An Automated System for Semantic Object Labeling with Soft Object Recognition and Dynamic Programming Segmentation

Jonas Cleveland,[1] Dinesh Thakur,[1] Philip Dames,[1] Cody Phillips,[1] Terry Kientz,[1] Kostas Daniilidis,[1] John Bergstrom,[2] and Vijay Kumar[1]

***Abstract*— This paper presents an automated system for generating a semantic map of inventory in a retail environment. Developing this map involves assigning a department label to each discrete section of shelving. We use a priori information to boost data from laser and camera sensors for object recognition and semantic labeling. We introduce a soft object map and a dynamic programming algorithm for point cloud segmentation. The primary contribution of this work is the integration of multiple systems including an automated path planning and navigation subsystem and a semantic mapping object recognition system. This work also represents an important contribution to robots working reliably in human environments. To our knowledge this is the first actual implementation of a fully automated robot inventory labeling system for a retail environment. The framework presented in this paper is easily scalable to other retail environments and is also relevant in any indoor environment with organized shelves, such as business storage facilities and hospital pharmacies.**

I. INTRODUCTION

One of the critical tasks in retail is to optimally manage the use of floor space within every store. In order to manage space correctly, especially when recommending future changes to space usage, one must have accurate knowledge of the way in which space is used at present, to monitor changes in space usage over time, and relate this usage to sales. In a retail chain like Walgreens, which has over 8000 stores in the United States, this knowledge is difficult and expensive to obtain. At present, given the location and size of each shelving fixture within the store, the staff of each store is expected to create a map showing: a) the departments (e.g. Diapers, First Aid, Deodorant) contained within the fixtures, and b) the linear space occupied by each department. Experience has shown that these maps can contain errors at the time of their creation, and that the ongoing process of revising the stores, due to seasonally fluctuating demand and to new product mixtures, can cause the errors to grow over time. The ability to autonomously and accurately determine department sizes and locations may produce significant benefit because it will free the staff to provide more customer care, reduce the costs associated with imperfect knowledge, and enable accurate optimization of store space allocations.

*We gratefully acknowledge support by Walgreens and by the NSF I/UCRC 1439681 grant.

[1]The authors are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, 19104, USA {jclev, tdinesh, pdames, codyp, tkientz, kostas, kumar}@seas.upenn.edu

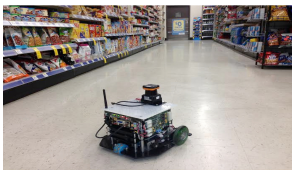
[2]John Bergstrom is with Walgreens, Deerfield, IL, 60015, USA john.bergstrom@walgreens.com

II. TECHNICAL BACKGROUND

Previous work presented in [1]–[7] discusses the creation of a semantic map by a robot in a human occupied environment. This corpus broadly defines a semantic map as the association of semantic information with a geo-position. While the household and academic building environments examined in previous work are important human working environments, these spaces provide innate advantages for an object recognition system that are not present in many other working environments. Typically in a home environment objects differ drastically in size, color, and shape characteristics [2]–[4], [6]. Furthermore, most of the systems deployed in these environments only handle recognition between a small number of classes. For instance in [1] their system is evaluated over a bicycle helmet, chair, and kitchen appliances. While these environments are described as cluttered [2]–[4], [6], there is still usually a clear line of sight to at least two faces on each object or a clear background corresponding to the object against which it is placed. 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 3D features are not discriminative. Objects of the same family are often similar in size and color characteristics. Furthermore, a retail environment often has over one thousand different types of products.

There has been a recent surge in research on robotic platforms in human domestic working environments. However, there is a surprisingly small amount of work in robotics in inventory storage environments. In [8], a detailed proposal is described for robots that automate the construction of planogram maps and handle other retail centric tasks such as merchandise management, visual merchandising, and inventory management. In [8]–[10], a system is described for mapping the physical structure of a retail environment. While previous work broadly references robotic technology for object-recognition and mapping neither describes in detail algorithms capable of multi-class object recognition or automated map creation.

