An Object-Based World Model for Change Detection and Semantic Querying

Julian Mason and Bhaskara Marthi

I. Introduction

Recent years have seen an interest in robots building models of their environment at a higher level of abstraction than traditional 2D or 3D occupancy grids. Occupancy grids support low-level operations like navigation, collision avoidance, and localization. In contrast, higher level models support semantic reasoning about the objects in the world and their properties. Such models are useful for any application that involves dealing with objects, including grasping, object search, and human-robot interaction.

We describe a system that builds higher level models of the world using a mobile robot equipped with a Kinect RGB-D sensor. Our representation is object-based, and makes few assumptions about structure in the environment or the quality of perceptual primitives available. The models produced by our system support a variety of applications and scale to large environments over long periods of time. We explore two such applications: semantic querying and change detection. We demonstrate our applications on a large dataset consisting of Kinect data over roughly 50 autonomous runs of our robot during a one-month period across a $1600m^2$ office space.

II. Representation

A goal of this project has been to build a system that can operate in an unconstrained, uninstrumented home or office environment, while making realistic assumptions about what can be provided by perception algorithms now or in the next few years. This means that we cannot assume reliable segmentation or classification of objects. Indeed, many objects will be of classes never seen before. Nevertheless we would like to extract as much useful information as possible from sensor data. Our only major perceptual assumption is that the world contains horizontal planar surfaces, on which objects can be found. Therefore, our ontology consists of (horizontal) planes and objects. Planes have a height and a convex bounding polygon. Objects have a pose, a (colorized) point cloud, one or more RGB camera images from the time of their segmentation, and various attributes extracted from these sensor data, including dominant color, size, and approximate shape. All objects and planes are represented in a fixed global coordinate frame, provided by the robot's localization system.

In addition to storing objects, we would like to make temporal queries about how the world has changed over time. Doing this requires data association between data collection runs. Rather than make runtime decisions about data association, our system stores a snapshot of the world for each run, allowing a variety of data association algorithms to be dropped in. We demonstrate one such algorithm based on spatial proximity; representative results can be seen in Figure 1.

(a) The first detection of this object.

(b) A second detection under different lighting conditions, roughly a day later.

(c) A third detection, from a different point of view, roughly two days after (a).

(d) The final detection in our data, roughly five days after (a).

Fig. 1: Examples of a correctly associated object over a five-day period. The bounding box for the point cloud of the detected object is shown in green. Figures (a) and (d) show the first and last detections of this object in our data (it was removed after the data for (d) was collected). Over this period, we encountered this object ten times, and suffered only one false negative. *This figure is best viewed in color.*

III. System

Our assumption that planar surfaces support "interesting" objects forms the basis of our perceptual pipeline. Our mobile platform mounts a Kinect roughly 1.5 meters above the floor; this allows it to look down onto tables and counters. These horizontal surfaces are then extracted using RANSAC, and points above the planes are extracted and clustered into distinct objects. We use a heuristic to discard incorrect candidate objects (usually parts of walls) based on their similarity to vertical planes. Objects are associated to objects

Julian Mason is with the Duke University Department of Computer Science. 308 Research Drive, Durham, NC 27708. mac@cs.duke.edu.

Bhaskara Marthi is with Willow Garage. 68 Willow Road, Menlo Park, CA 94025. bhaskara@willowgarage.com.

(and planes to planes) within a single run by checking for overlap of their 2D convex hulls. The output of the perception pipeline is a set of objects, with their associated perceptual data and attributes, and a set of planes, with their plane equations and associated perceptual data. A screenshot of our system running can be seen in Figure 2.

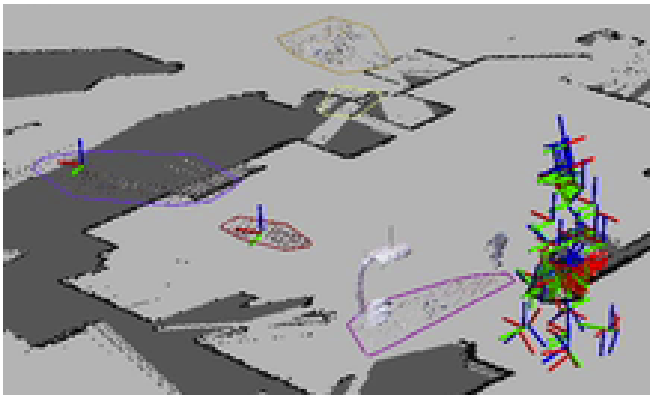

Fig. 2: Our system running. The robot is visible as a collection of coordinate frames. The colored polygons are the convex hulls of segmented planes. Immediately to the robot's left is the pointcloud of a segmented object (in this case, a white gooseneck table lamp). *This figure is best viewed in color.*

IV. Applications

Given a database of the above form, we can now pose semantic queries based on the attributes. An example query is "List all red, cylindrical objects near the robot's current location". As object metadata (including color and shape) is stored in a database, we can perform such queries efficiently. Although our current implementation is limited to queries, such attribute-based search could be part of an interactive interface, in which a human user describes an object, and the system returns a set of candidate objects (with their associated metadata). A selected object would then be presented to the user with a variety of possible robot actions, including "bring this object to me" or "check to see if this object is still there."

A second application is *change detection*. Given the database generated from a data collection run, the robot is instructed to re-observe each object in turn, and to report on its presence or absence. Run once, this allows an object inventory to be kept up-to-date; run repeatedly, this permits the movement of objects over time to be tracked, and their "behavior" inferred. Figure 1 presents a basic example: the object was consistently detected for several days, and then never again, because it was removed.

V. Dataset

To validate our system in a real environment, we have gathered a dataset of localized sensor data from passes over a standard indoor office environment[1], using a PR2 robot with a head-mounted Kinect. These were divided into two categories: "passive" collections and "rescan" collections. In a passive collection, the robot was given a set of waypoints, and navigated to each in turn. Although the waypoints remained fixed over the course of the entire experiment, the dynamic nature of the environment (and somewhat unpredictable nature of navigational planners) led to a variety of different robot trajectories. In a rescan collection, the robot was given a database generated from a previous passive collection; each object in this database was then used as a navigational goal. Specifically, the robot navigated to the object's location and observed it directly.

Data were usually collected three times a day; a passive collection in the morning and evening, and a rescan collection in the early afternoon. Robot availability (and other disturbances inherent to working in a heavily-trafficked indoor environment) kept us from maintaining a flawless schedule; nevertheless, we have RGB-D of our environment, spanning many times of day, and spread over a month.

[1]Data collection is ongoing as of this submission. Fifty one runs have been performed so far, and we expect to have around a hundred runs by the end of September 2011.