robotics

MDPI

Article

A Robot Architecture Using ContextSLAM to Find Products in Unknown Crowded Retail Environments

Daniel Dworakowski *, Christopher Thompson, Michael Pham-Hung and Goldie Nejat *

Autonomous Systems and Biomechatronics Laboratory (ASBLab), Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada; chriss.thompson@mail.utoronto.ca (C.T.); michael.pham.hung@mail.utoronto.ca (M.P.-H.)

* Correspondence: daniel.dworakowski@mail.utoronto.ca (D.D.); nejat@mie.utoronto.ca (G.N.)

Abstract: Grocery shoppers must negotiate cluttered, crowded, and complex store layouts containing a vast variety of products to make their intended purchases. This complexity may prevent even experienced shoppers from finding their grocery items, consuming a lot of their time and resulting in monetary loss for the store. To address these issues, we present a generic grocery robot architecture for the autonomous search and localization of products in crowded dynamic unknown grocery store environments using a unique context Simultaneous Localization and Mapping (contextSLAM) method. The contextSLAM method uniquely creates contextually rich maps through the online fusion of optical character recognition and occupancy grid information to locate products and aid in robot localization in an environment. The novelty of our robot architecture is in its ability to intelligently use geometric and contextual information within the context map to direct robot exploration in order to localize products in unknown environments in the presence of dynamic people. Extensive experiments were conducted with a mobile robot to validate the overall architecture and contextSLAM, including in a real grocery store. The results of the experiments showed that our architecture was capable of searching for and localizing all products in various grocery lists in different unknown environments.

Keywords: service robots; grocery robot architecture; crowded unknown environments; context identification; context mapping

check for updates

Citation: Dworakowski, D.; Thompson, C.; Pham-Hung, M.; Nejat, G. A Robot Architecture Using ContextSLAM to Find Products in Unknown Crowded Retail Environments. *Robotics* **2021**, *10*, 110. https://doi.org/10.3390/robotics10040110

Academic Editors: Giuseppe Carbone and Alessandro Di Nuovo

Received: 10 July 2021
Accepted: 20 September 2021
Published: 26 September 2021

1. Introduction

Finding products in grocery stores is a challenging task requiring an understanding of a store's layout, finding these products based on this understanding and negotiating crowded aisles. In general, customers have difficulty remembering the locations of products in aisles due to a lack of location cues and mnemonic aids [1]. Moreover, the layout of a store changes regularly due to several factors to improve sales such as: (1) seasons, and holidays; (2) periodic changes due to shopper behavior [2]; (3) product promotions, and (4) the daily introduction of new products [3]. The estimated 21,000 products [4] introduced yearly have a significant impact on store layouts due to the rearrangement of other products [5]. Therefore, shopping can be stressful for customers in such environments [6]. Even regular customers leave without buying all their desired items [7]. The inefficiency of the shopping experience reduces customer satisfaction [8] and the inability to make intended purchases represents significant lost revenue for stores [9].

It has been found that customer satisfaction is directly related to perceived employee knowledge [10]. Employee availability and lack of employee knowledge are exacerbated by turnover rates of grocery employees; a major contributor to the general decline in grocery store satisfaction [11].

Systems using multiple ceiling mounted cameras require modifying and incorporating sensors in the entire store to aid with product localization and server fees for data processing, which can be costly and invades customer privacy [12]. Existing smartphone apps rely on planograms of a store, that are manually created by staff, which is time-consuming and can contain product location errors. Compliance with planograms has been found to be less than 60% [13]. Furthermore, these apps are not widely accessible to all demographics including seniors (only 20% of seniors have access to a smartphone [14]). Our system aims to provide an autonomous robot for finding products on shelves within a grocery store without altering the environment and with minimal change to the shopping experience. Our robot has onboard sensing and computation capabilities and can be placed in any new environment to work immediately without prior knowledge, unlike existing camera systems that need to be installed and calibrated in each environment. Using robots to automate one-on-one customer assistance would aid in reducing customer stress thereby improving their shopping experience [6].

Service robots have been used to automate numerous tasks in human-centered environments such as searching for people using a preexisting map and their activity schedule to provide assistance [15], helping with cleaning/delivery [16,17], and surveillance [18]. With respect to grocery stores, robots have several advantages as they: (1) allow employees to perform complex tasks that cannot be easily automated (e.g., stocking shelves, rearranging products to improve revenue), and (2) lower operating costs by reducing employee turnover and salary requirements (e.g., the cost to replace an employee is between 75–150% of the position's salary [19]). They have already been used in retail stores to improve overall customer experience. This includes showing product locations, and carrying products for customers [20,21]. Studies have shown that service robots in grocery stores have high satisfaction and intent to use rates. For example, in [21], 92% of customers would use a robot again as a shopping companion. However, such robots require full prior knowledge of their environment and cannot behave autonomously without human input to generate/obtain a representation of the environment [20–22]. Their reliance on expert human knowledge of the environment prevents these robots from autonomously adapting to regular changes to a store layout or product locations, and prevents their use in multiple stores. Given that there are approximately 38,000 grocery stores in the U.S. alone [23], regular manual mapping due to layout changes would represent a significant time cost for stores (e.g., it can take at least 75 min to manually map a grocery-like environment with 29 aisles with a robot moving at 0.5 m/s [21]).

To autonomously handle new layout changes, a robot must generate/update a map and identify vital landmarks in the environment, while simultaneously localizing itself in the map. This is known as the Simultaneous Localization and Mapping (SLAM) problem [24,25]. Current trends in SLAM focus on data association techniques to fuse sensor readings, dealing with uncertainty, or decreasing time complexity [26–30]. In a grocery store environment, SLAM strategies need to take advantage of the texture rich features that are available to aid in finding products in crowded aisles containing many products and dynamic people. The performance of SLAM methods relies heavily on the ability to associate previously seen unique features between consecutive sensing frames to build the environment map and define the robot's path within this map [30,31]. As such, the choice of features and their repeatability impact the method's performance. Furthermore, annotating the map with contextual information is vital to the decision making process for mobile robots and enhances a robot's ability to interact with the environment [32]. For example, planning an optimal path to products of interest in a crowded store.

This paper presents the development of a generic grocery robot architecture for the autonomous search and localization of products in unknown dynamic (with unpredictable people) grocery stores. A unique context Simultaneous Localization and Mapping (contextSLAM) framework is developed that combines an Optical Character Recognition (OCR) system with laser range measurements using a Rao-Blackwellized particle filter to simultaneously generate an annotated map and localize a robot to address the grocery store product search problem. contextSLAM differs from existing methods in that: (1) it is the first to simultaneously combine contextual landmarks and occupancy grid information for online generation of an annotated map and robot localization, (2) it can uniquely update the context map to accommodate regularly changing grocery store layouts, and (3) it accounts for non-unique feature instances and rejects dynamic objects/people, while using landmarks pervasive in human environments such as text. The framework uses contextual information to aid robots with exploration and localization in dynamic environments, without the need to introduce artificial landmarks or sensors into the scene. Contextual information includes symbols on signs, objects, and posters that provide information for the localization of the robot and products of interest. It is the first work inspired by human exploration that considers common environmental cues to not only reason about the content of the environment, but also to aid in finding the item locations of interest. The advantage of considering scene text is that it is readily available in retail environments and thus can be incorporated as a robust source of localization information for robots.

2. Related Work

Herein, literature on: (1) retail robots, and (2) use of contextual evidence for mapping and robot localization is discussed.

2.1. Retail Robots

Robots have been designed to aid in a variety of tasks in stores including: (1) inventory management, or (2) assisting customers.

2.1.1. Inventory Management

Several systems have been designed for the automation of inventory management [22,33]. For example, in [22], a robot was used to create a semantic map of a retail environment. An operator manually moved the robot through the environment as GMapping generated a 2D occupancy grid map. Then a dynamic programming algorithm segmented the map into categories (e.g., oats, cans, etc.) based on the products observed while the robot autonomously scanned shelves to create a semantic map of product locations.

In [33], a robot was designed for autonomous oos detection, spacing, inventory compliance, and facing checks of products. An operator manually moved the robot through the environment to construct a 2D occupancy grid using GMapping. The robot then navigated identified aisles while taking depth and 2D images to detect shelves and the spacing between products.

2.1.2. Customer Service

Robots have been proposed for customer service tasks, such as finding products [34], and escorting people through a store [20,21]. In [20], a robot guided a user to desired products using an annotated map of product locations while a second carried their products. Localization and path planning were achieved using onboard laser scanners and external cameras in the store. In [21], a robot provided product location guidance, shopping companionship, and price checks. A teleoperated robot collected laser and sonar data for Map-Match SLAM [35] to generate a map. The map was labelled using data from the store's internal map via a manual transform.

Our previous workshop paper [34] introduced an autonomous robot to guide users to desired products in a static unknown environment. While exploring the environment, a map was created using GMapping and annotated with text detected in the environment. Human-robot interaction experiments in a lab environment showed that participants found the robot helpful, motivating our current research.

