Automated System for Semantic Object Labeling With Soft-Object Recognition and Dynamic Programming Segmentation

Jonas Cleveland, Dinesh Thakur, Philip Dames, Cody Phillips, Terry Kientz, Kostas Daniilidis, *Fellow, IEEE*, John Bergstrom, and Vijay Kumar

***Abstract*— This paper presents an automated robotic system for generating semantic maps of inventory in retail environments. In retail settings, semantic maps are labeled maps of stores where each discrete section of shelving is assigned a department label describing the types of products on that shelf. Starting from a metric map of the store, the robot autonomously extracts the shelf boundaries, generates a distance-optimal tour of the store to view every shelf, and follows the tour while avoiding unmapped clutter and moving people. The robot creates a point cloud of the store using the data collected from this tour. We introduce a novel soft-object assignment algorithm to create a virtual map and a dynamic programming algorithm to segment this map. These algorithms use *a priori* information about the products to boost data from laser and camera sensors in order to recognize and semantically label objects. The primary contribution of this paper is the integration of multiple systems for automated path planning, navigation, object recognition, and semantic mapping. This paper represents an important contribution toward deploying mobile robots in dynamic human environments.**

***Note to Practitioners*—One of the critical tasks in retail is to optimally manage the use of floor space within each store. Doing this correctly requires having accurate knowledge of the way in which space is currently used in the store, how this usage changes over time, and how this usage relates to sales. In a retail chain, such as Walgreens, which has over 8000 stores in the United States, this knowledge is difficult and expensive to obtain. Furthermore, each individual retail store may have dozens of different departments and may stock thousands of unique product types. While embedded-system and infrastructure-based solutions, such as radio-frequency identification tags, are technically straightforward, they are simply not scalable over a full product inventory. We present an autonomous system that uses computer vision to recognize products and departments and that is able to autonomously navigate around clutter and moving people. The only step that requires human input is manually driving the robot around the store to create an initial map of the shelf locations. To the best of our knowledge, this is the first implementation of a fully automated robotic inventory labeling system for a retail environment. The framework presented in this paper can also be used in other retail environments and in other indoor environments with organized shelves, such as business storage facilities and hospital pharmacies.**

***Index Terms*— Automation, robot vision systems, robots.**

Manuscript received December 20, 2015; revised May 27, 2016; accepted August 9, 2016. Date of publication December 22, 2016; date of current version April 5, 2017. This paper was recommended for publication by Editor John Wen upon evaluation of the reviewers' comments. This work was supported in part by Walgreens and in part by NSF under Grant I/UCRC 1439681.

J. Cleveland, D. Thakur, C. Phillips, T. Kientz, K. Daniilidis, and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: jclev@seas.upenn.edu; tdinesh@seas.upenn.edu; pdames@seas.upenn.edu; codyp@seas.upenn.edu; tkientz@seas.upenn.edu; kostas@seas.upenn.edu; kumar@seas.upenn.edu).

P. Dames is with Temple University, Philadelphia, PA 19122 USA (e-mail: pdames@temple.edu).

J. Bergstrom is with Walgreens, Deerfield, IL 60015 USA (e-mail: john.bergstrom@walgreens.com).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The Supplementary Material contains a video of the robot navigating in our model store and in a full retail store. The video also shows the map generation, shelf extraction, path planning, and autonomous navigation processes in a large, real-world retail store. This material is 80 MB in size.

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/TASE.2016.2631085

I. Introduction

OPTIMALLY managing retail space to maximize profits while providing customers with a good experience is a challenging task that requires having detailed knowledge of the way floor space is used. Maintaining this information is difficult within a single store, which may have dozens of product departments and thousands of unique products. The challenge is even greater in a large retail chain, such as Walgreens, which has over 8000 stores in the United States. At present, the staff of each store is expected to label a map of the shelves in a store with: 1) the departments (e.g., diapers, first aid, and deodorant) contained within the shelving fixtures and 2) the linear space occupied by each department. Experience has shown that these maps can contain errors at the time of their creation, and that additional errors are introduced while revising the store maps due to seasonally fluctuating demand and the introduction and removal of products. The creation of a novel automated system that accurately determines department size and location produces significant benefits by freeing the staff to provide more customer care, reducing the costs associated with imperfect knowledge, and enabling accurate optimization of store space allocations.

The material presented in this paper has been submitted to the U.S. Patent Office [1], [2].

II. Technical Background

There is extensive prior work on using robots to create semantic maps of human environments [3]–[8]. This corpus broadly defines a semantic map as the association of semantic information with a spatial location. Much of this previous work focuses on mapping household and academic building environments. These are important human working environments,

but the characteristics of these environments provide innate advantages for object recognition systems.

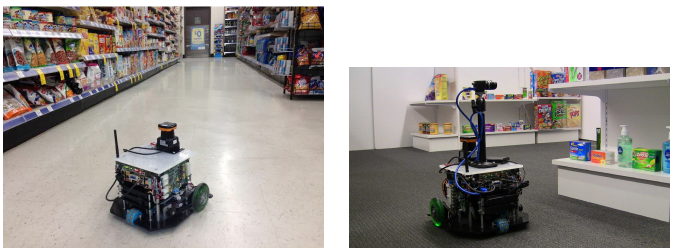
1) Typically, objects in a home environment differ drastically in size, color, and shape characteristics [4], [5], [7].
2) The number of distinct object classes is relatively small. For instance, Choudhary *et al.* [3] evaluate their system over a bicycle helmet, chair, and kitchen appliances.
3) Despite these environments being described as cluttered [4], [5], [7], there is still usually a clear line of sight to at least two faces on each object, or the objects are placed in front of a clean background so that there is little visual clutter.

Inventory storage facilities are important working environments that do not share these characteristics. Objects are placed adjacent to each other and are usually in packaging boxes so 3-D features are not discriminative. Objects of the same family are placed near one another and are often similar in size and color characteristics, making object classification difficult. Furthermore, most retail environments have thousands of unique product types.

Despite the recent significant interest in using robotic platforms in human domestic working environments, there is a surprisingly small amount of work in robotics in inventory storage environments. Mankodiya *et al.* [9] create a detailed proposal for robots that automate the construction of planogram maps and handle other retail-centric tasks, such as merchandise management, visual merchandising, and inventory management. Mankodiya *et al.* [10], Kumar *et al.* [11], and Frontoni *et al.* [12] describe systems for mapping the physical structure of a retail environment. While previous work broadly references robotic technology for object recognition and mapping, it does not describe in detail the algorithms necessary for multiclass object recognition or automated map creation.

We present an automated robotic semantic labeling system that tackles some of the problems unique to retail environments. We propose a novel formulation of the path-planning problem that guarantees complete traversal of the space while constraining the motion of the robot to ensure that the sensors view all relevant areas of the environment. Using existing navigation algorithms (Section IV-B), the robot is able to autonomously avoid unmapped static clutter, such as empty boxes or shopping carts, and moving people. Once the robot detects potential objects in the map, our soft-object recognition system (Section IV-C) identifies potential class labels from a template library. Our system uses these candidate single object labels and their likelihoods to classify larger physical-contextual regions (Section IV-E). The system extracts regions by segmenting the most likely object labels, which are each preassigned some region family. The segmentation step then updates the object-class likelihoods based on the labels of nearby objects. This soft-labeling differs from most existing systems and significantly improves the labeling accuracy of the final semantic map.

This paper combines computer vision and navigation algorithms to autonomously navigate and recognize objects in a spatially and visually cluttered environment. Our system yields an accurate annotated map and does not require any external infrastructure or additional structure in the retail environment [9], [10]. This paper is an important contribution in that we describe the implementation of an automated object discovery, map management, and path-planning system capable of semantically labeling a retail environment. This paper builds upon our previous work [12], including extended descriptions of the shelf extraction and template extraction algorithms, additional evaluation of the planning and navigation algorithms in multiple stores, and a video[1] showing the system in a real-world retail store.



(a) (b)

Fig. 1. Scarab robot in different stores. (a) Walgreens retail store. (b) Model store.

III. Infrastructure

In this section, we outline the salient features of a Walgreens retail environment and describe the mock store that we have built within the laboratory at the University of Pennsylvania. We also describe the robotic platform used in the experiments. Our system is a prototype for robotic inventory management within a Walgreens store. The installation of the system requires a human operator to teleoperate the robot in order to generate an occupancy grid map of the store. After this initial setup, the system is completely automated. Using the initial map, the robot autonomously plans a path through the store and follows this path to collect vision data about the products in the store. The robot uploads this data to a docking station, which processes the sensor data collected from the robot.

A. Walgreens Retail Environments

As described in Section II, retail stores such as the one shown in Fig. 1(a) are spatially and visually cluttered indoor environments. They often contain a series of parallel shelving units, at static locations, that hold the products. These shelving units must be at least five feet apart to allow customers to pass by one another and are typically capped by an end stand. Each shelving unit has multiple shelves arranged vertically. The store may also contain temporary displays and other transient objects, e.g., boxes, shopping carts, or baskets. These transient objects are often static during the course of a single mapping run but may move, appear, or disappear between runs.