We present an automated semantic labeling system that tackles some of the problems unique to this environment. We propose a path-planning algorithm that guarantees complete traversal of the space and constrains the robot motion such that the sensors capture all relevant areas of the environment during traversal. Our soft-object recognition allows potential objects to maintain multi-class labels which improves the



(a) Robot in a Walgreens store.

(b) Robot in the model store.

Fig. 1: Scarab robot in different stores.

final labeling output of the semantic map. Our system applies labels to individual objects which are used to classify larger physical-contextual regions. Regions are determined by segmentation over objects (each which are pre-assigned some region family). The segmentation step may then re-assign objects to a different label and class based on objects in close proximity. Thus our work describes computer vision and navigation algorithms for automated navigation and recognition that yields an annotated map without any external infrastructure or additional structure in the retail environment [8], [9]. Our work is an important contribution in that we describe the implementation of an automated object discovery, map management, and path planning system to tackle semantic labeling in a retail environment.

III. INFRASTRUCTURE

In this section we will 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 will 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 navigates within the store to collect vision data. 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 Sec. II, retail stores are cluttered, indoor environments. They often contain a series of parallel shelving units, at static locations, to hold the products. These shelving units must be at least five feet apart and are typically capped by an end-stand. Each shelving unit has multiple shelves arranged vertically. The store may also contain many transient objects, *e.g.*, boxes, shopping carts or baskets, and temporary displays. These transient objects may be 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 a standard width. All of the products within a vertical column of shelves all belong to the same department. This means that only a single shelf is required in order to determine the department labels. Certain products may be found within multiple departments, *e.g.*, cold medicine may be found within the medicine department as

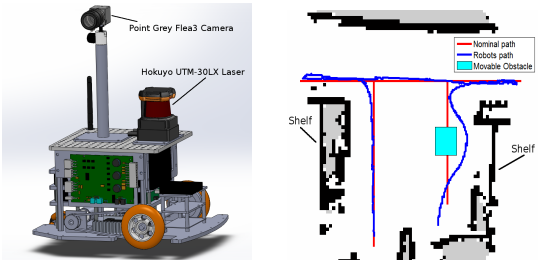


Fig. 2: (a) CAD drawing of the Scarab robot. (b) Example path of the robot avoiding an obstacle.

well as a seasonal flu and cold department. Fig. 1a shows our robot in a local Walgreens store.

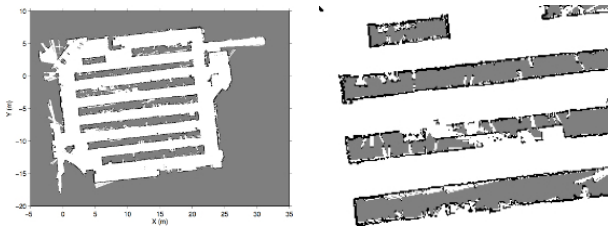
In order to perform realistic tests within a laboratory environment, we worked directly with Walgreens to design, build, and stock a model retail environment according to the company standards. Fig. 1b shows the robot in this model store. Each aisle is at least five feet long, with shelves on either side with enough depth for multiple products. Our model store contains six departments with similar dimensions to a Walgreens store. There are at least two departments on either side of the aisle, and over 60 different product brands that are found at Walgreens stores. The 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 of one square foot may appear in the environment, to emulate a box of products or a 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 [11], which is built in-house and shown in Fig. 2a. 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. Onboard processing for the navigation system is done using 2.4 GHz Intel Core i5 processor and 4 GB of RAM. On-board power is available through a pair of hot-swappable 14.4 V, 95 Wh LiPo batteries. The robot can also be directly plugged into the wall to charge.

The departments in most Walgreens stores are organized vertically, meaning that all of the shelves in a vertical column will contain the same type of products. Thus, the robot only needs to examine a single shelf in order to label the departments within a store. We mount the camera 46 cm from the ground plane to be able to detect products on bottom two



(a) Full store map. **(b)** Zoomed part of the map.

Fig. 3: Generated occupancy map of Walgreens store.

shelves. This increases the robustness of the system.

