(60°)	0.00	83.78	0.97	0.24	0.24	0.24	14.53	2.28	1.73	12.17	16.17
		H(120°)	0.48	0.00	82.81	1.21	0.00	0.00	15.50	2.37	1.86	12.89	17.12
		H(180°)	0.00	0.00	0.97	82.61	1.21	0.24	14.98	2.50	2.05	13.04	17.60
		H(240°)	0.49	0.24	0.00	0.49	78.73	0.00	20.05	2.57	2.18	15.95	20.71
		H(300°)	0.00	0.24	0.24	0.73	0.48	81.60	16.71	2.60	2.15	13.80	18.55
		H(∅)	1.49	1.58	1.37	0.37	0.74	1.25	93.20	2.08	1.50	5.10	8.68
												Overall Average Total Cost:	**15.91**

Fifty repetitions with different starting sensor poses were carried out on every scene. For each hypothesis, the measurement cost $\sum_{t=1}^{\tau} g_0$, the movement cost $\sum_{t=2}^{\tau} gcd(x_t, x_{t-1})$, and the decision cost J_D were averaged over all repetitions. The accuracy of each approach and the average costs are presented in Table I. The following conclusions can be made.

1) The active approaches for object classification and pose estimation significantly outperform the traditional single-view approach in terms of accuracy. In most cases, by making one to two extra measurements, they are able to choose the correct hypothesis more than 20% more frequently.
2) There is a steady improvement in performance when going from random viewpoint selection to greedy view planning and, finally, to NVP. Compared with the random and the GMI approaches, our NVP method needs less movement and less measurements on average and, as demonstrated by its lower average decision cost, is able to select more informative views.
3) The performance gain of NVP over GMI is not significant. In some scenarios, it might not justify the complicated offline training. For example, it is much easier to include additional constraints, such as occlusion avoidance, with greedy planning.
4) The most notable advantage of NVP comes from the adaptive stopping criterion. This is especially evident when the observed object is clutter ($H(\emptyset)$ is correct). In this case, the scores provided by the VP-Tree are not consistent and cause the probabilities of various hypotheses to increase and decrease frequently. As a result, the GMI and random approaches need many measurements to reach their prespecified stopping time. In contrast, NVP employs a longer planning horizon and recognizes that if the clutter class is likely, it is better to stop sooner than to attempt to increase the confidence as many (costly) measurements would be needed. The main advantage of NVP over GMI is not that it selects much more informative views but that it optimizes its stopping criterion.

B. Accuracy of the Orientation Estimates

Since the object orientations in a real scene are not discrete, a refinement step is needed if the algorithm detects an object of interest, i.e., decides on $\hat{c} \neq c_\emptyset$. The surfaces observed from an object are accumulated over time. After a decision, these surfaces are aligned using an iterative closest point (IPC) algorithm with the surface of the database model, corresponding to $H(\hat{c}, \hat{r})$. Thus, the final decision includes both a class and a continuous pose estimate.

Simulations were carried out to evaluate the accuracy of the continuous orientation estimates with respect to the ground truth. The following distance metric on $SO(3)$ was used to measure the error between two orientations represented by quaternions q_1 and q_2:

$$d(q_1, q_2) = \cos^{-1}\left(2\langle q_1, q_2\rangle^2 - 1\right)$$

where $\langle a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}, a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}\rangle = a_1a_2 + b_1b_2 + c_1c_2 + d_1d_2$ denotes the quaternion inner product.

A single object of interest (Watercan) was used: $\mathcal{I} = \{c_W\}$. The ground truth yaw (α) and roll (γ) of the Watercan were varied from 0° to 360° at 7.5° increments. The pitch (β) was kept at zero. Synthetic scenes were generated for each orientation.