The products are organized into departments, which are sections of shelves of standard widths that contain semantically similar products, e.g., different brands of cereal. In most stores, all of the products within a vertical column of shelves all belong to the same department. This means that only a single

[1] Available at https://www.youtube.com/watch?v=a7TDgwQ5FlM

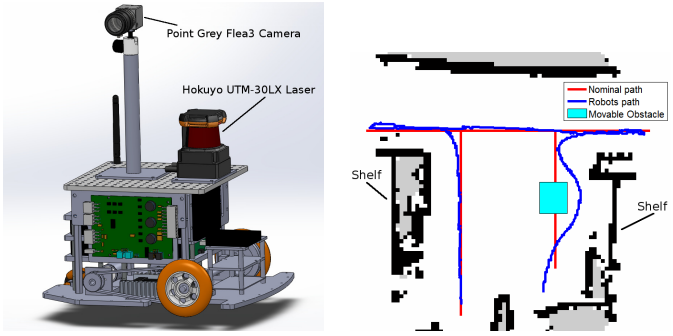


Fig. 2. (a) Computer-aided drawing drawing of the Scarab robot. (b) Example path of the robot avoiding an obstacle.

shelf is required in order to determine the department labels for the entire column of products. However, certain products may be found within multiple departments, e.g., cold medicine may be found within the Medicine department as well as a Seasonal Flu and Cold department.

We worked directly with Walgreens to design, build, and stock a model retail environment according to the company standards in order to perform realistic tests within a laboratory environment. Fig. 1(b) shows the robot in the model store. Each aisle is at least five feet long with shelves on either side with enough depth for multiple items. Our model store contains six departments with similar dimensions and product makeups to a Walgreens store. There are at least two departments on both sides of the aisle and over 60 different product brands that are found at Walgreens stores. The model shelving units are two shelves high, since only a single shelf is necessary to label a department.

In order to thoroughly test the performance of our system in a retail environment, Walgreens proposed a number of other constraints on the environment. First, at least one shelf section of longer than two feet is covered by glass to simulate products protected by a glass encasement, such as refrigerated products and electronics. Second, a movable object with a footprint of one square foot may appear in the environment to emulate a box of products waiting to be put on the shelves or a customer's shopping basket. Finally, at least two departments should have at least one product in common. In our model store, the Seasonal Flu section shares three product types with the Medicine department and one product type with the Skin Care department.

B. Robot

The platform is a modified Scarab robot [13], which is built in-house and shown in Fig. 2(a). It is a differential drive robot with a top speed of 1.4 m/s. It has a modular design with plug-and-play capability, where sensors and actuators can be easily swapped. We use a Hokuyo UTM-30LX laser and a Point Grey Flea3 USB camera for this application. Both sensors have USB3 data links. We mount the camera 46 cm from the ground plane to be able to detect products on the bottom two shelves, increasing the robustness of the system. Processing for the navigation system is done using the onboard computer with a 2.4-GHz Intel Core i5 processor and 4 GB of RAM. The robot is powered by a pair of hot swappable 14.4 V, 95-Wh LiPo batteries. The robot can also be directly plugged into the wall to charge.

C. Docking Station and Processing Computer

Data from the robot are transferred to the docking station at the end of its automated run. The docking station parses video from the camera into time-stamped frames and synchronizes the images with the position data. The docking station computer has a 2.8-GHz Intel Core i7 processor with 16 GB of RAM. The data link from the robot to the docking station is either Ethernet or WiFi.

The processing computer acts as an online server. It collects the data sent to it from the docking station. If this system was deployed in Walgreens, processing would be completed on Walgreens' servers and uploaded to an interface for viewing at Walgreens corporate headquarters. Currently, the data are formatted using MATLAB scripts and product recognition is handled by functions written in C. The computer has a 2.9-GHz Intel Core i7 with 8 GB of RAM. The output of the system is an image file containing the semantic map.

IV. Semantic Mapping

In this section, we first describe the planning and navigation algorithms necessary for the robot to successfully traverse cluttered retail environments, such as the Walgreens store shown in Fig. 4, in order to collect product images. These images are stamped with the pose of the robot in the map frame and uploaded to a server for processing. The system identifies the products within each individual image using a soft-object detector, maintaining multiple potential class labels for each object. Next, using the position of the robot, the most likely class label of each object, and the size information for each object, the system creates a virtual map of the store. Finally, the system segments this virtual map into departments using the object-to-department associations. Objects are then relabeled according to the departments to increase the final precision of the object labeling system. The image processing and semantic map generation occur off-line. Fig. 3 shows the planning, navigation, and semantic mapping system pipelines.

A. Path Planning

To perform the semantic mapping task, the robot must be capable of navigating in a cluttered, indoor environment. To deal with this, the robot plans a nominal path through the environment and then adapts this plan online based on local sensor information. These nominal paths maintain a desired distance from the products on the shelves, as shown in Fig. 6, in order for the robot to be able to correctly detect and identify products using the camera.

1) Map Generation: Initially, the robot is given an occupancy grid map of the environment or such a map is manually created [first block in Fig. 3(a)]. To manually create a map, a user steers the robot around the store, and the data from the laser range finder are fed into the `gmapping` package from robot operating system (ROS) [14] to create a high quality map of the store. An occupancy map generated from a Walgreens store in Philadelphia is shown in Fig. 4(a). The gray regions

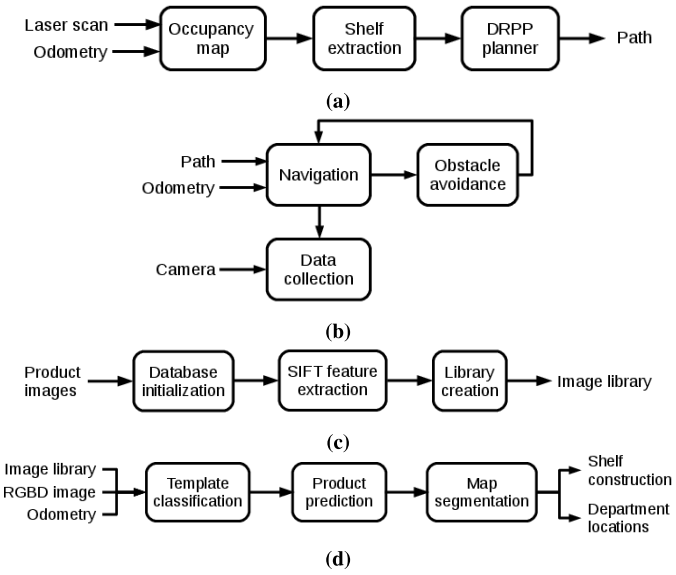


Fig. 3. System diagrams for planning, navigation, template library construction, and semantic mapping.

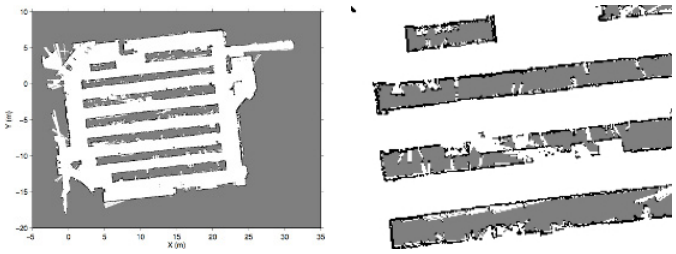


Fig. 4. Generated occupancy grid map of a Walgreens store. (a) Full store map. (b) Inset showing details of the shelves.

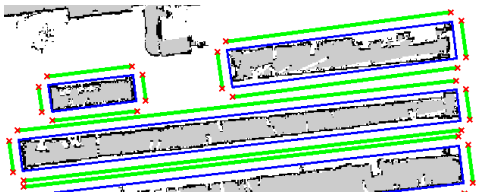


Fig. 5. Extracted shelves (blue boxes) and corresponding segments to be visited (green lines).

indicate unexplored areas while the white ones represent free space and black regions are occupied.

2) Shelf Extraction: The robot must first extract the contours of the shelves from the occupancy grid map before it can plan a path through the environment [second block in Fig. 3(a)]. This is done by morphologically closing the occupancy map using a disk structure and then filling holes [Fig. 5(a)]. We morphologically remove interior points [Fig. 5(b)] before using an edge-linking operation to detect potential shelving units. We discard any edges shorter than a certain threshold as clutter objects [Fig. 5(c)]. Note that shelves will typically result in closed edges, though some nonclosed edges within a certain neighborhood must be closed. Finally, we obtain an oriented bounding box for each closed

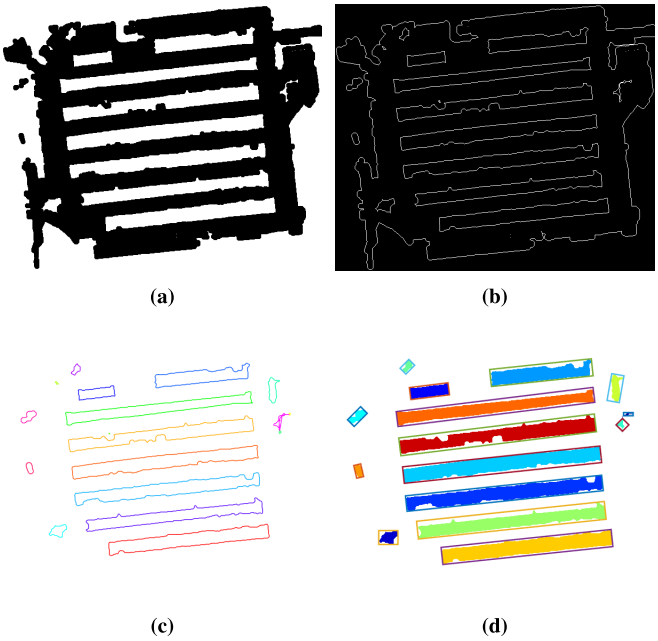


Fig. 6. Shelf extraction process. (a) Morphologically closed occupancy map. (b) Interior points removed. (c) Edges obtained from edge-linking operation. (d) Oriented bounding boxes for the edges.

edge [Fig. 5(d)]. For all of these steps, we use the open-source image processing implementations from [15].