C. Docking Station and Processing Computer

Data from the robot is transferred to the docking station at the end of its automated run. The docking station parses video from the camera into time stamped frames synchronizes with position data. The computer is a 2.8 GHz Intel Core i7 processor with 16 GB of RAM. The sensor data collected is through USB3 to the robot, and then transferred online via 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 were deployed in Walgreens as product, processing would be completed on Walgreen's servers and uploaded to an interface for view at Walgreens corporate headquaters. Matlab scripts are used to format the data and product recognition is handled by functions written in C. The computer is a 2.9 GHz Intel Core i7 with 8 GB of RAM. The semantic map generated can be viewed in an image output

IV. SEMANTIC MAPPING

In this section, we first describe the navigation and planning algorithms necessary for the robot to successfully traverse cluttered retail environments, such as Walgreens stores, in order to collect product images. These images are stamped with the pose of the robot and uploaded to a server for processing. The system identifies the products within each individual image using a soft object detector. Next, using the position of the robot and the possible class labels of 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 to increase the precision of the object labeling system. This image processing and semantic map generation occur offline.

A. Navigation

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. 4, in order to be able to correctly detect and identify products using the camera.

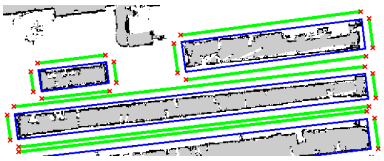


Fig. 4: Extracted shelves (blue boxes) and corresponding segments to be visited (green lines).

1) Localization: Initially, the robot is given an occupancy grid map of the environment, or such a map is manually created using, for example, the `gmapping` package from ROS [12]. An occupancy map generated from a Walgreens store is shown in Fig. 3a. The grey regions indicate unexplored areas while the white ones represent free space and black regions are occupied. The robot then uses the adaptive Monte Carlo localization algorithm [13] to track its position in the map. An implementation of this algorithm is available in the `amcl` ROS package [14].

2) Path Planning: To plan nominal paths through the environment, the robot first extracts the contours of the shelves from the occupancy grid map. The robot then creates a set of segments at some desired offset distance (2 ft) from each edge of the shelf contours, as shown in Fig. 4. Note that a typical shelf in a Walgreens store has four sides. Finally, the robot plans a distance-optimal path through the store such that it traverses every shelf segment.

Since the camera is mounted facing sideways with respect to the robot, each side of the shelf must be traversed in a particular direction (clockwise in our case). Each of these segments becomes a node in a graph and the connections between nodes are modeled as arcs, or directed links, since the distance from the end point 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. Thus, this planning problem is an Arc Routing Problem (ARP).

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. 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).

ARP solvers typically transform the problem to a node routing problem, also called Travelling Salesman Problems (TSP), as there are many readily available tools for TSP. Laporte [15] 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 Travelling Salesman Problem (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 3-node transformation proposed in [16]. 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 $u_i = u_{n+i} = u_{2n+i} = w_i, \forall w_i \in W$ [17]. 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 ∞.

Finally, we use the publicly available Concorde TSP solver [18], 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. The nominal path is then composed from this sequence by discretizing each segment into a series of waypoints.

3) Obstacle Avoidance: These waypoints are sequentially set as goals in the locally-reactive controller from Guzzi et al. [19]. The approach in [19] inflates all of the obstacles in the current laser scan and drives the robot toward the point in free space that is closest to the current goal. When the path in unobstructed the robot will drive straight towards 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.

B. Map Representation

This section describes our approach to creating a static semantic map. The static map 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. These objects and their physical characteristics are known a priori. Additionally, the departments and the products associated with each department are known a priori. Traditional semantic map approaches create a 3D point cloud from RGBD sensor input or the integration of laser data and camera data. A point cloud, M_p, consists of points $p_i, \cdots, p_n$, where $p_i \in \mathbb{R}^3$. Points are then grouped into segments, T_l, based on similar characteristics. The final output map is a group of labeled segments [20], [21].

Instead, we exploit the a priori information about the objects in the environment. We manage two concurrent map representations. The first is a traditional point cloud, M_p, generated from depth laser 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 the point cloud, a full scaled model of that object is represented in the virtual map and actively influences classification of regions in the point cloud. This virtual map provides several advantages. First, only information relevant to the robot's task is stored. Second, the object classification system is provided with an expectation of what the robot will view from multiple perspectives. Finally, space constraints are applied to accurately segment regions of nearby potential objects and improve the classification of neighboring regions.