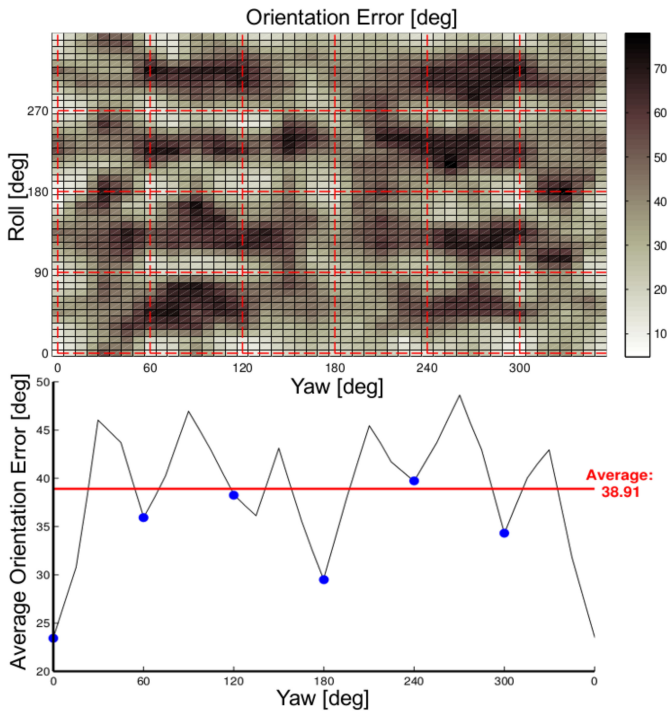

Fig. 6. Twenty-five hypotheses (red dotted lines) were used to decide on the orientation of a Watercan. The top plot shows the error in the orientation estimates as the ground truth orientation varies. The error averaged over the ground truth roll, the hypotheses over the object's yaw (blue dots), and the overall average error (red line) are shown in the bottom plot.

Hypotheses were formulated by discretizing the yaw space into six bins and the roll space into four bins:

$$H(c_\emptyset, r_\emptyset) = \text{ The object is } \textit{not} \text{ a Watercan}$$
$$H(c_W, r) = \text{ The object is a Watercan with orientation } r = (\alpha, \beta, \gamma) \in \{(i_y 60^\circ, 0, i_r 90^\circ) \mid i_y = 0, \ldots, 5 \; i_r = 0, \ldots, 3\}.$$

Fifty repetitions with different starting sensor poses were carried out on every test scene. The errors in the orientation estimates were averaged, and the results are presented in Fig. 6. As expected, the orientation estimates get worse for ground truth orientations that are further away from the hypothesized orientations. On the bottom plot, it can be seen that the hypothesized yaws correspond to local minima in the orientation error. This suggests that the number of hypotheses needs to be increased if a better orientation estimate is desired. Large errors result when the guessed hypothesis is not correct. Even when it is correct, if the real (continuous) pose of the object is far from the hypothesized one, the IPC algorithm might not perform well because it is very sensitive to the initialization. Still, a rather sparse set of hypothesized orientations was sufficient to obtain an average error of 39°. For these experiments, the average number of measurements was 2.85, and the average movement cost was 2.61.

C. Performance Evaluation in Real-World Experiments

In this section, we demonstrate that the real-world performance of NVP is similar to the simulation. We expect the same to be true for the rest of the view planning methods and did not carry out additional real experiments. It is unlikely that their behavior differs from the trends observed in Section VIII-A.

An Asus Xtion RGB-D camera attached to the right wrist of a PR2 robot was used as the mobile sensor. As before, the sensor's task was to detect if any Handlebottles ($\mathcal{I} = \{c_H\}$) are present on a cluttered table and estimate their poses. Fig. 7 shows the experimental setup. Twelve table setups were used, each containing two instances of the object of interest and eight to ten other objects. Ten repetitions were carried out for each setup, which in total corresponded to 40 true positive cases for every hypothesis. The results are summarized in Table II.

The performance obtained in the real experiments is comparable with the simulation results. On average, more movements and more measurements were required to make a decision in practice than in simulation, which can be attributed to the fact that the VP-Tree and the observation model were trained in simulation but were used to process real observations. The VP-Tree scores were inconsistent sometimes, which caused the hypotheses' probabilities to fluctuate and the sensor took longer to make decisions. Still, the results from the experiments are very satisfactory with an average accuracy of 76% for true positives and 98% for true negatives. To demonstrate that our approach can handle more complicated scenarios, several experiments were performed with two objects of interest (Handlebottle and Watercan): $\mathcal{I} = \{c_H, c_W\}$, and 53 hypotheses associated with likely poses for the two objects. See the video from Fig. 7 for more details.