Using either the bounding box or the extracted shelf contours, the robot creates a set of segments at some desired offset distance (2 ft) from each edge of the shelf, as Fig. 6 shows. This offset distance is set to allow the robot to avoid colliding with the shelves while keeping the products on the shelves in focus in the camera images. Note that with the onboard camera, even the smallest products are on the order of 50×50 pixels at this distance. Most shelves in a Walgreens store are rectangular and have end stands, so the robot must visit all four sides. These sides must be traversed in a particular direction (clockwise in our case), since the camera is mounted facing to the right with respect to the heading direction of the robot.

3) Planning: To plan an optimal path through the store, we represent the store as a directed graph. Each of the extracted shelf segments becomes a node in the graph. We create directed edges, or arcs, between all nodes, with the weight being the distance from the endpoint of the first shelf to the starting point of the second shelf. Note that these edges are not symmetric, since the distance from the endpoint of shelf A to the beginning of shelf B is different than the distance from the end of shelf B to the beginning of shelf A. This type of planning problem is an arc routing problem (ARP) [16]. Generic ARP solvers find a least-cost traversal of some arcs or edges of a graph subject to constraints. Let m be the number of shelves in the store and let n be the total number of sides/segments of the shelves to be visited. Consider a graph $G = (V, A \cup E)$ where $V = \{v_1, v_2, \cdots v_n\}$ is a set of vertices, A is a set of directed arcs $a_{ij} (i \neq j)$, and E is a set of undirected edges $e_{ij} (i < j)$. Let c_{ij} be the cost of traversing arc a_{ij} and d_{ij} be the cost of traversing edge e_{ij}. Let $A' \subset A$ and $E' \subset E$ be the subsets of arcs and edges that the

robot must traverse while the remaining arcs and edges, $A \setminus A'$ and $E \setminus E'$, are optional. For our problem, $E = E' = \emptyset$, and hence, the graph G is a directed graph. This class of ARPs is known as the directed rural postman problem (DRPP) [16] [third block in Fig. 3(a)].

ARP solvers typically transform the problem to a node routing problem, also called a traveling salesman problem (TSP), as there are many readily available tools to solve TSPs. Laporte [16] provides a unified approach for transforming various classes of ARPs into TSPs. The first step is to transform the DRPP on G into an asymmetric TSP (ATSP) on H, where $H = (W, B)$ is a complete graph. There is a vertex $w \in W$ for each arc of A' in the original graph and an arc $b_{jk} \in B$ with cost s_{jk} equal to the length of a shortest path from arc $a_{ij} \in A'$ to arc $a_{kl} \in A'$. The next step transforms the ATSP on H to a symmetric TSP (STSP) on a complete undirected graph I using a three-node transformation proposed in [17]. The new graph $I = (U, C)$ contains three copies of the vertices in H, i.e., $\exists u_i, u_{n+i}, u_{2n+i} \in U$, such that $u_i = u_{n+i} = u_{2n+i} = w_i, \forall w_i \in W$ [18]. Let the cost of the edges $c_{i,n+i}, c_{n+i,2n+i}$ be 0, the cost of edge $c_{2n+i,j}$ $(i \neq j)$ be s_{ij} (i.e., the cost of $b_{ij} \in B$), and the cost of all other edges be ∞.

We use the publicly available Concorde TSP solver [19], which uses a branch-and-cut algorithm to solve the STSP on the graph I. The solver provides a least-cost sequence of the segments for visiting the shelves. Our contribution to robot navigation is applying these well-studied problems and open-source solvers to the problem of robotic navigation in retail environments. This differs from previous approaches to retail navigation, such as [20], where a floor cleaning robot follows a human specified path.

B. Navigation

The nominal path for the robot is composed from the sequence of shelves found from the STSP problem. The robot discretizes each segment along the path to get a series of waypoints. These waypoints are sequentially set as goals in the locally reactive controller from Guzzi *et al.* [21].[2] This controller provides the Navigation and Obstacle Avoidance blocks in Fig. 3(b). This approach does not provide any formal safety guarantees for the robot, but we have had no problems throughout our extensive development and testing. In addition, the algorithm does not require perfect knowledge of the map or of the locations of other robots and people, only raw laser scans.

When the nominal path is unobstructed, the robot drives straight toward the next waypoint, and when a transient obstacle blocks the robot's path, the robot drives around the object and returns to the nominal path. To avoid obstacles, the approach in [21] inflates all of the obstacles in the current laser scan and steers the robot toward the point in free space that is closest to the current waypoint. This allows the robot to reactively replan safe paths around static obstacles, such as a box, as shown in Fig. 2(b), as well as avoid moving obstacles, such as people. We bias this replanning toward the nearest shelf to avoid having the obstacles block the camera view.

The robot uses the adaptive Monte Carlo localization (AMCL) algorithm [22] to track its position in the occupancy grid map as it moves. An implementation of this algorithm is available in the `amcl` ROS package [23]. This provides the robot with the odometry input in Fig. 3.

C. Object Detection

Recent computer vision literature has seen an explosion of techniques in a race toward the perfect image feature descriptor. At a high level, families of feature descriptors include those that use image gradients, binarized colors, and image patches [24]–[27]. Recent research is biased toward features with a compact descriptor length, such as Fast Retina Keypoint, to enable high performance on resource constrained platforms, such as mobile devices. While these descriptors reduce the computational overhead, SIFT remains the standard for performance in variable lighting conditions [24]–[27].

1) Template Object Library: The processing computer has a library of all of the product classes that could be found in a store. This library contains information about the physical characteristics and department associations for each product. Each individual product class S in the store has at least one corresponding image template, depending on the type of product. Some objects are constrained to one orientation on a shelf, such as objects hanging from shelves. Other objects might be placed in multiple positions or be deformable, e.g., a bag of chips. Each template ω consists of a set of image descriptors $D_\omega = \{d_k\}_{1 \leq k \leq K_\omega}$. Since accuracy is our primary objective, we extract SIFT descriptors from a uniform grid spaced every five pixels using a dense keypoint search across input images and training templates [28], [29]. The number of keypoints can be different for each template ω, so we normalize the total number of descriptors, K_ω, across all product types by random selection. The template object library $\Omega = \{\omega\}$ is the collection of all of the individual templates. Fig. 7 shows an example product template. We extract SIFT descriptors at multiple scales and organized in a list, associating them with the product name and Universal Product Code. In practice, we extract descriptors over two scales.

2) Camera Measurement: From each camera image, we extract a dense set of SIFT descriptors from the entire image. We cannot use conventional SIFT keypoint detection, because there are no prestored images for entire scenes or shelves. For our application, we found that dense SIFT (DSIFT) features yield higher accuracy [28] than standard SIFT features. We project each pixel to a plane based on laser depth data. We then perform a nearest-neighbor search over each keypoint in the projected plane to determine its nearest neighbor in the template library. Previous work, such as [30], does well to argue the advantages of nearest-neighbor classifiers over parametric approaches, such as support vector machines. We employ a naïve Bayes nearest-neighbor classifier [30] that minimizes

$$\prod_{i=1}^{N} ||d_i - NN_S(d_i) + \text{Dist}_{NN_S(d_i)}||^2 \quad (1)$$

[2]Our implementation is available online at https://github.com/bcharrow/scarab/tree/master/hfn

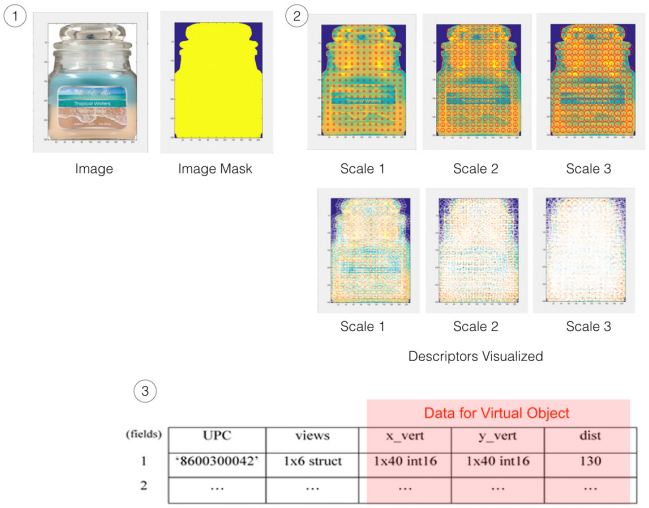


Fig. 7. Sequence for building template entry. (1) Create image mask of the object. (2) Extract keypoints at multiple scales (with the descriptors displayed in the following). (3) Build data library.