C. Object Representation and Discovery

We use as a reference the notation and object representation described in [1]. The robot pose is given by

$$x = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{1}$$

Assuming a guassian noise λ the centroid of the potential object C^r in the robot frame is given by

$$C^r = RC + t + \lambda \tag{2}$$

where $C \in \mathbb{R}^3$ is the centroid of potential object in map frame. The object itself can be represented by a vector similar to

$$O = \{K, D, C\} \tag{3}$$

where K represents a set of key points and D the set of corresponding descriptors.

In [1], each centroid is associated with a confidence from the covariance matrix of the SLAM output. In order to evaluate the existence of an object, the keypoints in the region are matched to K_ω associated to a potential object $\omega \in \Omega$ pre-stored in the object template database.

Here we depart from [1] and describe our adoption of the soft-object representation described in [22]. Traditionally, in a set of potential objects Ω each ω belongs to a different class S. While we do maintain a library over a set of potential template objects $\omega \in \Omega$ as in [1], we do not assign one class to a potential object but rather a list of classes s with corresponding probability. Thus the object representation above is changed to

$$O_j = \begin{Bmatrix} P(S_1), K_1, D_1, C_j \\ . \\ . \\ . \\ P(S_s), K_s, D_s, C_j \end{Bmatrix} \tag{4}$$

where

$$\sum_s P(S_s) = 1. \tag{5}$$

In [1] the space is discretized into a number of regions using [20], [21]. Each region is classified as an object or surface using [20], [21]. The centroid C of the object corresponds to the centroid of the segment T. Rather we determine C by finding voting peaks for each template nearest-neighbor classifier. Previous work, such as [23], does well to argue the advantages of nearest neighbor classifiers over parametric approaches, such as SVMs.

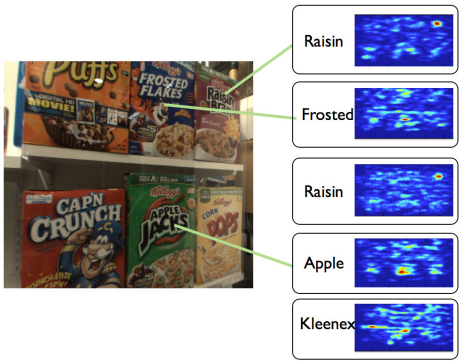


Fig. 5: A camera frame and its corresponding product heatmaps.

1) Template Object Library: In our application, we want to search for multiple classes within each individual image, rather than image matching or a one-class search. Each product in the store has at least one corresponding 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, for example a bag of chips. We extract descriptors d_k $(1 \leq k \leq K_\omega)$ from each product class S with dense SIFT keypoint search to generate a pre-stored object template database Ω. Since the number of keypoints generated can be different for each distinct class ω, the total number of descriptors, K_ω, is normalized across all product types by random selection.

2) Camera Measurement Step: For each video frame, we extract keypoints using a mesh grid centered in the image. We project each pixel to a plane based on laser depth data. We perform a nearest-neighbor search over each keypoint in the projected plane to determine its closest template neighbor. We employ a naïve Bayes nearest neighbor classifier [23] that minimizes

$$\prod_{i=1}^{N} ||d_i - NN_S(d_i) + Dist_{NN_S(d_i)}||^2 \tag{6}$$

In (6) $d_1, \cdots, d_N$ are the descriptors extracted from the current frame, $NN_S(d_i)$ is the nearest neighbor descriptor of d_i in class S [23] and $Dist_{NN_S(d_i)}$ is the trained distance for each descriptor. *Dist* 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

$$f(v_x^i, v_y^i) = \exp\left(-\frac{(x_w^i - v_x^i)^2}{2\sigma_x^{2_i}} - \frac{(y_w^i - v_y^i)^2}{2\sigma_y^{2_i}}\right) \tag{7}$$

$$H_S = \sum_i \frac{1}{||d_i - NN_S(d_i) + Dist_{NN_S(d_i)}||^2} \int_x \int_y f(v_x^i, v_y^i) dxdy \tag{8}$$

where $[x_w, y_w]^T$ the center of the Gaussian vote stamp, σ_x and σ_y are the window size of the Gaussian vote stamp, and $[v_x, v_y]^T$ is a vector from the template point to the center of the template itself.