While some of the aforementioned robots were able to provide customer assistance, the majority require a priori knowledge of the store layout or product locations [21,22,33], or external sensors in the environment [20]. Therefore, existing architectures cannot be directly applied to our problem, as there is no available/limited prior information to exploit in the search process as we consider scenarios where: (1) the robot is deployed for the first time in a new store(s), or (2) a store layout has changed. We address these limitations using a novel autonomous robot architecture for grocery stores that allows a robot to explore unknown crowded dynamic environments to find products of interest without any prior knowledge of product locations or layout of an environment. This work extends our preliminary research by developing a robust online approach using: (1) context detection via deep learning-based OCR, and (2) a fused mapping approach merging context and laser range data via our novel context Simultaneous Localization and Mapping (contextSLAM) method.

2.2. Mapping and Localization Using Contextual Information

Contextual features in the environment have been used for a variety of purposes, including: (1) localization [36–38], (2) map annotation [21,22,39–43], (3) SLAM using text [44,45], and (4) Semantic SLAM [32,46–49].

2.2.1. Localization

In general, the types of contextual features used for localization have included unique fiducial markers [36,38], or visual or range features [37]. In [36], a method used laser range and unique April tag based visual context measurements in a graph SLAM framework. Robot localization was then performed using only the April tags. In [37], a SLAM method used a reinforcement learning policy to switch between localizing with an occupancy grid or landmark map. The method maintained separate particle weights for both maps. The policy selected which map to use based on a handcrafted state representation. In [38], a method used color-based segmentation to extract regions of text from unique name plates in the environment to associate the strings to nodes in a topological map.

2.2.2. Map Annotation

Context, and feature-annotated maps have been used for SLAM purposes such as in [21,22,39–43]. In [39], a method for combining and correcting occupancy grids using feature measurements was presented. The features (e.g., object locations) were used for localization and to partition the environment into triangular regions, with local occupancy grids defined relative to these features. In [40], a modified tinySLAM method was presented. The tinySLAM method fused laser data, odometry, and RFID tags detected in the environment to create an occupancy grid.

Both [42,43] proposed methods to generate annotated occupancy grids of environments. In [43], an OCR system annotated a pre-existing occupancy grid with the text on door signs. Alternatively, in [42] an existing map of features was transformed to fit into an occupancy grid as it was generated using robot observations during navigation.

2.2.3. SLAM Using Text Features

In [44], OCR was used in SLAM by fusing measurements to the centroids of unique text instances with visual inertial odometry via incremental Smoothing and Mapping. In [45], SLAM was performed using the planar features of text instances in the environment by minimizing the projected photometric error of detected text boxes via bundle adjustment.

2.2.4. Semantic SLAM

More recently, SLAM methods have been detecting and classifying objects in the environment to semantically segment and label maps [32,46–49]. In [32], Mask-RCNN was used for the image segmentation task and combined with an RTAB-Map SLAM algorithm to generate semantic point clouds of the environment. In [46], a method that combined semantic segmentation information from PSPNet with 3D Point cloud data from ORB-SLAM was proposed to create a 3D semantic map and in [47], a SLAM method combining planar surfaces of semantically detected objects and visual inertial odometry was proposed.

In both [48,49], dynamic objects were considered. Namely, in [48], a SLAM approach used surfel-based mapping and semantic labels to filter out dynamic obstacles from 3D point cloud readings. The semantic segmentation resulted in point-wise labels for each point in the point cloud. In [49], optical flow was combined with MASK-RCNN SLAM to filter dynamic feature points. The Mask-RCNN network was used to detect and mask potentially moving objects.

The methods described above have shown that contextual information can be used for various robotic tasks. However, none of the methods can be used to solve our grocery store search problem. In particular, some methods require an existing map of the environment [21,22,42,43] or training on an existing map [37] prior to the creation of a context map. Others represent their occupancy grid based on the locations of distinct features [39]. If features are occluded, error accumulation occurs, resulting in an inconsistent map. Furthermore, some methods require the environment to be modified with artificial landmarks (i.e., April or RFID tags) [36,40], or require unique landmarks [38,44]. Lastly, it is infeasible to create an object detector for all items within a grocery store environment, as the number of classes required for a generic classifier may be on the order of 10^5 [50,51]. As a result, semantic SLAM methods, which inherently use object classifiers, are limited in the number of products that they can accurately identify, which limits the robot's ability to search for and navigate to a variety of different products [32,46–49]. Using text, on the other hand, results in a more accurate classification of products since all products in the environment have a text label, either on the shelf or the packaging themselves. In [45], text features were used for SLAM, however, only the planar surfaces of the detected text were utilized and the method did not annotate the map with the contextual information of the text strings found [45]. Therefore, these approaches cannot be used for our grocery search problem as: (1) real grocery environments contain repeated features (e.g., same text on signage, shelves, posters, etc.), and (2) introducing artificial landmarks requires each environment to be modified prior to the robot being used in the store.

In this paper, we propose a new online approach, contextSLAM, that simultaneously utilizes both the occupancy grid and observed non-unique context in the environment obtained via OCR to localize the robot and generate a contextually rich map. Our approach allows for online intelligent robot exploration of the environment by using the context map during navigation. Furthermore, we make use of scene text that is pervasive within a grocery store environment without the limitation that the contextual information be unique.

3. Grocery Robot System Architecture

The proposed grocery product search problem requires a mobile robot to search for and locate various products in an unknown environment which may contain other independent agents (e.g., people). The store contains a set of products $\mathbb{P}_{\text{gr}}$. The task begins when a user provides a subset of products to locate, $\mathcal{Q} \subseteq \mathbb{P}_{\text{gr}}$. A product is considered to be located when the robot detects it on a shelf.

The grocery environment contains an open area where the robot's home location is. Aisles exist perpendicular to the edges of this open area and are formed by shelving units containing products on both sides. Aisles have openings at both ends and have a traversable width, w_a, and minimum length l_{min}. Aisle signs are placed visibly within the aisle and display contextual information. Static text landmarks (e.g., on aisle signs) are present throughout the environment. Unique fiducial markers are placed beside products of interest on shelves, as the product identification problem is beyond the scope of this paper.

3.1. Architecture Overview

For a robot to autonomously search a grocery store for user desired products, we have developed the grocery robot architecture in Figure 1. A search is initiated by a user providing a search query. This triggers the *Explore* state within the *Action Deliberation* module that uses the *Frontier Detection* module to implement frontier-based exploration. Navigation goals are sent to the *Navigation System* which provides motion commands to the *Low-Level Controller*. Frontiers are determined using the map generated by the *Context Mapping* module. This map is a fusion of data from the *Context Identification* and *Obstacle Detection* modules and *Odometry*. Parallel to exploration, the *Aisle Detection* module finds aisles in the environment. When there is evidence of the presence of a product in the search query in an aisle, the *Action Deliberation* module transitions to the *Aisle Found* state and the *Navigation System* localizes the robot in front of the aisle. The *Aisle Search* state allows the robot to navigate a crowded aisle. The *Product Detection* module is used to detect products within images from the onboard *Camera.* When a product in the search query is found the user is notified. The search continues until all products are found, no frontiers remain, or the maximum number of search attempts has been reached. The following subsections detail the main modules of the architecture.

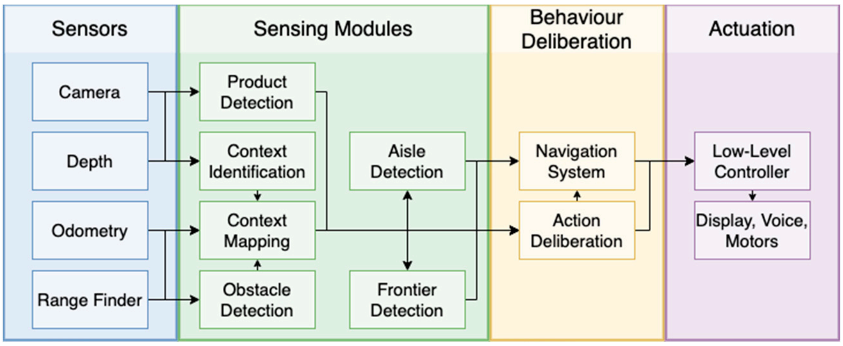


Figure 1. Grocery robot architecture.

3.2. Context Identification

The *Context Identification* module uses 2D images of the environment to identify contextual information to be used to find the locations of products in $\mathcal{Q}$. It utilizes OCR to identify text that is pervasive within grocery stores (e.g., on aisle signs).

The product function, $P(\mathcal{S})$, is used to determine whether a set of text indicates the presence of a product, where:

$$P(\mathcal{S}) = \{p_{\text{gr}} | p_{\text{gr}} \in \mathbb{P}_{\text{gr}}, \; \Pr(p_{\text{gr}} | \mathcal{S}) > \phi_{min}\}, \tag{1}$$

and ϕ_{min} is a probability threshold. $\mathcal{S}$ is a set of strings and $\Pr(p_{\text{gr}}|\mathcal{S})$ is the probability that a product p_{gr} is present given a set of strings $\mathcal{S}$. Aisle signs have the property that the context detected on the signs s_a is guaranteed to generate a non-empty set, i.e., $|P(\{s_a\})| > 0$, based on this assumption we set the threshold ϕ_{min} to 1 for our experiments in Section 5. In general, the threshold is a user-defined value and is set based on the desired precision and recall of the probabilistic model representing the association of scene text to the presence of products.