The detection process demonstrated in the video takes a long time. One reason is that the cost of an incorrect decision was set very high compared with the cost of moving in order to minimize the mistakes made by the observer. As a result, the sensor takes many measurements but changing this behavior simply requires adjusting λ. There are several other aspects of our framework that slow down the processing and need improvement, however. First, the occlusion model should be used in the planning stage to avoid visiting viewpoints with limited visibility. Second, as an artifact of the way we trained the observation model, the sensor has to turn toward the centroid of every object in its field of view. This is slow and undesirable. The observation model can be modified, at the expense of a more demanding training stage, to include sensor poses that do not face the object's centroid. Finally, an unavoidable computational cost is due to the feature extraction from the observed surfaces and the point cloud registration needed to localize the sensor in the global frame as our method assumes that the sensor has accurate self-localization.

IX. Conclusion

This study has addressed the problem of classification and pose estimation of semantically important objects by actively controlling the viewpoint of a mobile depth sensor. A novel static detector, i.e., the VP-Tree, which combines detection and pose estimation in 3-D, was introduced. To alleviate the

Fig. 7. Example of the experimental setup (left), which contains two instances of the object of interest (Handlebottle). A PR2 robot with an Asus Xtion RGB-D camera attached to the right wrist (middle) employs the NVP approach for active object classification and pose estimation. In the robot's understanding of the scene (right), the object which is currently under evaluation is colored yellow. Once the system makes a decision about an object, it is colored green if it is of interest, i.e., in $\mathcal{I}$, and red otherwise. Hypothesis $H(0^\circ)$ (Handlebottle with yaw 0°) was chosen correctly for the green object. See the attached video or http://www.seas.upenn.edu/∼atanasov/vid/Atanasov_ActiveObjectDetection_TRO14.mp4 for more details.

TABLE II
RESULTS FOR A REAL-WORLD BOTTLE DETECTION EXPERIMENT

		True Hypothesis							Avg Number of Measurements	Avg Movement Cost	Avg Decision Cost	Avg Total Cost
		H(0°)	H(60°)	H(120°)	H(180°)	H(240°)	H(300°)	H(∅)				
Predicted (%)	H(0°)	87.5	2.5	5.0	0.0	0.0	0.0	5.0	2.53	2.81	9.38	14.72
	H(60°)	2.5	80.0	0.0	0.0	0.0	0.0	17.5	2.66	2.52	15.00	20.18
	H(120°)	7.5	0.0	72.5	0.0	0.0	0.0	20.0	3.16	3.43	20.63	27.22
	H(180°)	0.0	0.0	0.0	70.0	10.0	2.5	17.5	2.20	1.72	22.5	26.42
	H(240°)	0.0	0.0	0.0	2.5	75.0	2.5	20.0	2.39	2.51	18.75	23.65
	H(300°)	0.0	0.0	0.0	0.0	5.0	72.5	22.5	2.57	2.18	20.63	25.38
	H(∅)	0.0	0.0	0.97	0.0	0.0	0.97	98.05	2.17	1.93	1.46	5.56
											Overall Average Total Cost:	20.45

difficulties associated with single-view recognition, we formulated hypotheses about the class and orientation of an unknown object and proposed a soft detection strategy, in which the sensor moves to increase its confidence in the correct hypothesis. Nonmyopic planning was used to select views that balance the amount of energy spent for sensor motion with the benefit of decreasing the probability of an incorrect decision.

The validity of our approach was verified both in simulation and in real-world experiments with an RGB-D camera attached to the wrist of a PR2 robot. The performance of the NVP approach was compared with greedy view selection and with the traditional static detection. The results show that the active approaches provide a significant improvement over static detection, while the nonmyopic approach outperforms the greedy method but not significantly. The main advantage of nonmyopic planning over the greedy approach is the adaptive stopping criterion, which depends on the observations received online. Our framework has several other advantages over existing work. The idea of quantifying the likelihood of the sensor observations using a probabilistic observation model (see Section V) is general and applicable to real sensors. The proposed planning framework is independent of the static object detector and can be used with various existing algorithms in machine perception. Finally, instead of using an information-theoretic cost, the probability of an incorrect decision is minimized directly. The drawback of our approach is that it requires an accurate estimate of the sensor pose and contains no explicit mechanism to handle occlusions during the planning stage. Moreover, the sequence of views is selected with respect to a single object instead of all objects within the field of view.