This yields a voting table for each product, as seen in Fig. 5. This voting algorithm over all templates is bounded by $O(N^2Ks)$ complexity, where N is 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 $P(S_s) > \varepsilon$, 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. 6 shows.

Thus, per frame we can have multiple objects detected. These objects are transformed to the map frame according to the respective robot poses and the laser depth data and aggregated into a single map M_{vp}. For each frame measurement, if a detected object centroid is within some threshold distance to an object within the virtual map it will contribute

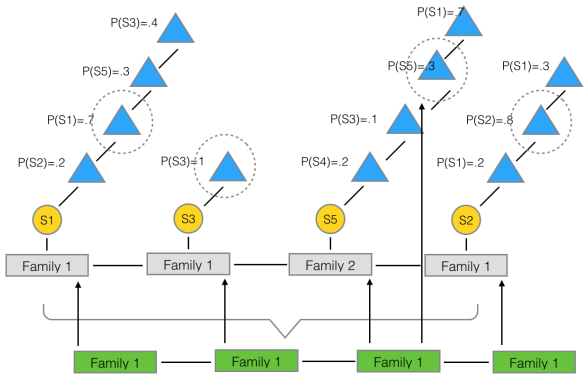


Fig. 6: Segmenting. Each object (yellow circle) can maintain a class label (blue triangle) determined by the class with the highest probability. Each of these objects will label the map with their mapped family (department). After the segmentation step an object's label might be changed based on the neighboring family labels. The label applied is the one with the highest probability belonging to the new family label.

to the formation of that virtual object and not initialize a new virtual object. In this process, the class probabilities are summed and normalized. The descriptors and key points are appended for each respective class. The centroid location is averaged.

D. Map Segmentation

Finally, 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: difference in intensity, color, motion, location, etc. 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 algorithm as a dynamic programming problem. Dynamic programing has been applied over images in several domains including noise filtering, edge detection, and contour segmentation. Most notably, in [24], dynamic programming is applied to parse the facade of a building. The approach in [24] initializes the segmentation process by first labeling each pixel based on a classifier

$$P(S_s) = \log mprob_s(S_s) - \log \Sigma_{S_\omega \in S_\Omega} mprob_s(S_\omega) \tag{9}$$

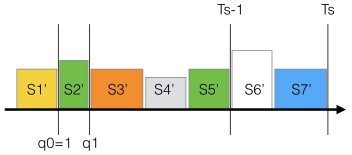


Fig. 7: Initializing Segmenting. Segmentation occurs over one dimension where each virtual object S is grouped into a region.

where $mprob_s$ 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. A most likely label is applied to each pixel and then structural information is used to further constrain the pixels.

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 in the map M_{vp} based on the maximum likelihood class S, using the size and shape of the object from the template library. Our segmentation algorithm runs over the virtual map and combines object recognition and region segmentation in the same step. The department boundaries are initialized using the boundaries of the virtual objects, $< q_x^{\max}, q_y^{\max}, q_x^{\min}, q_y^{\min} >$, as shown in Fig. 7. We then determine segments T composed of objects VO, where each T corresponds to a department in the Walgreens store and the objects are products or shelves. Recalling that each product type is associated with one or more department label, we use the labels of nearby objects to influence the final estimate of each object's class. If a virtual object 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 most likely class that can appear in T.

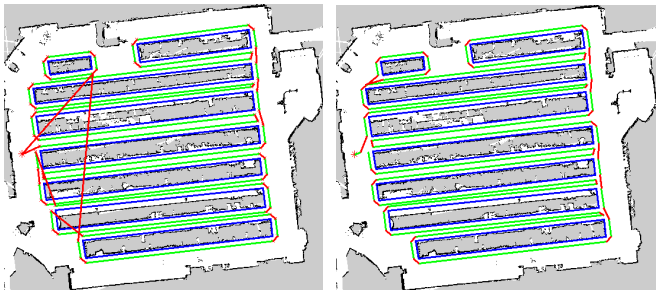
The dynamic programming problem is formulated as a segmentation of a one-dimensional signal, $q[0], q[1], \cdots q[T-1]$, into T_l segments, where each q is a boundary of the model object-class estimated to be located at that position [25]. For T_l departments there are $T_l - 1$ transitions $\{t_1, \cdots, t_{T_l-1}\}$.