Our grocery OCR system combines convolution neural network detectors and trains them to find text in images of the environment. It begins with a RetinaNet single stage text object detector [52] with a ResNeXt-50 [53] backbone, which predicts which regions in an image contain text. Region proposals (formed by a quadrilateral) are placed into standard size containers using a homography transform. Text strings in the regions are identified with a character region neural network [54] and filtered using a dictionary. Detections are formed by a tuple containing the text strings, s, and the relative 3D world coordinate, p_{xyz}, of each region's center. The 2D images are obtained from an RGB-D Camera, such that each pixel is associated with a depth measurement. The world coordinate is found by sampling a concurrently captured point cloud region as the text. The observation set, $o = \{(s_i, p_{xyz,i})\}$, is provided to the *Context Mapping* module.

3.3. Obstacle Detection

Grocery shoppers make unpredictable stops and change directions to complete their shopping goals. Dynamic people can introduce false obstacles within maps of the environment, which can lead to map misalignments and localization errors [55]. These map errors can prevent planners from finding valid navigation plans through an environment to goal locations. To detect people we adapted the leg detection method in [56] and incorporated a per laser beam score with respect to each beam intersecting a leg. Beams that are observing dynamic obstacles are clustered based on distance with a minimum of three members in a cluster, Figure 2. A random forest assigns a confidence to each cluster indicating the likelihood of containing a leg. A higher confidence is given to cluster members closer to the cluster center than the edges. The weighted scan is provided to the *Context Mapping* module to prevent the addition of dynamic obstacles to the map.

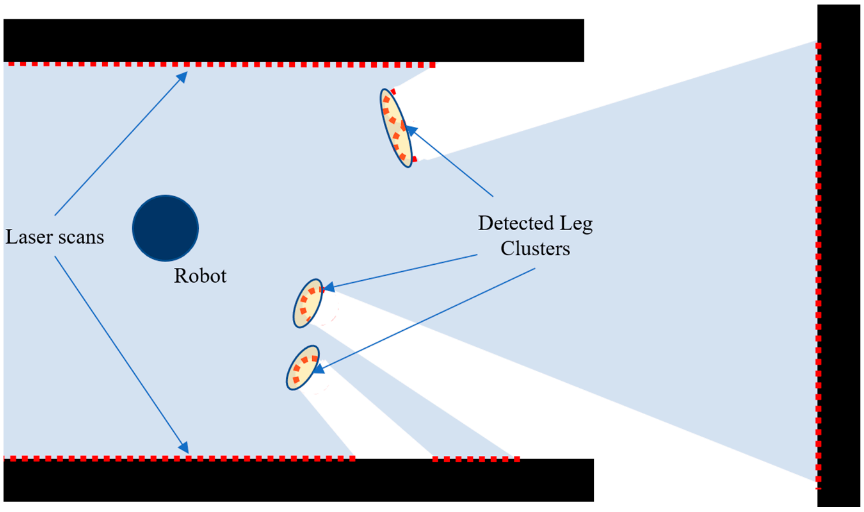


Figure 2. Laser scans (red dotted lines) are clustered based on distance and assigned a confidence weight (yellow ellipsoids). Black rectangles are static objects.

3.4. *Context Mapping*

The *Context Mapping* module uses the novel contextSLAM approach we have developed for the online creation of an annotated occupancy grid map of the environment. ContextSLAM incorporates sparse, non-unique, landmarks represented as environment text and laser scan information provided by the Obstacle Detection module within a Rao-Blackwellized particle filter (RBPF). The landmarks are used concurrently with the annotated context map to localize the robot in the environment. We expand the Rao-Blackwellized factorization in [57] to incorporate context observations $o_{1:t} = [o_1, \ldots, o_t]$. This is achieved by introducing a new probabilistic model, and particle weights and robot pose estimation representations. The joint distribution of the context, context map $\mathbb{M}$, robot state trajectory $x_{1:t} = [x_1, \ldots, x_t]$, 2D laser scan measurements $z_{1:t} = [z_1, \ldots, z_t]$, and odometry $y_{1:t-1} = [y_1, \ldots, y_{t-1}]$ is:

$$\Pr(x_{1:t}, \mathbb{M} | o_{1:t}, z_{1:t}, y_{1:t-1}) = \Pr(\mathbb{M} | x_{1:t}, o_{1:t}, z_{1:t}) \Pr(x_{1:t} | o_{1:t}, z_{1:t}, y_{1:t-1}), \quad (2)$$

where $\Pr(\mathbb{M}|x_{1:t}, o_{1:t}, z_{1:t})$ represents the probability of the current context map given the robot trajectory, measurements and odometry. $\Pr(x_{1:t}|o_{1:t}, z_{1:t}, y_{1:t-1})$ represents the probability of the trajectory given the measurements, odometry, and context. We obtain these probabilities using a particle filter.

Particle weights within the particle filter, $W_t^{(i)}$, indexed by i, are calculated based on the following relationship:

$$W_t^{(i)} = \Pr\left(x_{1:t}^{(i)} \middle| o_{1:t}, z_{1:t}, y_{1:t-1}\right) / \pi\left(x_{1:t}^{(i)} \middle| o_{1:t}, z_{1:t}, y_{1:t-1}\right), \quad (3)$$

where $\Pr\left(x_{1:t}^{(i)} \middle| o_{1:t}, z_{1:t}, y_{1:t-1}\right)$ is the posterior over the potential trajectories, and $\pi\left(x_{1:t}^{(i)} \middle| o_{1:t}, z_{1:t}, y_{1:t-1}\right)$ is the proposal distribution. We obtain a recursive formulation for the weights by constraining the distributions of the current trajectory to be a product of the previous likelihoods:

$$\begin{aligned} W_t^{(i)} &= \frac{\Pr\left(x_t^{(i)}\middle|x_{1:t-1}^{(i)}, o_{1:t}, z_{1:t}, y_{1:t-1}\right)\Pr\left(x_{1:t-1}^{(i)}\middle|o_{1:t-1}, z_{1:t-1}, y_{1:t-2}\right)}{\pi\left(x_t^{(i)}\middle|x_{1:t-1}^{(i)}, o_{1:t}, z_{1:t}, y_{1:t-1}\right)\pi\left(x_{1:t-1}^{(i)}\middle|o_{1:t-1}, z_{1:t-1}, y_{1:t-2}\right)} \\ &= \frac{\Pr\left(x_t^{(i)}\middle|x_{1:t-1}^{(i)}, o_{1:t}, z_{1:t}, y_{1:t-1}\right)}{\pi\left(x_t^{(i)}\middle|x_{1:t-1}^{(i)}, o_{1:t}, z_{1:t}, y_{1:t-1}\right)} W_{t-1}^{(i)}. \end{aligned} \tag{4}$$

Furthermore, we can incorporate the previous context, $o_{1:t-1}$, and laser scan measurements, $z_{1:t-1}$, into the previous map estimate, $\mathbb{M}_{t-1}^{(i)}$. Given this decomposition, the one time-step of the proposal distribution incorporating the context and laser scan measurements is defined as: $\pi\left(x_t^{(i)}\middle|x_{1:t-1}^{(i)}, o_{1:t}, z_{1:t}, y_{1:t-1}\right) = \Pr\left(x_t^{(i)}\middle|\mathbb{M}_{t-1}^{(i)}, x_{t-1}^{(i)}, o_t, z_t, y_{t-1}\right)$.

Using Bayes' rule, the proposal distribution is:

$$\Pr\left(x_t^{(i)}\middle|\mathbb{M}_{t-1}^{(i)}, x_{t-1}^{(i)}, o_t, z_t, y_{t-1}\right) = \frac{\Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x_t^{(i)}\right)\Pr\left(x_t^{(i)}\middle|x_{t-1}^{(i)}, y_{t-1}\right)}{\int \Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x\prime\right)\Pr\left(x\prime\middle|x_{t-1}^{(i)}, y_{t-1}\right)dx\prime}. \tag{5}$$