Future work will focus on improving the occlusion model and using it during the planning stage. This will necessitate replanning as the motion policy will no longer be computable offline. The effect of introducing sensor dynamics in the active hypothesis testing problem is of great interest as well.

ACKNOWLEDGMENT

The authors would like to thank T. Koletchka and B. Cohen for their help with ROS and the PR2.

REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.
[3] T.-J. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-D objects using surface descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1140–1157, Nov. 1989.
[4] A. Hero, III and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors J.*, vol. 11, no. 12, pp. 3064–3075, Dec. 2011.
[5] M. Spaan, T. Veiga, and P. Lima, "Active cooperative perception in network robot systems using POMDPs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 4800–4805.
[6] K. Jenkins, "Fast adaptive sensor management for feature-based classification," Ph.D. dissertation, Dept. Elect. Eng., Boston Univ., Boston, MA, USA, 2010.

[7] C. Kreucher, K. Kastella, and A. Hero, III, "Sensor management using an active sensing approach," *Signal Process.*, vol. 85, no. 3, pp. 607–624, 2005.
[8] E. Sommerlade and I. Reid, "Information theoretic active scene exploration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–7.
[9] R. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.
[10] N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. Pappas, and K. Daniilidis, "Hypothesis testing framework for active object detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4216–4222.
[11] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 966–1005, Aug. 1988.
[12] E. Krotkov and R. Bajcsy, "Active vision for reliable ranging: Cooperating focus, stereo, and vergence," *Int. J. Comput. Vis.*, vol. 11, no. 2, pp. 187–203, 1993.
[13] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1016–1030, Oct. 1999.
[14] M. Huber, "Probabilistic framework for sensor management," Ph.D. dissertation, Fakultät für Informatik, Univ. Karlsruhe, Karlsruhe, Germany, 2009.
[15] B. Browatzki, V. Tikhanoff, G. Metta, H. Bulthoff, and C. Wallraven, "Active object recognition on a humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2021–2028.
[16] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Appearance-based active object recognition," *Image Vis. Comput.*, vol. 18, no. 9, pp. 715–727, 2000.
[17] C. Potthast and G. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 148–164, 2014.
[18] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 145–157, Feb. 2002.
[19] S. Ekvall, P. Jensfelt, and D. Kragic, "Integrating active mobile robot object recognition and SLAM in natural environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5792–5797.
[20] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *J. Artif. Intell. Res.*, vol. 42, no. 1, pp. 427–486, 2011.
[21] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–294, 2008.
[22] A. Krause, "Optimizing sensing: Theory and applications," Ph.D. dissertation, Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2008.
[23] V. Karasev, A. Chiuso, and S. Soatto, "Controlled recognition bounds for visual learning and exploration," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2915–2923, 2012.
[24] M. Naghshavar and T. Javidi, "Active sequential hypothesis testing," *Ann. Stat.*, vol. 41, no. 6, pp. 2703–2738, 2013.
[25] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robot.: Sci. Syst.*, Zurich, Switzerland, Jun. 2008. Available: http://www.roboticsproceedings.org/rss04/p9.html
[26] S. Ong, S. Png, D. Hsu, and W. Lee, "POMDPs for robotic tasks with mixed observability," in *Proc. Robot.: Sci. Syst.*, Seattle, USA, Jun. 2009. Available: http://www.roboticsproceedings.org/rss05/p26.html.
[27] R. Eidenberger and J. Scharinger, "Active perception and scene modeling by planning with probabilistic 6D object poses," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1036–1043.
[28] L. Paletta and A. Pinz, "Active object recognition by view integration and reinforcement learning," *Robot. Auton. Syst.*, vol. 31, no. 1-2, pp. 71–86, 2000.
[29] J. Velez, G. Hemann, A. Huang, I. Posner, and N. Roy, "Modelling observation correlations for active exploration and robust object detection," *J. Artif. Intell. Res.*, vol. 44, pp. 424–453, 2012.
[30] P. Felzenszwalb, R. Girshick, and D. McAllester, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[31] B. Sankaran, Sequential hypothesis testing for next best view estimation," Master's thesis, Dept. Comput. Inf. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, 2012.
[32] C. Laporte and T. Arbel, "Efficient discriminant viewpoint selection for active Bayesian recognition," *Int. J. Comput. Vis.*, vol. 68, no. 3, pp. 267–287, 2006.
[33] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 2442–2449.
[34] M. Göbelbecker, C. Gretton, and R. Dearden, "A switching planner for combined task and observation planning," in *Proc. AAAI Conf. Artif. Intell.*, 2011.
[35] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2818–2824.
[36] M. Sridharan, J. Wyatt, and R. Dearden, "Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs," *Artif. Intell.*, vol. 174, no. 11, pp. 704–725, 2010.
[37] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. Comput. Vis. Pattern Recog.*, 2006, pp. 2161–2168.
[38] R. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Inst. für Informatik, Technische Univ. München, Munich, Germany, 2009.
[39] D. Bertsekas and S. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Belmont, MA, USA: Athena Scientific, 2007.