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*** n **do**
  $opt(j,0) \leftarrow 0$
end
for $k \leftarrow 1$ ***to*** k_{max} **do**
  for $j \leftarrow 1$ ***to*** n **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
$optsoln \leftarrow \max_{1 \le k \le k_{max}} opt(n,k)$
$T_1 \ldots T_{k_{opt}} \leftarrow StandardDPArgMaxTraceback(opt, optsoln)*$
return $T_1 \ldots T_{k_{opt}}$

Algorithm 1: DPDEPARTMENTSEGMENTATION finds the optimal segmentation of a string of products into departments using dynamic programing (DP). For clarity, the algorithm above 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.



(a) Greedy solution. **(b)** Optimal solution.

Fig. 8: Planner output for greedy and optimal.

The i^{th} segment has probability density function (PDF)

$$p_i(q[t_{i-1}], \cdots, q[t_i - 1]) \tag{10}$$

With the assumption that each department is statistically independent, the PDF of the dataset is

$$\prod_{i=1}^{T_l} p_i(q[t_{i-1}], \cdots, q[t_i - 1]) \tag{11}$$

where $t_0 \equiv 0$, $t_{T_l} \equiv T$, and the MLE segmenter chooses $t_1, t_2, ..., t_{T_l-1}$ to maximize (10).

V. ANALYSIS

A. Evaluation of the Navigation System

We tested the optimal planner presented in Sec. IV-A against a greedy planner, which drives the robot to the nearest unvisited segment. Fig. 8a shows a path followed by robot for a greedy solution while Fig. 8b shows output of the optimal planner. The green lines show the arcs (shelf sides) that must be visited, the red lines show the connections betweens arcs. 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 travelled. Table I shows the path cost for greedy verses optimal planner for four trials with different starting locations of the robot.

B. Evaluation of Computer Vision Pipeline

Recent computer vision literature has seen an explosion of techniques in a race towards the perfect image feature descriptor. At a high-level, families of feature descriptors include those that are gradient-based, binarized color, and image patches [26]–[29]. Recent research is biased towards features with a compact descriptor length, such as FREAK, 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 multiple lighting conditions [26]–[29]. Since accuracy is our primary objective, we extract SIFT descriptors from a five by five pixel mesh grid across input images

TABLE I: Comparison of greedy vs optimal planner

Greedy cost (m)	Optimal cost (m)	difference (m)	% difference
315.58	287.75	27.83	9.22
316.21	287.92	28.29	9.36
322.38	296.60	25.78	8.32
329.67	287.51	42.16	13.66

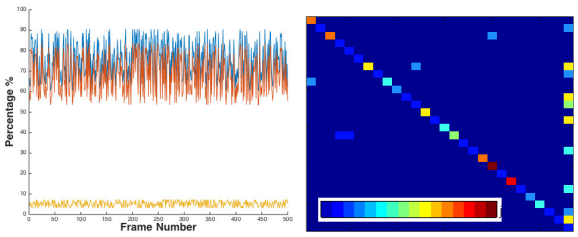


Fig. 9: (a) Precision(blue), Recall(red), and Localization error(yellow) as evaluated over a test dataset in a Walgreens store (b) A confusion matrix for the test datasets. The logo actually in the scene (rows) is plotted against the logo that is recognized (columns). The last row and column represent the background. The left-bottom bar shows the color key from min to max.

and training templates [30]. We first evaluate our recognition pipeline over a dataset taken at Walgreens on a mobile device camera with 2.2 mm focal length and 8 megapixel resolution. Fig. 9 summarizes the results of this initial study and is a confusion table evaluating the accuracy of the product recognition.