By substituting the proposal distribution Equation (5) into Equation (4), and applying Bayes' Rule to the posterior distribution, we then obtain the following weight update:

$$\begin{aligned} W_t^{(i)} &= \frac{\eta\Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x_t^{(i)}\right)\Pr\left(x_t^{(i)}\middle|x_{t-1}^{(i)}, y_{t-1}\right.}{\frac{\Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x_t^{(i)}\right)\Pr\left(x_t^{(i)}\middle|x_{t-1}^{(i)}, y_{t-1}\right)}{\int \Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x'\right)\Pr\left(x'\middle|x_{t-1}^{(i)}, y_{t-1}\right)dx'}} W_{t-1}^{(i)} \\ &= W_{t-1}^{(i)}\eta\cdot \int \Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x'\right)\Pr\left(x'\middle|x_{t-1}^{(i)}, y_t\right)dx' \\ &\propto W_{t-1}^{(i)}\cdot \int \Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x'\right)\Pr\left(x'\middle|x_{t-1}^{(i)}, y_t\right)dx', \end{aligned} \tag{6}$$

where η= $1/\Pr(z_t, o_t|z_{1:t-1}, o_{1:t-1}, y_{1:t-1})$ is constant for all weights and is a normalization factor resulting from Bayes' Rule. We use a Gaussian approximation of the proposal distribution in Equation (6):

$$\begin{aligned} W_t^{(i)} &= W_{t-1}^{(i)} \int \Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x'\right)\Pr\left(x'\middle|x_{t-1}^{(i)}, y_t\right)dx' \\ &\cong W_{t-1}^{(i)}\boldsymbol{\Sigma}_{j=1}^{n_s}\Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x_{s,j}\right)\Pr\left(x_{s,j}\middle|x_{t-1}^{(i)}, y_t\right), \end{aligned} \tag{7}$$

where n_s is the number of sampled poses. States $x_{s,j}$ are sampled from $x_{s,j} \sim \left\{x_{s,j}\middle|\|x_{s,j} - x_t^{*(i)}\| < \delta\right\}$, where δ is a sampling radius [58]. To obtain the current most likely pose, $x_t^{*(i)}$, given a laser scan and context, we use:

$$x_t^{*(i)} = \underset{x}{\operatorname{argmax}}\Pr\left(x\middle|\mathbb{M}_{t-1}^{(i)}, x_t^{+(i)}, z_t, o_t\right), \tag{8}$$

where $x_t^{+(i)} = x_{t-1}^{(i)} \oplus y_{t-1}$ is the predicted next state based on the odometry and motion model. To avoid particle saturation, we use the adaptive resampling method in [57].

As the 2D laser scans, z_t, detect the horizontal plane of obstacles for mapping, the beams used in z_t are independent of the *Context Identification* module that localizes context, o_t, which are above this horizontal plane. Thus, z_t and o_t are conditionally independent given the map such that:

$$\Pr\left(z_t, o_t\middle|\mathbb{M}_{t-1}^{(i)}, x_t^{(i)}\right) = \Pr\left(z_t\middle|\mathbb{M}_{occ_{t-1}}^{(i)}, x_t\right)\Pr\left(o_t\middle|\mathcal{K}_{t-1}^{(i)}, x_t\right), \tag{9}$$

where $\mathbb{M}_{\text{occ}}$ is an occupancy grid and $\mathcal{K}$ is a set of Extended Kalman Filters (EKFs) that track each detected string and the probability density of its location in the environment. As a result, we arrive at our final form for the weight update:

$$W_t^{(i)} = W_{t-1}^{(i)}\Sigma_{j=1}^{n_s}\Pr\left(z_t\middle|\mathbb{M}_{occ_{t-1}}^{(i)}, x_{s,j}\right)\Pr\left(o_t\middle|\mathcal{K}_{t-1}^{(i)}, x_{s,j}\right)\Pr\left(x_{s,j}\middle|x_{t-1}^{(i)}, y_t\right). \quad (10)$$

The set of tuples $(\kappa_k, s_k) \in \mathcal{K}$, in the map $\mathbb{M}$, contain an EKF, κ_k, and a string, s_k for each context landmark detected in the environment. When contextSLAM receives a context detection, $\left(s_i, p_{xyz,i}\right) \in o$, we find the set of EKFs, $\mathcal{K}_{si}$, with matching strings. We then find the EKF within $\mathcal{K}_{si}$ that minimizes the Mahalanobis distance $d_m\left(\kappa_k, p_{xyz,i}\right)$ between the context location and the EKF:

$$\left(\kappa_{\min}, p_{xyz,\min}\right) = \underset{\left(\kappa_i, p_{xyz,i}\right)\in\mathcal{K}_{si}}{\operatorname{argmin}} d_m\left(\kappa_i, p_{xyz,i}\right), \quad (11)$$

where the estimated covariance associated with each EKF, κ_i, is used when computing d_m. If the minimum distance is below a threshold, then the EKF $\kappa_{\min}$ is updated using $p_{xyz,\min}$. Otherwise a new EKF is created using the measurement and appended to $\mathcal{K}$. Thresholding allows for non-unique context instances within the environment.

When updating an EKF, the predictions for the expected location and covariances of the string are simply the last estimates, since context observations are not expected to move in the map frame from one time step to another:

$$\hat{o}_{i,t|t-1} = \hat{o}_{i,t-1}, \quad (12)$$

$$\hat{\Sigma}_{i,t|t-1} = \hat{\Sigma}_{i,t-1}. \quad (13)$$

The measurement model for each context, i, is computed from the a priori estimate, $\hat{o}_{i,t|t-1}$, and the current measured pose, $x_{xy\theta}$, such that:

$$h\left(\hat{o}_{i,t|t-1}\ ,\ x_{xy\theta}\right) = \begin{bmatrix} \left|\left|\hat{o}_{i,t|t-1\ xy} - x_{xy}\right|\right|^2 \\ atan\left(\hat{o}_{i,t|t-1\ x} - x_x,\ \hat{o}_{i,t|t-1\ y} - x_y\right) - x_\theta \end{bmatrix}. \quad (14)$$

Then, the context's 3D world coordinate, $p_{xyz,i}$, is projected into polar coordinates, $[r_{i,t}, \theta_{i,t}]^T$, where $r_{i,t}$ and $\theta_{i,t}$ are the distance and heading of detected context, i, in the robot's frame. The projected point is then used to compute the residual error $y_t = [r_{i,t}, \theta_{i,t}]^T - h\left(\hat{o}_{i,t|t-1}\ ,\ x_{xy\theta}\right)$. The residual error is then used with the standard EKF correction update:

$$y_{i,t} = \begin{bmatrix} r_{i,t} \\ \theta_{i,t} \end{bmatrix} - h\left(\hat{o}_{i,t|t-1}\ ,\ x_{xy\theta}\right), \quad (15)$$

$$Q_{i,t} = J_{i,t}\hat{\Sigma}_{i,t|t-1}\ J_{i,t}^T + R_{i,t}, \quad (16)$$

$$K_{i,t} = \hat{\Sigma}_{i,t|t-1}\ J_{i,t}^T * Q_{i,t}^{-1}, \quad (17)$$

$$\hat{o}_{i,t|t} = \hat{o}_{i,t|t-1} + K_{i,t}y_{i,t}, \quad (18)$$

$$\hat{\Sigma}_{i,t|t} = (I - K_{i,t}J_{i,t})\hat{\Sigma}_{i,t|t-1} + K_{i,t}R_{i,t}K_{i,t}^T, \quad (19)$$

where R_t is the noise associated with range measurement $[r, \theta]^T$, J_t is the Jacobian of $h\left(\hat{o}_{t|t-1}\ ,\ x_{xy\theta}\right)$ with respect to $\hat{o}_{t|t-1}$, and I is the identity matrix.

ContextSLAM provides the most confident particle and its corresponding context map to the other modules. An overview of the contextSLAM Algorithm is presented in Algorithm 1.