where $d_1, \ldots, d_N$ are the descriptors extracted from the current frame, $NN_S(d_i)$ is the nearest-neighbor descriptor of d_i in class S [30], and $\text{Dist}_{NN_S(d_i)}$ is a probability score based on the number of times the quantized nearest neighbor occurs in a training set normalized over the total descriptors in that set. We use a voting scheme across all classes of the form

$$H_S = \sum_i \frac{1}{||d_i - NN_S(d_i) + \text{Dist}_{NN_S(d_i)}||^2} \times \int_x \int_y f\left(v_x^i, v_y^i\right) dxdy \tag{2}$$

where $[v_x, v_y]^T$ is a vector from the template point to the center of the template itself

$$f\left(v_x^i, v_y^i\right) = \exp\left(-\frac{\left(x_w^i - v_x^i\right)^2}{2\sigma_x^{2i}} - \frac{\left(y_w^i - v_y^i\right)^2}{2\sigma_y^{2i}}\right) \tag{3}$$

$[x_w, y_w]^T$ is the center of the Gaussian vote stamp, and σ_x and σ_y are the window sizes of the Gaussian vote stamp.

This yields a voting table for each product, as seen in Fig. 8. This voting algorithm over all templates is bounded by $O(N^2 Ks)$ complexity, where N is the number of incoming features, K is the number of features per template, and s is the number of templates. We sort product heat maps according to their maxima, $p(S_s) = \max_{x,y}(H_S)$. C_j is initialized at x, y for $p(S_s) > \epsilon$, where ϵ is some threshold value, and each centroid is associated with a probability confidence $(C_j, p(S_s))$ based on the object classifier output, as Fig. 9(c) shows. This algorithm provides the Template Classification block in Fig. 3(d).

D. Map Representation

A semantic map in this context consists of objects relevant to a robot working in a retail environment—shelves and products—where sections of shelving are labeled according to their department. Since retail environments are densely packed with many different products, each image of the shelves will likely contain multiple product classes.

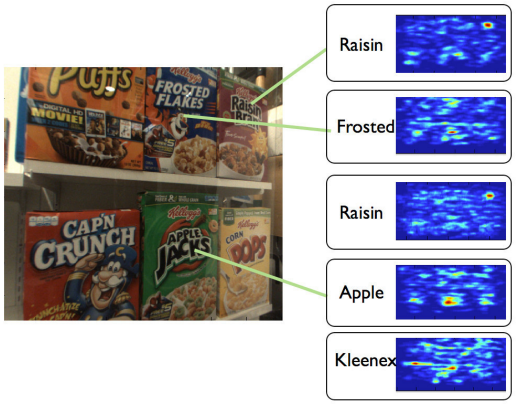


Fig. 8. Camera frame and its corresponding product heatmaps.

Traditional semantic mapping approaches create a 3-D point cloud from RGB-D (RedGreenBlue-Depth) sensor input or the integration of laser data and camera data [3]. A point cloud, M_p, consists of points $p_i, \ldots, p_n \in \mathbb{R}^3$. These points are grouped into segments, T_l, based on similar characteristics. The space is discretized into a number of regions, with each region being classified as either an object or a surface using the methods described in [32] and [33]. Each potential object belongs to a single class S [3] and objects are represented using a tuple

$$O = \{K, D, C\} \tag{4}$$

where $C \in \mathbb{R}^3$ is the centroid of the potential object in the map frame, K is a set of key points, and D is a corresponding set of image feature descriptors. The position in the map frame is computed using the sensor data combined with the pose estimate from the localization system. Each object also has a confidence measure, which comes from the covariance matrix of the localization system.

We take an alternative approach that exploits the *a priori* information about the objects in the environment. We manage two concurrent map representations, as shown in Fig. 9(a) and (b). The first is a traditional point cloud, M_p, generated from laser range data and a monocular camera. The second is a virtual map of the recognized objects and their positions, M_{vp}. This is distinct primarily, because once an object is recognized in the point cloud, a full scaled model of that object is added to the virtual map, as Fig. 10 shows. Our novel virtual map provides several advantages. First, it only stores information that is relevant to the robot's task. Second, it provides the object classification system with information about what the robot will see from different viewpoints. Finally, it improves object classification by applying spatial constraints to accurately segment and classify nearby objects.

Rather than giving objects hard labels, we adopt the soft-object representation described in [33]. We maintain a list of potential classes S_s along with a corresponding probability for each class. Thus, the object representation described earlier is

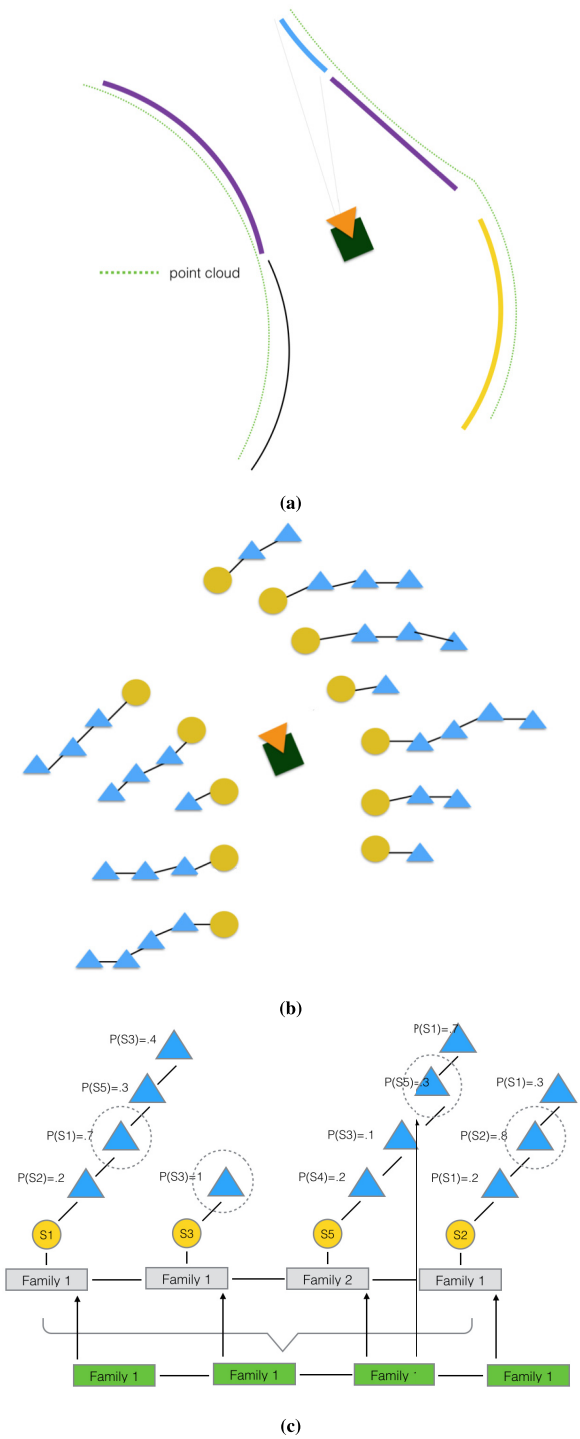


Fig. 9. Figures compares (a) conventional map and (b) our soft-object semantic map. Conventional approaches attempt to assign each point in a cloud to a particular object, where the color indicates different objects. Our approach initially assigns multiple labels (blue triangles) to each object (yellow circles). (c) We use the most likely class for each object to create an initial department segmentation and labeling. We update the likelihood of the labels for each object using this department segmentation and label the final map using the most likely updated class for each object.

changed to

$$O = \begin{Bmatrix} p(S_1), K_1, D_1, C \\ \vdots \\ p(S_s), K_s, D_s, C \end{Bmatrix} \tag{5}$$

Fig. 10. Example shelf in the virtual map overlaid on an image of the physical store.

where there are s potential class labels for the object and

$$\sum_{i=1}^{s} p(S_i) = 1. \tag{6}$$

Objects are transformed to the map frame using the robot pose associated with the image and the laser depth data. We aggregate these objects into a single map M_{vp}. We add a new object to the virtual map when an object is detected in an image and that detected object is not within some threshold distance of an existing object. When a new object is added, we normalize the class probabilities $p(S|\omega)$, save the descriptors and key points for each respective class K_ω and D_ω, and find the centroid location C.

E. Map Segmentation

We wish to partition the map into departments of semantically related products. Map segmentation is most often performed over a point cloud, M_p, which is represented by an undirected graph $G = (V, E)$. The vertices, $v_a \in V$, are points in M_p, and the edges, $e_{ab} \in E$, correspond to pairs of neighboring vertices (v_a, v_b). Each edge e_{ab} has a corresponding weight, w_{ab}, which is a nonnegative measure of dissimilarity between neighboring elements v_a and v_b. In image segmentation, the elements in V are pixels, and the weight is a measure of the dissimilarity between two pixels, e.g., the difference in intensity, color, motion, or location. Segmentation algorithms partition V into components, such that each region T corresponds to a connected component in the graph G.