VI. Experiments

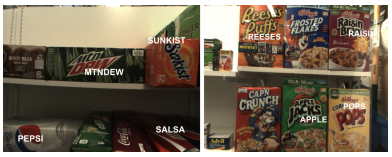
We conduct a series of experiments to test the ability of the robot to navigate a retail environment with natural clutter and to test the semantic labeling system.

First we tested our system over the model store mentioned and then we evaluated the ability to scale by testing it over an aisle in an actual Walgreens store. In each instance the movable object was placed in a random position. Fig. 2b shows the path of the robot in one of the test runs, where the robot moved closer to the shelf to avoid an obstacle which was not initially present in the map. Fig. 10a and Fig. 11a show two camera sequence measurements over which object detection is performed in the model and actual store. These measurements are then used to build the virtual object maps shown in Fig. 10b and Fig. 11b.

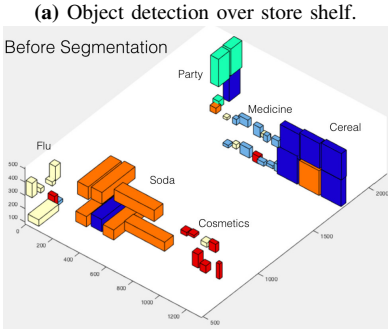
Our product recognition rate was highest for large objects. Small products, such as medicine and cosmetics, were more difficult. The robots bias towards the inside of the random obstacle benefited recognition performance – products appeared larger even if products from only one shelf were captured in the frame. Only a single row of products was placed on the shelves, but this is not an issue as the camera will typically only capture the first row of products on a shelf. The positions of products changed between runs, but product labels always faced outward from the shelf, as is typically seen in retail environments.

We use two metrics to evaluate performance. In order to evaluate the dimensions of each department, we use the model stores measurements as a ground truth and observed the physical distance error between the labeled sections in the map and their actual dimension in the model store given in Table II. In order to evaluate that departments were correctly positioned in the generated map (Fig. 10c), we evaluate the percentage of shelf pixels correctly labeled in the output image map as shown in Table III.

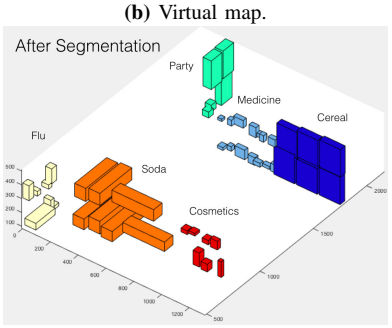
In each of our trials, the order of the departments on the shelves is successfully determined. Furthermore, the



(a) Object detection over store shelf.



(b) Virtual map.



(c) Segmented map.

Fig. 10: Semantic mapping process for model store.

average measurement error across departments is relatively small when considering the application. A retailer will have sufficient information to determine the reorganization of items and departments. The system is successfully able to traverse small objects and products covered by glass are correctly classified. A brief analysis of the map error shows that it suffered on departments of short length and worst on departments on the end stands. This error has multiple sources. Departments of the shortest length contain the smallest items. Departments placed on an end stand are captured while the robot is turning and the camera is not fully orthogonal to the shelf. This decreases the rate of classification.

VII. Conclusions

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 system is able to reliably navigate thoughout the environment and detect products from images acquired by onboard cameras,

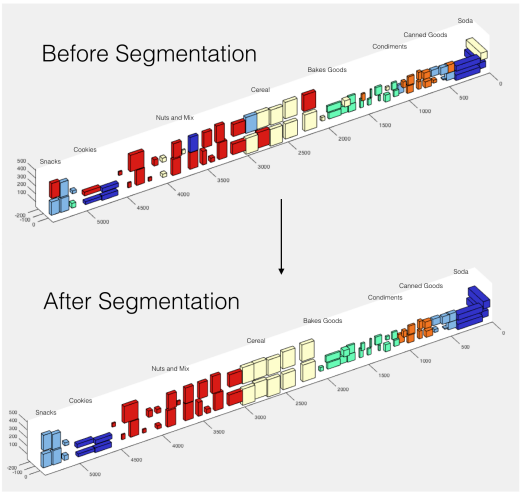
TABLE II: Semantic map performance evaluation

Physical distance error	12.25%, 13.3175 cm
Classification accuracy	86.84%

TABLE III: Measurement errors for departments

Cereal	Medicine	Superbowl	Skin care	Flu	Soda
7.65%	13.51%	20.43%	35%	31.89%	22.15%

(a) Object detection over store shelf.