Algorithm 1: contextSLAM: RBPF method extension to include context.

```
Require:
$\Phi_{t-1}^{(i)}$, the sample set of the previous time step; $z_t$, the current laser scan from Obstacle Detection; $o_t$, the current context observation from Context Identification; and $y_{t-1}$, the current odometry observation.
Ensure:
$\Phi^t = \{\}$ #The new sample set
for $\phi_{t-1}^{(i)} \in \Phi_{t-1}$do

    $(x_{t-1}^{(i)}, W_{t-1}^{(i)}, \mathbb{M}_{t-1}^{(i)}) = \phi_{t-1}^{(i)}$
    $\left(\mathbb{M}_{occ_{t-1}}^{(i)}, \mathcal{K}_{t-1}^{(i)}\right) = \mathbb{M}_{t-1}^{(i)}$ #Expand context map into grid and context EKFs.
   $x_t^{+(i)} = x_{t-1}^{(i)} \oplus y_{t-1}$ #Motion model
    $x_t^{*(i)} = \underset{x}{\operatorname{argmax}} \Pr(x \Big| \mathbb{M}_{t-1}^{(i)}, x_t^{+(i)}, z_t, o_t)$ #Max probability state of $x_t^{(i)}$.
    If $x_t^{*(i)} =$ failure then
        $x_t^{(i)} \sim \Pr\left(x_t^{(i)} | x_{t-1}^{(i)}, y_{t-1}\right)$
            $W_t^{(i)} = W_{t-1}^{(i)} \Pr(z_t \Big| \mathbb{M}_{occ_{t-1}}^{(i)}, x_{s,j}) \Pr(o_t \Big| \mathcal{K}_{t-1}^{(i)}, x_{s,j})$ #Next particle weights.
    Else
        for $j = 1, \ldots, n_s$ do #Sample around the node
            $x_{s,j} \sim \left\{ x_{s,j} \Big| \Big| x_{s,j} - x_t^{*(i)} \Big| \Big| < \delta \right\}$
        end for
        $\mu_t^{(i)} = (0,0,0)^T$ #Compute Gaussian proposal
        $\Sigma = 0$
        $n_\mu^{(i)} = 0$
        for all $x_{s,j} \in \{x_{s,1}, \ldots, x_{s,n_s}\}$ do
            $\mu_t^{(i)} \leftarrow \mu_t^{(i)} + x_{s,j} \Pr(z_t \Big| \mathbb{M}_{occ_{t-1}}^{(i)}, x_{s,j}) \Pr(o_t \Big| \mathcal{K}_{t-1}^{(i)}, x_{s,j}) \Pr(x_{s,j} \Big| x_{t-1}^{(i)}, y_t)$
            $n_\mu^{(i)} \leftarrow n_\mu^{(i)} + \Pr(z_t \Big| M_{occ_{t-1}}^{(i)}, x_{s,j}) \Pr(o_t \Big| \mathcal{K}_{t-1}^{(i)}, x_{s,j}) \Pr(x_{s,j} \Big| x_{t-1}^{(i)}, y_t)$
        end for
        $\mu_t^{(i)} \leftarrow \mu_t^{(i)} / n_\mu^{(i)}$
        for all $x_{s,j} \in \{x_{s,1}, \ldots, x_{s,n_s}\}$ do
            $\Sigma_t^{(i)} \leftarrow \Sigma_t^{(i)} + (x_{s,j} - \mu_t^{(i)})(x_{s,j} - \mu_t^{(i)})^T \cdot$
            $\Pr(z_t \Big| M_{occ_{t-1}}^{(i)}, x_{s,j}) \Pr(o_t \Big| \mathcal{K}_{t-1}^{(i)}, x_{s,j}) \Pr(x_{s,j} \Big| x_{t-1}^{(i)}, y_t)$
        end for
        $\Sigma_t^{(i)} \leftarrow \Sigma_t^{(i)} / n_\mu^{(i)}$
        $x_t^i \sim \mathcal{N}\left(\mu_t^{(i)}, \Sigma_t^{(i)}\right)$ #Sample new pose
        $W_t^{(i)} = W_{t-1}^{(i)} n_\mu^{(i)}$ #Update particle weights
    end if
    $\mathbb{M}_{occ_t}^{(i)} = \text{integrateScan}\left(\mathbb{M}_{occ_{t-1}}^{(i)}, x_t^{(i)}, z_t\right)$ #Update occupancy grid
    $\mathcal{K}_t^{(i)} = \text{integrateText}\left(\mathcal{K}_{t-1}^{(i)}, x_t^{(i)}, z_t\right)$ #Update maps with context
    $\Phi_t^{(i)} = \Phi_t^{(i)} \cup \left\{ \left( x_t^{(i)}, W_t^{(i)}, \left( \mathbb{M}_{occ_t}^{(i)}, \mathcal{K}_t^{(i)} \right) \right) \right\}$ #Update sample set
end for
$N_{eff} = 1/\Sigma_{i=1}^{|\Phi_t|} (W_t^{(i)} / \Sigma_{j=1}^{|\Phi_t|} W_t^{(j)})^2$
If $N_{eff} < T$ then
    $\Phi_t = \text{resample}(\Phi_t)$
end if
```

3.5. *Aisle Detection*

The *Aisle Detection* module identifies potential aisles within the context map generated by the *Context Mapping* module. We examine the contours detected in the context map to find a set of candidate aisles, $\mathcal{A}$, that satisfy the geometric constraints related to the minimum/maximum aisle dimensions, and wall parallelism. The set of context EKFs associated with aisle a_k are:

$$\mathcal{A}_{\mathcal{C}_{\|}} = \{s_m | (\kappa_m, s_m) \in \mathcal{K}, d_c(a, \kappa_{\mathrm{m},\mu}) < w_{a,\min}/2\}, \tag{20}$$

where $d_c(a_k, \kappa_{m,\mu})$ is the minimum distance between an aisle and the point $\kappa_{m,\mu}$ which is the mean of κ_m. Aisle product and aisle geometry information is provided to the Action Deliberation module to aid in searching for products.

3.6. *Action Deliberation*

The *Action Deliberation* module controls the robot's actions based on the information provided by the other modules. It uses a finite state machine that contains three main states, Figure 3. It is initiated by user input, leading to the *Begin Exploration* state transition. A copy of the search query, $\mathcal{Q}_{\mathrm{cp}} = \mathcal{Q}$, tracks the set of unfound products. The main states are presented below:

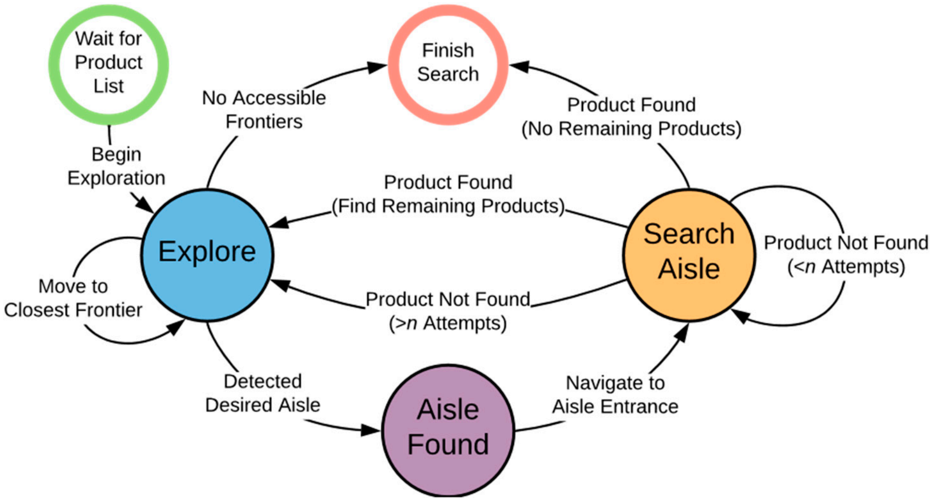

Wait for Product List
Finish Search
No Accessible Frontiers
Product Found (No Remaining Products)
Begin Exploration
Product Found (Find Remaining Products)
Explore
Search Aisle
Product Not Found (<n Attempts)
Move to Closest Frontier
Product Not Found (>n Attempts)
Detected Desired Aisle
Aisle Found
Navigate to Aisle Entrance

Figure 3. The finite state machine used for robot Action Deliberation.

3.6.1. Explore

The robot explores an unknown environment by navigating to frontiers provided by *Frontier Detection*. Frontiers are detected by examining the context map provided by *Context Mapping* module using frontier exploration [58], which provides a set of points representing a boundary between explored and unexplored space in the context map. The point with the minimum estimated travel time is provided to the *Navigation* module. The *Navigation* module uses the ROS Move Base package [59], where costmaps are used to plan navigation paths around dynamic obstacles which are executed by the *Low-Level Controller*. In the *Explore* state, if an aisle paired with context is found, the robot transitions to the *Aisle Found* state.

3.6.2. Aisle Found

In the *Aisle Found* state the robot has found an aisle that may contain a product of interest. An aisle is of interest if a product of interest has a Levenshtein distance less than 2 with any context found within the aisle. It navigates to the entrance of that aisle and transitions into the *Search Aisle* state.

3.6.3. Search Aisle

In the Search Aisle state the robot searches the aisle, navigating from the entrance to the opposite end using the Navigation module to locate the products of interest associated with the current aisle $\mathcal{P}_{\text{gr}} = \text{P}(\mathcal{A}_e)$. $\mathcal{P}_{\text{gr}}$ and the map are updated as new observations become available, and replanning takes place. The Product Detection module detects the products with onboard cameras. We use April tags [60] due to their robustness in cluttered environments for this module, as the product identification problem is beyond the scope of this paper. Several papers have only focused on this specific recognition problem [61–63]. Furthermore, the use of April tags is also a common approach in exploration problems when the focus is not on object/product recognition [36,64,65]. When a product, p_{gr}, is found the set of products is updated $\mathcal{Q}_{\text{cp}} \leftarrow \mathcal{Q}_{\text{cp}} \backslash \text{p}_{\text{gr}}$, $\mathcal{P}_{\text{gr}} \leftarrow \mathcal{P}_{\text{gr}} \backslash \text{p}_{\text{gr}}$. The robot continues searching the aisle until $|\mathcal{P}_{\text{gr}}| = 0$ or it has searched an aisle n times. If there are more products on the list, i.e., $|\mathcal{Q}_{\text{cp}}| > 0$, the robot exits the *Search Aisle* state and enters the *Explore* state. If $|\mathcal{Q}_{\text{cp}}| = 0$, it enters the *Finish Search* state.

3.6.4. Finish Search

In the *Finish Search* state the robot has finished searching for products and returns to its home location. As the primary robot task is product search, the environment does not have to be fully explored when the search is complete.

4. Blueberry Robot Implementation

The grocery robot architecture is integrated into our Blueberry platform (Figure 4). The head has an RGB and depth camera for context identification and localization. The torso contains RGB cameras for product detection. The lidar generates a 3D point cloud which is converted into a 2D laser scan used for mapping and obstacle detection.

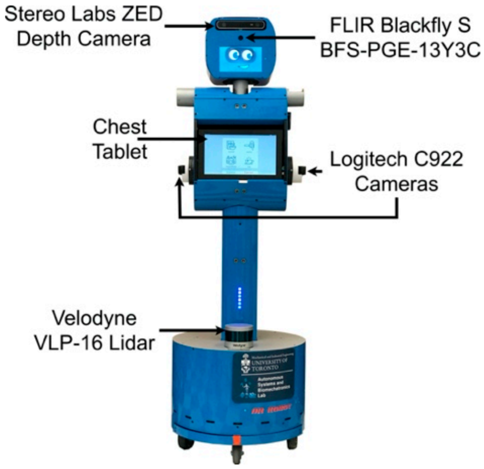


Figure 4. Blueberry Robot with labeled robot components.