Other work represents the segmentation problem as a dynamic programming (DP) problem. DP has been applied over images in several domains, including noise filtering, edge detection, and contour segmentation. Most notably, in [34], DP is applied to parse the facade of a building. The approach in [34] initializes the segmentation process by first labeling each pixel based on a classifier

$$p(S_s) = \log p_{m,s}(S_s) - \log \Sigma_{S_\omega \in S_\Omega} p_{m,s}(S_\omega) \tag{7}$$

where $p_{m,s}$ is the multinomial probability distribution of pixel s over the label space S_Ω and S_s is the normalized log-likelihood output of the classifier. We apply the maximum likelihood label to each pixel, and then use structural information to further constrain the pixels.

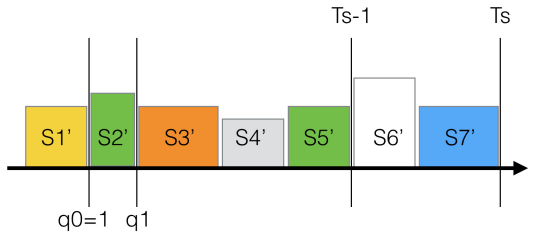


Fig. 11. Segmentation occurs over one dimension where each VO S is grouped into a region T.

A conventional approach would determine the class S and label all points in a segment T with the associated class. Instead, we place a virtual object (VO) of the maximum likelihood class in the virtual map M_{vp}. This VO has the size and shape of the object class from the template library. Our segmentation algorithm runs over this virtual map, combining object recognition and region segmentation into a single step. We initialize the department boundaries using the boundaries of the VOs, as Fig. 11 shows. We then determine segments (T) composed of objects VO (products or shelves), where each segment corresponds to a department in the Walgreens store. Recalling that each product type is associated with one or more department labels, we use the labels of nearby objects to influence the final estimate of each object's class. If a VO is assigned to a department T during the segmentation, but the class S_{VO} cannot appear in department T, then the class S_{VO} changes to the maximum likelihood class that can appear in T.

The DP problem, outlined in Algorithm 1, is formulated as a segmentation of a 1-D signal, $q[0], q[1], \cdots q[s-1]$, into k segments, where each q is a boundary of the model object class estimated to be located at that position [35]. For k departments, there are $k-1$ transitions, $\{t_1, \ldots, t_{k-1}\}$, in addition to the start, $t_0 \equiv 0$, and end, $t_k \equiv s$, of the shelf. The i^{th} segment has the probability density function (pdf)

$$p_i(q[t_{i-1}], \ldots, q[t_i - 1]). \tag{8}$$

With the assumption that each department is statistically independent, the pdf of the data set is

$$\prod_{i=1}^{k} p_i(q[t_{i-1}], \ldots, q[t_i - 1]) \tag{9}$$

where the maximum likelihood estimate segmenter chooses $t_1, t_2, \ldots, t_{k-1}$ and k to maximize (9). This algorithm provides the Map Segmentation block in Fig. 3(d).

Algorithm 1 DP Department Segmentation finds the optimal segmentation of a string of products into departments using DP. For clarity, the algorithm demonstrates the computation of the score of the optimal segmentation. As is typical for DP solutions, the segmentation labels are recoverable by additionally storing the argmax and performing a standard DP traceback* from the optimal solution.

Input: String $S_1 \ldots S_s$ of products S
Output: Segmentation of products into $T_1 \ldots T_k$ of departments T
Let $opt(j,k)$ be the optimal solution score using $S_1 \ldots S_j$, with k segments
Let $score(i,j,t)$ be the score of the department t using products $S_i \ldots S_j$
Let k_{max} be the maximum number of segments to consider
Let $optsoln$ be the optimal solution resulting from the optimal segments
for $j \leftarrow 1$ ***to*** s **do**
  $opt(j,0) \leftarrow 0$
end
for $k \leftarrow 1$ ***to*** k_{max} **do**
  for $j \leftarrow 1$ ***to*** s **do**
    $opt(j,k) \leftarrow \max_{1 \le i < j} \left[opt(i,k-1) + \max_{t \in T} score(i+1,j,t) \right]$
  end
end
$optsoln \leftarrow \max_{1 \le k \le k_{max}} opt(s,k)$
$T_1 \ldots T_{k_{opt}} \leftarrow StandardDPArgMaxTraceback(opt, optsoln)*$
return $T_1 \ldots T_{k_{opt}}$

V. Analysis

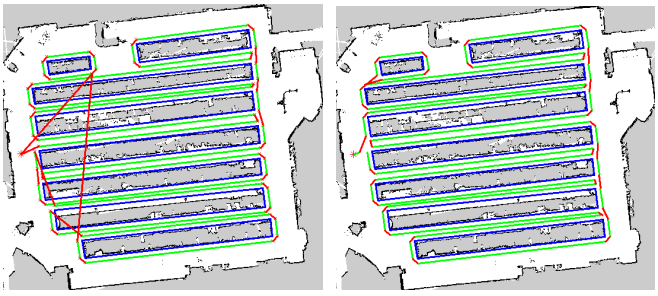
A. Evaluation of the Planning and Navigation Pipelines

To validate the efficacy of our planning and navigation systems, we tested the robot in several different scenarios, varying the configuration of the planner and testing in multiple environments. We demonstrate that the optimal planner described in Section IV-A is able to generate efficient paths through the stores and the navigation algorithm described in Section IV-B is able to accurately track these paths. We perform this analysis in simulation using the Gazebo robot simulator[3] and the maps created in two real-world stores.

We first tested the optimal planner presented against a greedy planner that drives the robot to the nearest unvisited segment. Fig. 12(a) shows a path followed by the robot for a greedy solution, while Fig. 12(b) shows the output of the optimal planner. In either case, the robot must visit all of the shelf segments, so the difference is the order in which the robot visits the shelf segments. While the greedy planner does well for most of the run, visiting the last few shelves requires the robot to traverse the width of the store, significantly increasing the total distance traveled compared with the optimal planner.

As described in Section IV-B, we examine two different methods of traversing the shelves: following the straight lines of the bounding box and following the contours of the shelves. The contour-following method, shown in Fig. 13, has longer paths than the line-based method, shown in Fig. 12(b). To guide the robot, we interpolate these lines or contours to get a sequence of waypoints for the robot to visit. We also

[3] Available at http://gazebosim.org/



(a) (b)

Fig. 12. Output paths for the greedy and optimal planners in the first store. In order for the robot to visit all of the shelf segments (green lines), it must drive between shelves (red lines). The shelf segments are straight lines that we extract from the bounding boxes of the shelves. (a) Greedy solution. (b) Optimal solution.

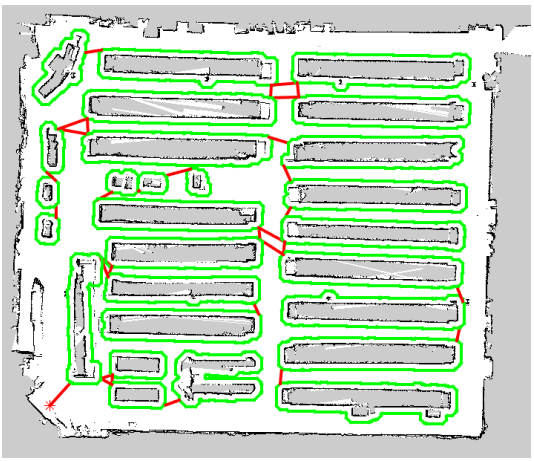
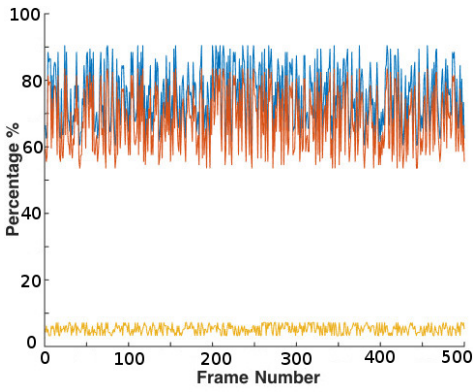


Fig. 13. Optimal planner output in a second, larger store. The robot starts and ends the experiment in the lower left corner. The robot follows the contours of the shelves.

study the effect of the waypoint spacing, selecting a dense set of points (3-cm spacing) and a spare set of points (30-cm spacing).

Table I shows the average path length for various planner configurations. Each configuration was tested using four different starting locations of the robot. The nominal path lengths using the bounding box and contour methods from Section IV-A are given in columns "Line" and "Contour." The actual path lengths from the Gazebo simulations are in the "Gazebo" column and the reported percent difference in column "% diff CG" is between the planned and simulated contour-following paths in columns "Contour" and "Gazebo." The rows labeled "Shelf" give the length of the shelf segments to be viewed, while "Greedy" and "Optimal" are the total path lengths, which includes both the shelf segments and the connections between shelves. The rows labeled "% diff GO" show the percent difference between the "Greedy" and "Optimal" plans.