Nikolay Atanasov (S'07) received the B.S. degree in electrical engineering from Trinity College, Hartford, CT, USA, in 2008, and the M.S. degree in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2012, where he is currently working toward the Ph.D. degree in electrical and systems engineering.

His research interests include computer vision with applications to robotics, planning and control of mobile sensors, sensor management, detection, and estimation theory.

Bharath Sankaran (S'09) received the B.E. degree in mechanical engineering from Anna University, Chennai, India, in 2006, the M.E. degree in aerospace engineering from the University of Maryland, College Park, MD, USA, in 2008, and the M.S. degree in robotics from the University of Pennsylvania, Philadelphia, PA, USA, in 2012. He is currently working toward the Ph.D. degree in computer science with the University of Southern California, Los Angeles, CA, USA.

His research interests include applying statistical learning techniques to perception and control problems in robotics, addressing perception-action loops, and biped locomotion.

Jerome Le Ny (S'05–M'09) received the B.S. degree from the École Polytechnique, Palaiseau, France, in 2001, the M.Sc. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2003, and the Ph.D. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008.

He has been an Assistant Professor with the Department of Electrical Engineering, École Polytechnique de Montréal, Montreal, QC, Canada, since May 2012. From 2008 to 2012, he was a Postdoctoral Researcher with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, USA. His research interests include robust and stochastic control and scheduling and dynamic resource allocation problems, with applications to autonomous and embedded systems, multirobot systems, and transportation systems.

George J. Pappas (S'90–M'91–SM'04–F'09) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998.

He is currently the Joseph Moore Professor and Chair of the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He also holds a Secondary Appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a Member of the GRASP Lab and the PRECISE Center. He had previously served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks.

Dr. Pappas has received various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, and the National Science Foundation PECASE. He has also received the Eliahu Jury Award for Excellence in Systems Research for his Ph.D. research.

Kostas Daniilidis (S'90–M'92–SM'04–F'12) received the undergraduate degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1986 and the Ph.D. degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 1992, under the supervision of H.-H. Nagel.

He is currently a Professor of computer and information science with the University of Pennsylvania, Philadelphia, PA, USA, where he has been with the Faculty since 1998. His research interests include visual motion and navigation, image matching, 3-D object and place recognition, and camera design.

He was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2003 to 2007. He founded the series of IEEE Workshops on Omnidirectional Vision. In June 2006, he Cochaired, with Pollefeys, the Third Symposium on 3-D Data Processing, Visualization, and Transmission, and he was the Program Chair of the 11th European Conference on Computer Vision in 2010. He was the Director of the Interdisciplinary GRASP Laboratory from 2008 to 2013, and he has been serving as the Associate Dean for Graduate Education of Penn Engineering since 2013.