5. Experiments

We conducted extensive experiments to determine: (1) the accuracy of contextSLAM in comparison to an existing popular SLAM approach, and (2) the success rate of the architecture searching unknown real environments for products.

5.1. Map Performance

The performance of contextSLAM is validated by the accuracy of the robot's entire trajectory using its generated map. This accuracy is defined by the root mean square error (RMSE):

$$\text{RMSE} = \left(\Sigma_{k=1}^{t}(x_k - \hat{x}_k)^2/t\right)^{0.5}. \tag{21}$$

We measure the accuracy of the estimated robot's entire trajectory $x_{1:t}$ with respect to the ground truth trajectory, $\hat{x}_{1:t}$. Experiments were performed using the Stage Simulator [66] on four 19.6 × 29.5 m^2 environments consisting of: (1) mixed: aisles, dead-ends, and closed paths, Figure 5a; (2) dead-ends: aisles having dead-ends, Figure 5b; (3) loops: many closed paths, Figure 5c; and (4) circles: circular spaces with closed paths and dead-ends, Figure 5d. The ground truth trajectory was obtained from the ground truth state at each timestep as reported by Stage.

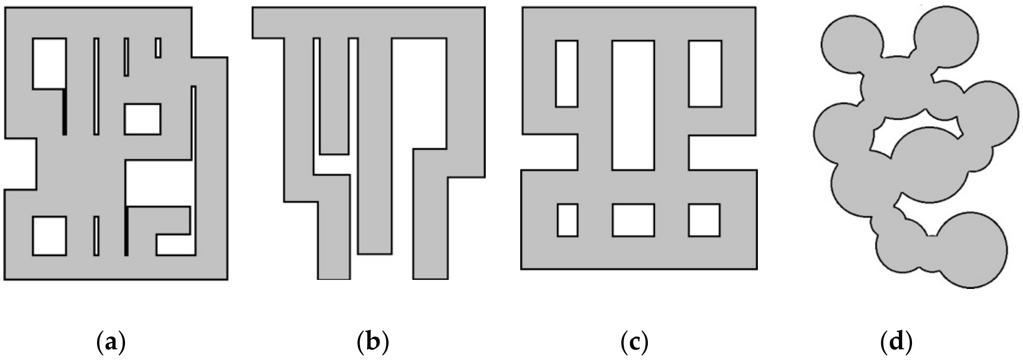

(**a**) (**b**) (**c**) (**d**)

Figure 5. Simulated Environments: (**a**) mixed; (**b**) dead-end; (**c**) loops; (**d**) circles.

To simulate crowded retail conditions, we incorporated: (1) scene text, and (2) dynamic obstacles. Unique text were placed in 30 random locations in each environment and were published directly to the *Context Mapping* module. Context observations were tuples containing a randomly generated text string and the associated 2D world coordinate.

To simulate limited visibility of the cameras, a context observation was considered visible when it was within 3 m and 1.0 rad of the robot's heading. When the robot navigated the environment, and the true location of a context was within its 2D sensory range, the *Context Mapping* module would receive an array of all visible context observations, where the provided 2D world coordinate would have a simulated measurement error modelled by a Gaussian distribution. In each environment, 45 dynamic obstacles were moving in circular motion paths. Each obstacle's path radius was uniformly sampled between (0,1] m with velocity (0,1] m/s.

We manually navigated the robot within each environment four times, with unique paths. Each path was tested 13 times with different measurement errors and each trial was repeated 3 times (39 trials in total). The measurement error was represented by a Gaussian distribution with $\left(0, \sigma_r^2 \in \Xi\right)$ and $\left(0, \sigma_\theta^2 \in \Xi\right)$ for the range and angular measurements, such that Ξ = {0.005,0.015,0.025,0.035,0.045}. In each trial a map is generated using contextSLAM and robot location estimates are obtained. For comparison purposes, we also conduct the same experiments using the popular GMapping SLAM approach. The RMSE of the predicted trajectory using estimates from both methods is presented in the boxplot in Figure 6 and a visualization of the predicted trajectories is presented in Figure 7.

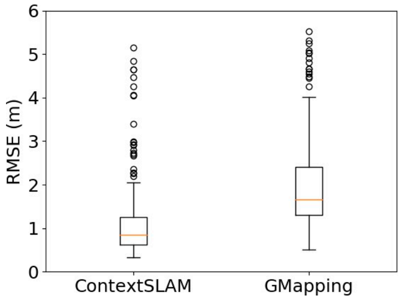


Figure 6. RMSE boxplot of predicted trajectories of contextSLAM and GMapping.

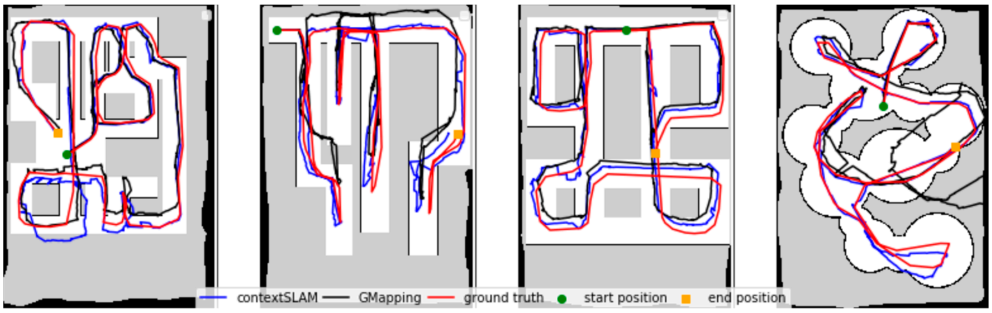


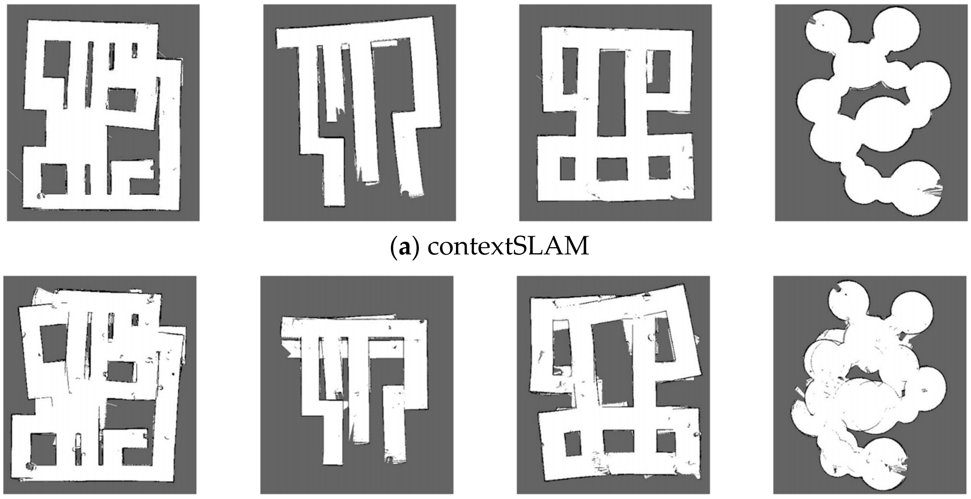
Figure 7. Predicted trajectories of contextSLAM (blue) and GMapping (black) in the four environments with respect to the ground truth (red).

5.1.1. Trajectory Prediction Results

The results show that our contextSLAM method had a lower RMSE of 1.14 m, compared to a RMSE of 1.98 m for GMapping. Furthermore, the Wilcoxon Sign Rank non-parametric test showed that the RMSE is statistically significantly smaller for our contextSLAM approach ($Z = 3150.0, p < 0.001$). In Figure 7, we see that there is a significant improvement on trajectory predictions, particularly in environments that are lacking in corners, such as aisles having dead-ends and circular spaces.

5.1.2. Map Generation Results

Figure 8 shows examples of maps generated by contextSLAM (Figure 8a) and GMapping (Figure 8b), with 0.015 m radial and 0.025 rad angular context detection error. The maps demonstrate that contextSLAM maps had fewer false walls when compared to GMapping. The presence of false walls shows that GMapping failed to recognize and accurately localize in previously seen areas. For example, in the dead-end environment, an extra corridor was mapped. The use of salient text features in the environment allowed contextSLAM to build the environment map more effectively in areas that otherwise lacked corner features such as the circle environment. As a result, the use of contextual features shows a significant improvement towards loop closures by providing additional salient landmarks to localize against.



(**a**) contextSLAM

(**b**) GMapping

Figure 8. Maps generated by (**a**) ContextSLAM and (**b**) GMapping at 0.015 m radial and 0.025 rad angular context detection error.