We see that in store 1, densely selecting waypoints (rows "1D" in Table I) increases the path length by more than 10 m compared with the sparse waypoints (rows "1S" in Table I). When tested on the robot in a simulated environment, the sparse path was much more similar to the actual path



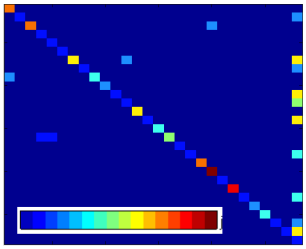


Fig. 14. (a) Precision (blue curves), recall (red curves), and localization error (yellow curves) as evaluated over a test data set in a Walgreens store. (b) Confusion matrix for the test data sets. Each row corresponds to a product in the store and each column is a recognized product. The last row and column represent the background. The left-bottom bar shows the color key from min to max.

TABLE I

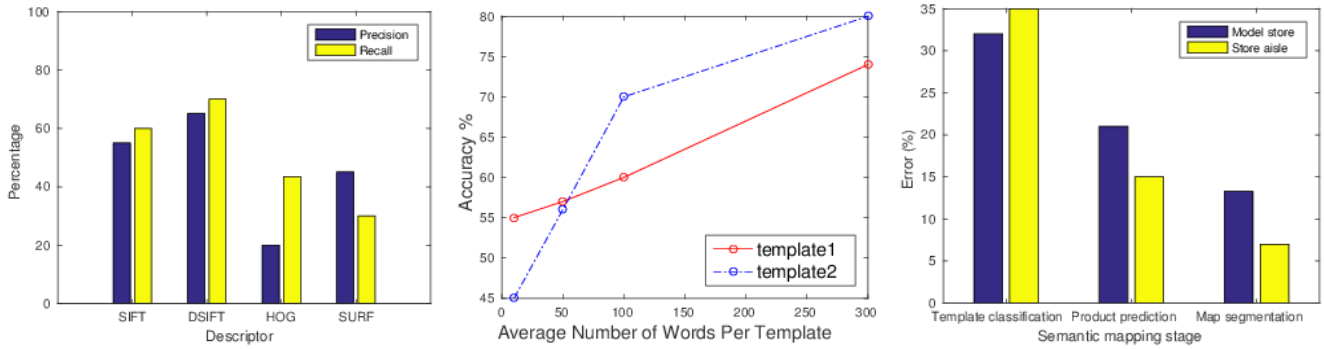
COMPARISON OF AVERAGE PATH LENGTH OF DIFFERENT PLANNING METHODS IN MULTIPLE ENVIRONMENTS

		Line [m]	Contour [m]	Gazebo [m]	% diff CG
1D	Shelf	271.4	296.0	296.0	–
	Greedy	325.4	342.9	333.0	2.97
	Optimal	295.6	311.2	292.2	6.58
	% diff GO	9.15	9.24	12.32	–
1S	Shelf	271.4	282.1	282.1	–
	Greedy	325.4	329.0	329.2	0.07
	Optimal	294.7	297.2	289.4	2.69
	% diff GO	9.44	9.67	12.11	–
2S	Shelf	511.5	547.6	547.6	–
	Greedy	644.5	657.2	636.1	3.31
	Optimal	585.8	599.8	570.3	5.18
	% diff GO	9.10	8.73	10.35	–

(2.69% difference) than the dense path (6.58% difference). This is due to the robot "smoothing" out the path as it drives along. The simulated path length is also smaller than the planned path length, again due to this "smoothing" behavior of the navigation algorithm. We also see that the optimal planner is 9%–12% better than the greedy planner across all configurations, a significant margin given that 90%–95% of the total path length for the optimal planner is simply traversing the shelves. The same trends hold in store 2, where we also used a path with sparse waypoints (rows "2S" in Table I).

B. Evaluation of Semantic Mapping Pipeline

In order to independently evaluate our computer vision pipeline, we collected a data set at Walgreens using the camera on a mobile device with 2.2-mm focal length and 8-Mpixel resolution. Fig. 14 summarizes the results of this initial study



(a) **(b)** **(c)**

Fig. 15. Evaluation of the computer vision system. (a) Product classification precision and recall for different image feature descriptors. (b) Product classification accuracy versus the number of DSIFT keypoints per template in the template object library. (c) Department classification error during each stage in the semantic mapping pipeline in two environments.

and is a confusion matrix evaluating the accuracy of the product recognition.

We next analyze our system using the data from the store aisle shown in Fig. 10. We evaluate our system in a similar way to [36]. In [36], the goal is to recognize products on grocery store shelves. However, the number of object classes is much lower than the number of classes in this paper. Fig. 15 shows the results of this analysis. First, we examine how different image feature descriptors affect the product classification rate. Fig. 15(a) shows the individual product classification precision and recall for different image features. Overall, we see that SIFT and DSIFT have significantly higher precision and recall rates than the histogram of gradients and speeded up robust features. We use the DSIFT features, since the primary goal of our mapping system is accuracy.

Next, we examine the effect of the number of features per template object on the product classification rate. Fig. 15(b) shows the product classification accuracy as we vary the number of words (features) per template for two example products. In general, the classification accuracy increases as the number of features increases. However, at some point, the returns begin to diminish, as we see for template 2. The cutoff point varies for each template and depends on the size and visual complexity of the packaging, with larger and more complex packages seeing higher returns for large word counts.

Finally, we examine the effects of each stage in our semantic mapping pipeline from Fig. 3(d) on the classification error in the final map. Fig. 15(c) shows the department classification error rate for different configurations, where we see significant decreases in the error rate after both our soft assignment algorithm (Section IV-C) and our DP segmentation algorithm (Section IV-E). This result holds across environments, though the effects are smaller in the model store where there are fewer product types.

Fig. 16 shows the pixel point cloud generated during the robot's trajectory. Measurements are taken at each frame. We have developed a graphical user interface (GUI) to enable a human to easily view the virtual map generated by the robot. Each department along the shelf is color labeled. The human user is able to click on a section to explore the virtual product map in that department. Figs. 17(c) and 18(b) show the screenshots of this GUI.

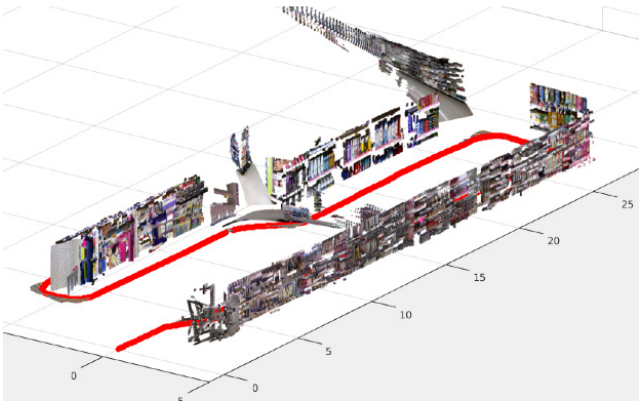


Fig. 16. Example aisle in the pixel point cloud generated by a robot following the red trajectory.

VI. Experiments

We conducted a series of experiments to test the ability of the robot to navigate a retail environment with natural clutter and to test the semantic labeling system. First, we tested our system in the model store from Section III-A. Only a single row of products was placed on the shelves in the model store, since the camera will typically only capture the first row of products on a shelf even when there are products behind them. The positions of products changed between runs, but product labels always faced outward from the shelf, as is typical in retail environments. In each trial, a movable object (i.e., a box) was placed in a random position. Fig. 2(b) shows the path of the robot in one of the test runs. In this trial, the robot moved closer to the shelf to avoid the movable object, which is not included in the map of the store. Overall, the robot is biased to move toward the shelves. This improves object recognition, since each product is larger in the image, even if the total number of products in each image decreases. Our product recognition rate was highest for large objects. Small products, such as medicine and cosmetics, were more

TABLE II
MEASUREMENT ERRORS FOR DEPARTMENTS IN MODEL STORE

Cereal	Medicine	Superbowl	Skin care	Flu	Soda
7.7%	13.5%	20.4%	35.0%	31.9%	22.2%

TABLE III
MEASUREMENT ERRORS FOR DEPARTMENTS IN REAL STORE AISLE

Snacks	Cookies	Nuts and Mix	Cereal
5.1%	3.2%	21%	5.1%
Baked Goods	**Condiments**	**Canned Goods**	**Soda**
6.1%	4.3%	5.1%	2.25%

TABLE IV
SEGMENTATION STATISTICS

	Model store	Real store
Total length of path	3.97 m	9.89 m
# of departments	6	8
Average length of departments	0.39 m	0.72 m
Initial # of products misclassified	10	25
Final # of products misclassified	3	17

difficult to detect. We repeated these tests in an aisle of an actual Walgreens store.