(b) Automated semantic map generated over an actual Walgreens store aisle. We pre-select which Virtual Objects to tabulate the database with according to their frequency and recurrence across all Walgreens stores.

Fig. 11: Semantic mapping process for an aisle in Walgreens store.

use the products to determine the most likely department in the image, and labels the corresponding portion of the map with the appropriate department label. The environment navigation is done in such a way that it minimizes the total distance travelled by the robot while guaranteeing that no shelf is visited more than once.The product recognition is done by combining the performance of weak classifiers over associated objects. The system is able to correctly and accurately label a model store containing six departments and over sixty product types.

Our current work is direct toward scaling up the system in a typical Walgreens store that might shelve hundreds of product types. The main challenges includes scaling up to over 1,000 product types and 120 department labels. In order to improve the robustness of the labeling system, we plan to use additional visual information such as text and barcodes.

REFERENCES

[1] S. Choudhary, A. J. Trevor, H. I. Christensen, and F. Dellaert, "Slam with object discovery, modeling and mapping," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1018–1025.

[2] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Ruhr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 4263–4270.

[3] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in Neural Information Processing Systems*, 2011, pp. 244–252.

[4] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Ruhr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 4263–4270.

[5] J. St"uckler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from rgb-d video," *Journal of Real-Time Image Processing*, pp. 1–11, 2013.

[6] R. B. Rusu, *Semantic 3D object maps for everyday robot manipulation*. Springer Publishing Company, Incorporated, 2013.

[7] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz, "Semantic object maps for robotic housework-representation, acquisition and use," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4644–4651.

[8] K. Mankodiya, R. Gandhi, and P. Narasimhan, "Challenges and opportunities for embedded computing in retail environments," in *Sensor Systems and Software*. Springer, 2012, pp. 121–136.

[9] S. Kumar, G. Sharma, N. Kejriwal, S. Jain, M. Kamra, B. Singh, and V. K. Chauhan, "Remote retail monitoring and stock assessment using mobile robots."

[10] E. Frontoni, M. Contigiani, and G. Ribighini, "A heuristic approach to evaluate occurrences of products for the planogram maintenance," in *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*. IEEE, 2014, pp. 1–6.

[11] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multirobot teams," *Robotics Automation Magazine, IEEE*, vol. 15, no. 1, pp. 53–61, March 2008.

[12] B. Gerkey. (2014, Sept.) gmapping. [Online]. Available: http://wiki.ros.org/gmapping

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[14] B. Gerkey. (2014, Sept.) amcl. [Online]. Available: http://wiki.ros.org/amcl

[15] G. Laporte, "Modeling and solving several classes of arc routing problems as traveling salesman problems," *Computers & Operations Research*, vol. 24, no. 11, pp. 1057–1061, 1997.

[16] R. M. Karp, *Reducibility among Combinatorial Problems*, ser. The IBM Research Symposia Series, R. Miller, J. Thatcher, and J. Bohlinger, Eds. Springer US, 1972.

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

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

[19] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. D. Caro, "Human-friendly Robot Navigation in Dynamic Environments," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 423–430.

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

[21] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[22] R. Anati, D. Scaramuzza, K. G. Derpanis, and K. Daniilidis, "Robot localization using soft object detection," pp. 4992–4999, 2012.

[23] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[24] A. Cohen, A. G. Schwing, and M. Pollefeys, "Efficient structured parsing of facades using dynamic programming," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3206–3213.

[25] R. Bellman, "The theory of dynamic programming," DTIC Document, Tech. Rep., 1954.

[26] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2681–2684.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[28] B. Jeffrey and J. Shi, "Nested shape descriptors," *ICCV*, 2013.

[29] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee, 2012, pp. 510–517.

[30] F. A. Wichmann, J. Drewes, P. Rosas, and K. R. Gegenfurtner, "Animal detection in natural scenes: critical features revisited," *Journal of Vision*, vol. 10, no. 4, p. 6, 2010.