5.2. Using the Grocery Robot Architecture to Find Products

We evaluated the overall performance of the grocery robot architecture using Blueberry in two environments: (1) a grocery store-like environment, and (2) a real grocery store. The experiments test the feasibility and robustness of the architecture and its application to real-world environments with significant amounts of contextual information and product variety in both a controlled and real store setting. A series of trials were performed where the robot was provided a list of three to four items to find that were distributed within aisles in the environment. The robot started in its home location and explored the unknown environment with a maximum speed of 0.4 m/s to find all the items and returned home again. We consider the worst-case scenarios where the robot is deployed within a previously unseen store (no map information is available) or after layout changes have occurred (requiring the generation of a new map). The experiments were approved by the University of Toronto Ethics Committee (protocol number 37011) and all participants gave their informed consent prior to participating in the experiments.

5.2.1. Store-Like Environment

The 7 × 10 m^2 environment consisted of an open area with the robot's home location and three parallel aisles containing the products on the list (Figure 9a). The open area represented the front of a store and was 2 × 4 m^2. Each aisle was 1.8 × 9 m^2. Hanging over the middle of each aisle was a two-sided aisle sign containing six product categories in that aisle (Figure 9b). The signs were 0.9 × 0.6 m^2 and 2.7 m above the ground.

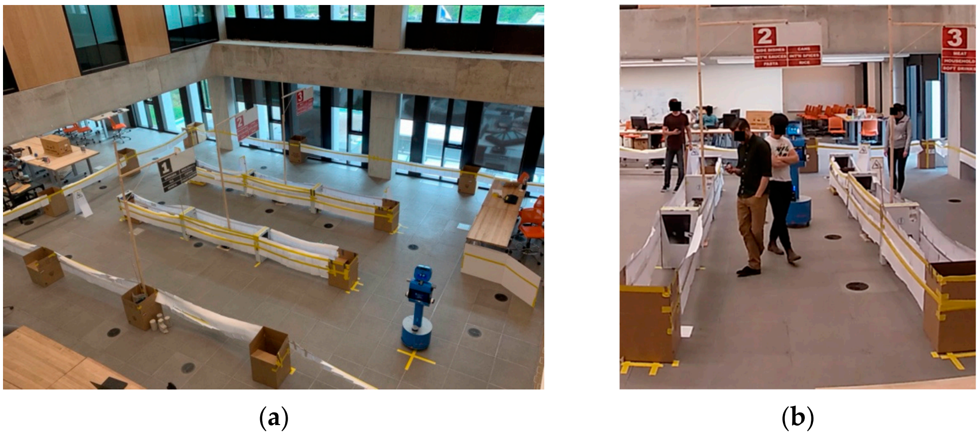


Figure 9. Store-like environment: (**a**) top down view; (**b**) store-like aisle.

Product search queries were generated using all combinations of 3–4 products consisting of: tea, cereal, pasta and household products. Two trials were performed for each query, for a total of 20 trials. Half were conducted with just the robot in the environment and the other half with up to three dynamic people in the environment with their own goals.

Store-Like Environment Results and Discussions

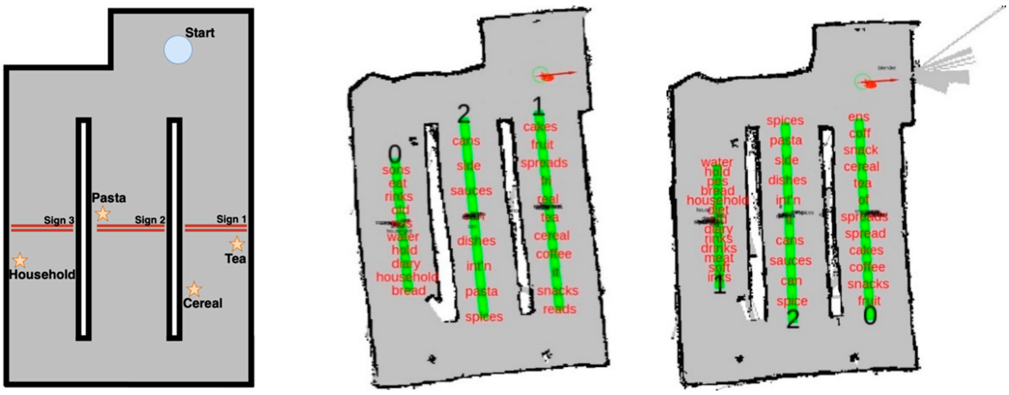
Table 1 presents the number of attempts taken by Blueberry to find each product on the list and the physical search time to navigate the scene to find all the products for each trial. The number of attempts defines the number of times the robot traveled down an aisle to find the specified product location. In trials 1–8, three products were requested for the search, and in trials 9 and 10, all four products were requested. In Table 1, the non-requested products are represented by N/A. Blueberry was able to find all the requested products in every trial. In the experiments without people, the mean time to find a product was 79.5s (σ = 10.17), with a mean of 1.06 searches (σ = 0.24). In the experiments with dynamic people the mean time to find a product was 120.5 s (σ = 60.5), with a mean of 1.34 searches (σ = 0.64). The combined average computation time for running the context identification, mapping and planning methods together was only 60 ms.

Table 1. Architecture performance and total search time in store-like environment.

Product / Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
No People—Number of Attempts to Find a Product										
Tea	1	1	N/A	N/A	1	1	1	1	1	1
Cereal	1	1	1	1	N/A	N/A	1	1	1	1
Pasta	2	1	1	1	2	1	N/A	N/A	1	1
Household	N/A	N/A	1	1	1	1	1	1	1	1
Total Time (s)	230	170	240	285	250	205	230	270	270	235
With Dynamic People—Number of Attempts to Find a Product										
Tea	1	1	N/A	N/A	1	1	1	1	1	1
Cereal	1	1	1	2	N/A	N/A	2	2	1	1
Pasta	1	1	3	3	1	2	N/A	N/A	1	1
Household	N/A	N/A	1	1	3	2	1	1	1	1
Total Time (s)	225	200	835	519	390	346	282	320	360	303

There were six trials with dynamic people that required two or three searches for individual products. The trials that required three aisle searches were 3–5. In trials 3 and 4, there were instances of two people standing close together in the aisle (Figure 9b). The leg detector could not distinguish their legs and detected them as static objects, and, thus, they were included in the map. The robot then selected an alternative path, for example when searching for pasta in trial 3, it went through aisle 1 to navigate around them and reach its goal pose in aisle 2. Once the robot obtained new measurements of the aisle, the map was updated, and Blueberry followed a direct path through the aisle. In trial 5, the robot could not find a safe path around a large number of people blocking the aisle. It replanned, and on its 3rd search it was able to find a safe path through the aisle around the people. In general, since our person detection technique detects dynamic clusters as legs, wearing different clothing with varying colors or patterns did not impact its performance.

Sample context maps generated by contextSLAM are shown in Figure 10 for trial 9. The green lines represent aisle locations, numbered in the order of detection. The number is displayed if the aisle was associated with any context. Clusters of black text in the middle of the aisles are detected text on aisle signs. Context that has been matched with an aisle is shown in red.

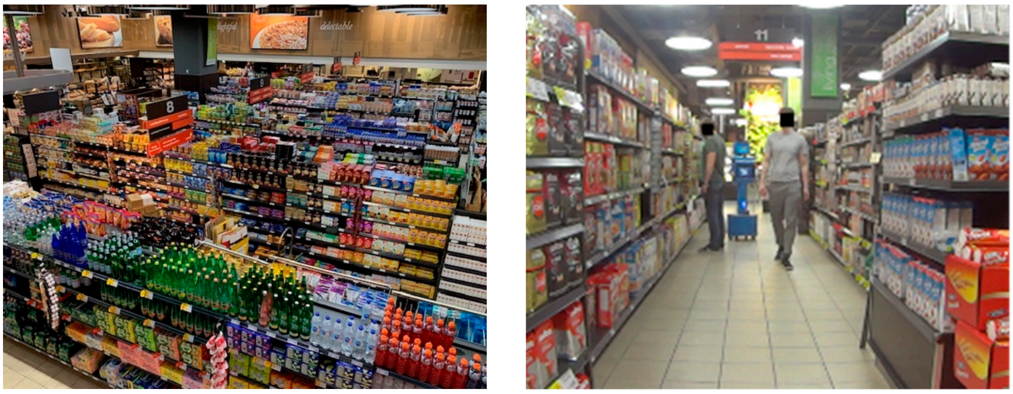


(**a**) (**b**) (**c**)

Figure 10. Grocery-like environment: (**a**) Layout, (**b**,**c**) Context maps made in trial 9 with no people and dynamic people, respectively.

5.2.2. Grocery Store Environment

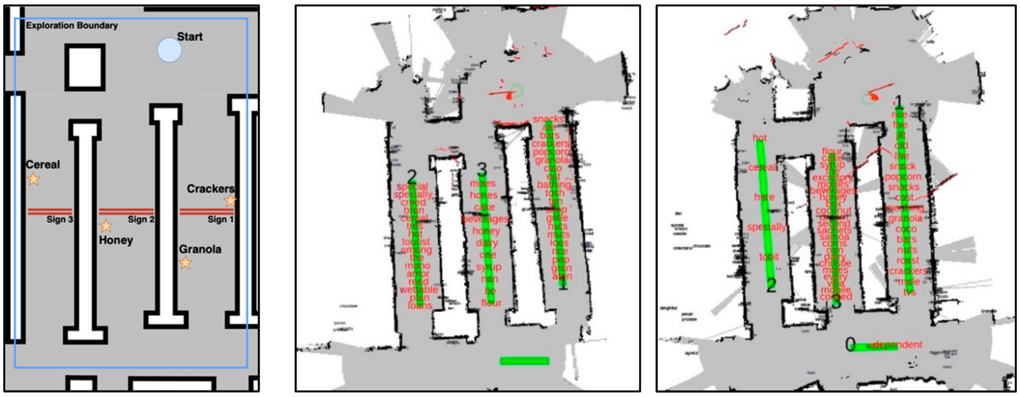
Similar experiments were performed in a real grocery store (Figure 11a), which had significantly more clutter and context due to posters and packaging.