Figs. 17(a) and 18(a) show two camera sequences overlaid with the object classes detected in the images. These sequences are from the model and actual store, respectively. The system aggregates these measurements from single images in order to build the VO maps shown in Figs. 17(b) and 18(b). To evaluate the accuracy in the physical dimensions of each department, we compute the error between the labeled departments in the final semantic map and their actual dimension in the model store. The average error in the size of each department was 13.3 cm, which is 12.3% of the department size. Tables II and III show the percentage of shelf pixels that were correctly labeled in the output map [Figs. 17(c) and 18(b)] for each department. The overall error was 13.2% in the model store and 4.6% in the store aisle.

Table IV shows the results of our segmentation algorithm in both the model store and the actual store aisle. The actual store is larger and it contains larger departments. These larger departments have more products in them, leading to more products being initially misclassified after the template classification step and in the final map after the map segmentation step. The real store has significantly more misclassified products. Some of these were reclassified, if they did not belong to the estimated department, but even the maximum likelihood template in the correct department was still incorrect.

Despite these errors in individual product classifications, our system is able to successfully determine the identity and order of the departments on the shelves in each trial. The system has the highest error with departments that are very narrow and departments on the end stands of the aisles. This is likely due to the fact that small departments contain fewer items, so errors in individual product classification have a larger impact on the department label. Departments on the end stands have the additional challenge that the robot views

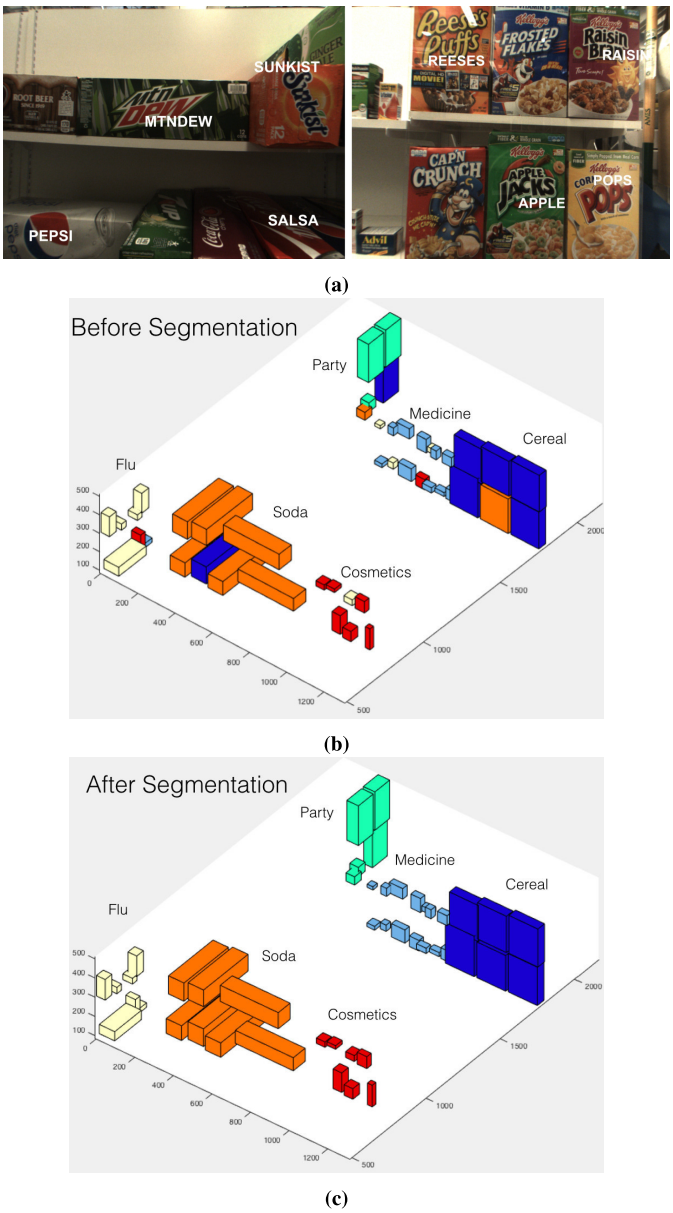


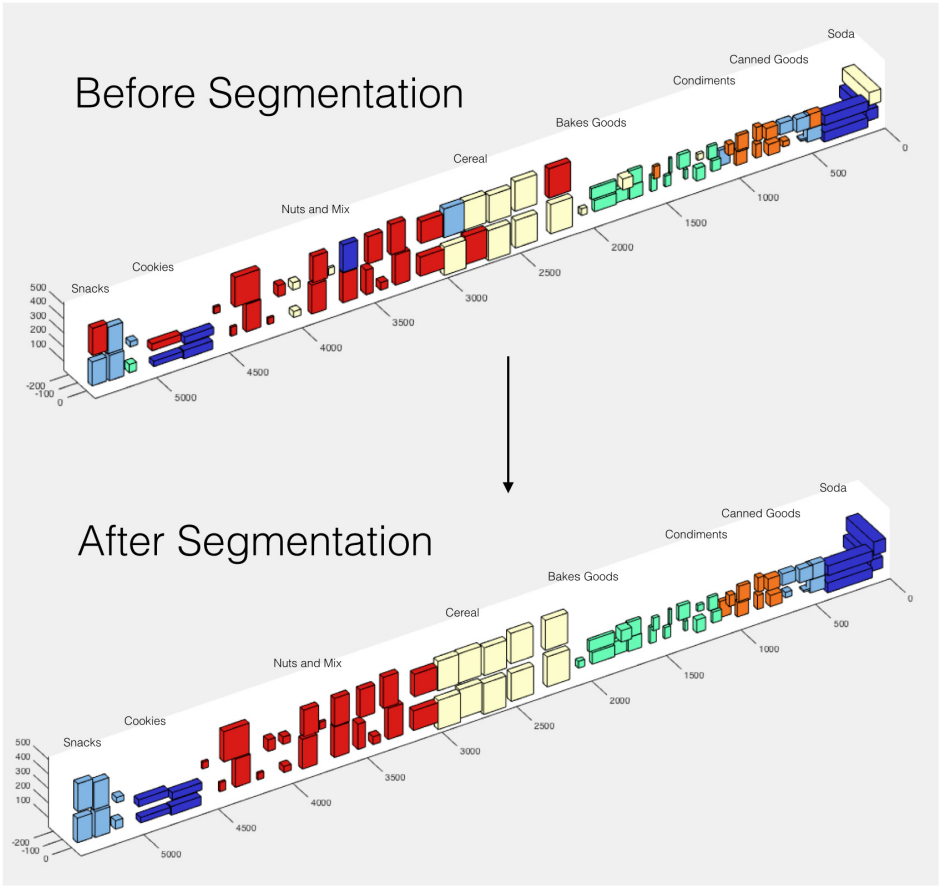
Fig. 17. Semantic mapping process for model store. (a) Object detection over store shelf. (b) Virtual map. (c) Segmented map.

them while it is turning. This means that the camera is not fully orthogonal to the shelf, which increases the classification error, because most of the products are not in focus in the images. Despite these errors, the average measurement error across departments is relatively small, considering the application and intended use of the semantic map: a retailer will have sufficient information about the layout of the store to make decisions about reorganizing items and departments. The system is able to successfully navigate around small clutter objects in the store and is able to correctly classify products covered by glass or plastic windows.

The attached multimedia files contain a video[4] of the robot navigating in our model store and in a full retail store. The video also shows the map generation, shelf extraction, path

[4]Also available at https://www.youtube.com/watch?v=a7TDgwQ5FlM

(a)



(b)

Fig. 18. Semantic mapping process for an aisle in Walgreens store. (a) Object detection over store shelf. (b) Automated semantic map generated over an actual Walgreens store aisle. We preselect which VOs with which to tabulate the database according to their frequency and recurrence across all Walgreens stores. The products are colored according to the departments. We can see that the map segmentation step significantly reduces the product classification error.

planning, and autonomous navigation processes in a large, real-world retail store.

VII. Conclusion

In this paper, we describe an automated robotic system that can successfully navigate a retail environment to construct a semantic map of the inventory in the store. Our robotic system is able to reliably navigate through the cluttered environment to collect images using the onboard camera. The robot autonomously generates a distance-optimal path that visits each shelf only once, and it follows this path while avoiding any unmapped obstacles, such as boxes, shopping baskets, or people. Using the data collected from the store, the system then detects all of the potential products in each image by combining the performance of weak classifiers over associated objects. Each potential object is given a soft label to account for the fact that many products share similar shape, color, and size characteristics. The system then automatically segments the map into regions using the most likely product labels and assigns department labels to each region, using information about which products belong to each department. Finally, the products are relabeled using the department labels, significantly improving the accuracy of the product labeling.

We provide experimental results showing our system correctly and accurately labeling a model store and an aisle from an actual retail store. The model store contains over 60 product

types in six departments while the retail store aisle contains over 500 product types in eight departments. We also show that our robotic system is capable of autonomously navigating and collecting data from a full scale retail store.

We are currently working to scale up our system and to test it out with other retailers. Departments and product types vary widely across retailers, for instance, the average Walgreens contains some 15 000 product classes over 11 000 square feet, while the average Costco contains 40 000 product types over 150 000 square feet. We are also focusing on improving the accuracy of the labeling system by using additional visual information, such as barcodes. We are also looking at ways to take advantage of the fact that store departments usually remain within the same general vicinity in the floor plan even when their size and exact product contents change [37]. This is due to factors, such as consumer traffic patterns and the placement of specialty containers (e.g., refrigerators). Future work will represent each aisle with its own soft-object semantic map. This allows the robot to use a smaller template object library, which will only include the departments typically near that location, and will improve classification accuracy and speed.

REFERENCES

[1] J. Cleveland *et al.*, "Autonomous vehicle for mapping store layout," U.S. Patent 15 158 376, May 18, 2016.

[2] J. Cleveland *et al.*, "Automatic mapping of store layout using soft object recognition," U.S. Patent 15 15 464, May 18, 2016.

[3] S. Choudhary, A. J. B. Trevor, H. I. Christensen, and F. Dellaert, "SLAM with object discovery, modeling and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 1018–1025.

[4] N. Blodow *et al.*, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2011, pp. 4263–4270.

[5] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3D point clouds for indoor scenes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 244–252.

[6] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from RGB-D video," *J. Real-Time Image Process.*, vol. 10, no. 4, pp. 599–609, Dec. 2015.

[7] R. B. Rusu, *Semantic 3D Object Maps for Everyday Robot Manipulation*. Berlin, Germany: Springer-Verlag, 2013. [Online]. Available-http://www.springer.com/us/book/9783642354786

[8] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz, "Semantic object maps for robotic housework—Representation, acquisition and use," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 4644–4651.

[9] K. Mankodiya, R. Gandhi, and P. Narasimhan, "Challenges and opportunities for embedded computing in retail environments," in *Sensor Systems and Software*. F. Martins, L. Lopes, H. Paulino, eds., Berlin, Germany: Springer-Verlag, 2012, pp. 121–136.

[10] S. Kumar *et al.*, "Remote retail monitoring and stock assessment using mobile robots," in *Proc. IEEE Int. Conf. Technol. Pract. Robot Appl. (TePRA)*, Apr. 2014, pp. 1–6.

[11] E. Frontoni, M. Contigiani, and G. Ribighini, "A heuristic approach to evaluate occurrences of products for the planogram maintenance," in *Proc. IEEE/ASME Int. Conf. Mechatron. Embedded Syst. Appl. (MESA)*, Sep. 2014, pp. 1–6.

[12] J. Cleveland *et al.*, "An automated system for semantic object labeling with soft object recognition and dynamic programming segmentation," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Gothenburg, Sweden, Aug. 2015, pp. 683–690.

[13] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multirobot teams," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 53–61, Mar. 2008.

[14] B. Gerkey. (Sep. 2014). *Gmapping*. [Online]. Available: http://wiki.ros.org/gmapping

[15] P. D. Kovesi. (Sep. 2014). *MATLAB and Octave Functions for Computer Vision and Image Processing*. [Online]. Available: http://www.peterkovesi.com/matlabfns/

[16] G. Laporte, "Modeling and solving several classes of arc routing problems as traveling salesman problems," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1057–1061, Nov. 1997.

[17] R. M. Karp, *Reducibility Among Combinatorial Problems* (The IBM Research Symposia Series), R. E. Miller, J. W. Thatcher, and J. Bohlinger, eds. New York, NY, USA: Springer, 1972.

[18] R. Roberti, "Exact algorithms for different classes of vehicle routing problems," *4OR*, vol. 11, no. 2, pp. 195–196, 2013.

[19] D. Applegate, R. Bixby, V. Chvatal, and W. Cook. (Sep. 2014). *Concorde*. [Online]. Available: http://www.math.uwaterloo.ca/tsp/concorde/index.html

[20] G. Lawitzky, "A navigation system for cleaning robots," *Auto. Robots*, vol. 9, no. 3, pp. 255–260, Dec. 2000.

[21] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. D. Caro, "Human-friendly robot navigation in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 423–430.

[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[23] B. Gerkey. (Sep. 2014). *AMCL*. [Online]. Available: http://wiki.ros.org/amcl

[24] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2681–2684.

[25] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.

[26] J. Byrne and J. Shi, "Nested shape descriptors," in *Proc. ICCV*, Dec. 2013, pp. 1201–1208.

[27] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 510–517.

[28] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[29] F. A. Wichmann, J. Drewes, P. Rosas, and K. R. Gegenfurtner, "Animal detection in natural scenes: Critical features revisited," *J. Vis.*, vol. 10, no. 4, p. 6, Apr. 2010.

[30] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[32] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proc. Semantic Perception Mapping Exploration (SPME)*, 2013.

[33] R. Anati, D. Scaramuzza, K. G. Derpanis, and K. Daniilidis, "Robot localization using soft object detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4992–4999.

[34] A. Cohen, A. G. Schwing, and M. Pollefeys, "Efficient structured parsing of facades using dynamic programming," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3206–3213.

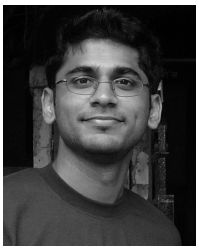
[35] R. Bellman, "The theory of dynamic programming," DTIC Document, Fort Belvoir, VA, USA, Tech. Rep., 1954.

[36] P. A. Titus and P. B. Everett, "The consumer retail search process: A conceptual model and research agenda," *J. Acad. Marketing Sci.*, vol. 23, no. 2, pp. 106–119, 1995.

[37] G. Varol and R. S. Kuzu, "Toward retail product recognition on grocery shelves," *Proc. SPIE*, vol. 9443, p. 944309, Mar. 2015.

Jonas Cleveland received the bachelor's degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, and the master's degree in robotics from the University of Pennsylvania, Philadelphia, PA, USA.

He is currently the Co-Founder of Cognitive Operational Systems, LLC (COSY), Philadelphia, a spin-off company out of GRASP Laboratory. His current research interests include computer perception and machine learning.



Dinesh Thakur received the M.S. degree in robotics from the University of Pennsylvania, Philadelphia, PA, USA, in 2011.

He is currently a Research Specialist with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania. His current research interests include multi-robot path planning and controls.

Philip Dames received the B.S. and M.S. degrees in mechanical engineering from Northwestern University, Evanston, IL, USA, in 2010, and the Ph.D. degree in mechanical engineering and applied mechanics from the University of Pennsylvania, Philadelphia, PA, USA, in 2015.

He is currently an Assistant Professor with the Mechanical Engineering Department, Temple University, Philadelphia. His current research interests include intersection of estimation, control, and communication in multiagent systems.

Cody Phillips received the B.S. and M.S. degrees in computer science from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, USA, with a focus on projective geometry and its applications to reconstruction and pose estimation of transparent surfaces of revolution.

He is also a Lead Developer for Cognitive Operational Systems, a spin-off company out of GRASP.

Terry Kientz received the B.S. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 1989. He is currently pursuing the M.S. degree in robotics from the University of Pennsylvania, Philadelphia, PA, USA.

He has advanced training as a Rehabilitation Engineer from the University of Virginia, Charlottesville, VA, USA, in 1996. He has been with the GRASP Laboratory, Mechanical Engineering Department, University of Pennsylvania, for over 15 years, where he is also a Project Engineer. His current research interests include engineering and fabrication from rehabilitation and medical device design to one of a kind prototype development.

Kostas Daniilidis (F'12) received the bachelor's 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 Hans-Hellmut Nagel.

He is a Professor of Computer and Information Science with the University of Pennsylvania, Philadelphia, PA, USA, where he has been on the faculty since 1998. He was the Director of the interdisciplinary GRASP Laboratory, Philadelphia, from 2008 to 2013 and is the Associate Dean for doctoral education of Penn Engineering, Philadelphia, since 2013. His current research interests include visual motion and navigation, active perception, 3-D object detection and localization, and semantic localization.

Dr. Daniilidis was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2003 to 2007. He founded the series of the IEEE Workshops on Omnidirectional Vision. In 2006, he co-chaired with Pollefeys the Third Symposium on 3-D Data Processing, Visualization, and Transmission, and he was Program Co-Chair of the 11th European Conference on Computer Vision in 2010.

John Bergstrom received the B.S. degree in civil engineering from the University of Illinois, Champaign, IL, USA, in 1980.

He was the Principal Scientist with Walgreens Boots Alliance, Deerfield, IL, USA. He is currently the Chief Innovation Officer with the Progredi Group.

Vijay Kumar received the B.Tech. degree from the Indian Institute of Technology Kanpur, Kanpur, India, and the Ph.D. degree from the Ohio State University, Columbus, OH, USA, in 1987.

He has been with the Department of Mechanical Engineering and Applied Mechanics, as a Faculty Member, since 1987, with secondary appointments with the Departments of Computer and Information Science and the Electrical and Systems Engineering. He was the Assistant Director for robotics and cyber physical systems at the White House Office of Science and Technology Policy, 2012 to 2014. He is currently the UPS Foundation Professor with the School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, USA. His current research interests include robotics, specifically multi-robot systems, and micro aerial vehicles.

Dr. Kumar was a Fellow of the American Society of Mechanical Engineers in 2003, and the Institution of Electrical and Electronic Engineers in 2005, and a member of the National Academy of Engineering in 2013.