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

Figure 11. Real grocery store: (**a**) top down view; (**b**) crowded store aisle.

The experiment was conducted in an 8 × 14 m^2 section of the store which contained an approximately 5 × 7 m^2 open area and three aisles. The aisles in the search area were approximately 2.5 × 12 m^2. Two-sided signs were over the middle of each aisle (Figure 11b) and used different fonts than in experiment 1. Each sign contained 3–5 product categories. Search queries were generated using combinations of three or four products. A total of 10 trials were conducted, five with just the robot, and five with two dynamic people randomly walking and looking at items in the aisles. The robot always started in the open area in front of the aisles (Figure 12a). A video of Blueberry searching for products using our grocery robot architecture in this environment is presented here on our lab's YouTube channel: https://youtu.be/9RYUxPVIhkM.



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

Figure 12. Real-store environment: (**a**) Layout, (**b**,**c**) Context maps made in trial 9 with no people and dynamic people, respectively.

Grocery Store Environment Results and Discussion

Table 2 shows the number of search attempts needed to find each product and the time to find all products for each trial. The robot found all requested products in each trial. Trials without people, had a mean of 1.06 searches (σ = 0.24). Trials with people had a mean of 1.44 searches (σ = 0.50).

Table 2. In store experiments—attempts to find a product and total search time.

	No People					Dynamic People				
Product / Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Crackers	1	N/A	1	1	1	2	N/A	2	1	1
Cereal	1	1	N/A	1	1	2	1	N/A	2	2
Granola	1	2	1	N/A	1	1	2	2	N/A	1
Honey	N/A	1	1	1	1	N/A	1	1	1	1
Total Time (s)	400	364	405	290	424	396	390	240	395	420

It is interesting to note that in some cases, the robot completed the trials with dynamic people in less time. For example, in trial 3 without people Blueberry searched the full environment to find all the desired products. The frontier exploration approach resulted in the robot taking a longer path by searching all aisles. However, in the dynamic people condition, all the products were found without having to search one of the aisles. In this case Blueberry chose to navigate the adjacent aisle, thus resulting in a shorter trial time.

The context maps generated in trial 5 are shown in Figure 12. The maps highlight the significant increase in text, observed in the grocery store compared to the store-like environment. Even with increased detections, the robot was able to efficiently find the products by associating text with their respective aisles, allowing for the successful completion of each of the queries.

6. Conclusions

In this paper we present a novel grocery robot architecture for searching for products in unknown cluttered grocery environments. The architecture uniquely combines an OCR system with a new context Simultaneous Localization and Mapping framework (contextSLAM). The contextSLAM method builds a map of the environment using context in the store such as aisle signs for which the robot can use to find products of interest. Experiments showed that a robot using the architecture can find multiple products in different environments with unpredictable dynamic people. Future work will include the investigation and integration of a product detection system such as SKU or few-shot object detection methods, and human-robot interaction studies with shoppers in a grocery store.

Author Contributions: Conceptualization, D.D., C.T. and G.N.; methodology, D.D., C.T. and G.N.; software, D.D. and C.T.; validation, D.D. and C.T.; formal analysis, C.T.; investigation, D.D., C.T. and M.P.-H.; resources, D.D., C.T., M.P.-H. and G.N.; data curation, D.D. and C.T.; writing—original draft preparation, D.D. and M.P.-H.; writing—review and editing, D.D., M.P.-H. and G.N.; visualization, D.D., C.T. and G.N.; supervision, G.N.; project administration, D.D., C.T., M.P.-H. and G.N.; funding acquisition, G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by AGE-WELL Inc., the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chairs program (CRC), the Vector Institute Scholarship in Artificial Intelligence, and the NVIDIA GPU grant.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Research Ethics Committee of the University of Toronto (protocol code 37011 on 12 December 2018).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to sincerely thank Longo Brothers Fruit Markets Inc. and the staff at the local Longo's store for allowing us access to the store to conduct our experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sommer, R.; Aitkens, S. Mental Mapping of Two Supermarkets. *J. Consum. Res.* **1982**, *9*, 211–215. [CrossRef]
2. Binder, T. Walgreens' Beverage Category Reset Leverages VR. *Path Purch. IQ* **2018**. Available online: https://www.pathtopurchaseiq.com/walgreens-beverage-category-reset-leverages-vr (accessed on 29 September 2020).
3. Why Does Costco Have a Treasure Hunt Atmosphere? *Costco* **2020**. Available online: https://www.customerservice.costco.com/app/answers/detail/a_id/716/why-does-costco-have-a-treasure-hunt-atmosphere (accessed on 29 September 2020).
4. USDA ERS-New Products. Available online: ers.usda.gov/topics/food-markets-prices/processing-marketing/new-products/ (accessed on 6 July 2020).
5. Creighton, S.A.; Iii, J.H.B.; Froeb, L.; Kovacic, W.E.; Davis, A.H.; Straight, R.A. *Slotting Allowances in the Retail Grocery Industry: Selected Case Studies in Five Product Categories*; Federal Trade Commission: Washington, DC, USA, 2003. Available online: https://www.ftc.gov/sites/default/files/documents/reports/use-slotting-allowances-retail-grocery-industry/slottingallowancerpt031114.pdf (accessed on 19 September 2021).
6. Aylott, R.; Mitchell, V.-W. An Exploratory Study of Grocery Shopping Stressors. *Int. J. Retail Distrib. Manag.* **1998**, *26*, 362–373. [CrossRef]
7. Park, C.W.; Iyer, E.S.; Smith, D.C. The Effects of Situational Factors on In-Store Grocery Shopping Behavior: The Role of Store Environment and Time Available for Shopping. *J. Consum. Res.* **1989**, *15*, 422–433. [CrossRef]
8. Nilsson, E.; Gärling, T.; Marell, A. Effects of Time Pressure, Type of Shopping, and Store Attributes on Consumers' Satisfaction with Grocery Shopping. *Int. Rev. Retail Distrib. Consum. Res.* **2017**, *27*, 334–351. [CrossRef]
9. *Getting Availability Right*. Oliver Wyman. 2012. Available online: https://www.oliverwyman.com/our-expertise/insights/2012/oct/getting-availability-right.html (accessed on 20 September 2020).
10. Fisher, M.; Krishnan, J.; Netessine, S. *Retail Store Execution: An Empirical Study*; Social Science Research Network: Rochester, NY, USA, 2006.
11. Redman, R. Supermarkets slip in customer satisfaction. *Supermark. News* **2019**. Available online: https://www.supermarketnews.com/consumer-trends/supermarkets-slip-customer-satisfaction (accessed on 20 September 2020).
12. Tonioni, A. Computer Vision and Deep Learning for Retail Store Management. Ph.D. Thesis, University Bologna, Bologna, Italy, 2019.
13. Wiles, S.; Kumar, N.; Roy, B.; Rathore, U. Planogram Compliance: Making It Work. *Cognizant* **2013**, 1–7. Available online: https://cupdf.com/document/planogram-compliance-making-it-work.html (accessed on 19 September 2021).
14. Cameron, J.; Berg, M.; Derr, J.; Duncan, G.; Hyde, E.; Menin, B.; Stumpf, L.; Titzel, M.; Wright, F. PCoA State of Older Adults During COVID-19 Report. In *Pennsylvania Council on Aging Report*; 2020. Available online: aging.pa.gov/organization/pa-council-on-aging/Documents/State%20of%20Older%20Adults%20During%20COVID/PCoA%20State%20of%20Older%20Adults%20During%20COVID-19%20Report.pdf (accessed on 19 September 2021).
15. Mohamed, S.C.; Rajaratnam, S.; Hong, S.T.; Nejat, G. Person Finding: An Autonomous Robot Search Method for Finding Multiple Dynamic Users in Human-Centered Environments. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 433–449. [CrossRef]
16. Mišeikis, J.; Caroni, P.; Duchamp, P.; Gasser, A.; Marko, R.; Mišeikienė, N.; Zwilling, F.; de Castelbajac, C.; Eicher, L.; Früh, M.; et al. Lio-A Personal Robot Assistant for Human-Robot Interaction and Care Applications. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5339–5346. [CrossRef]
17. Kornatowski, P.M.; Bhaskaran, A.; Heitz, G.M.; Mintchev, S.; Floreano, D. Last-Centimeter Personal Drone Delivery: Field Deployment and User Interaction. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3813–3820. [CrossRef]
18. Scherer, J.; Rinner, B. Multi-UAV Surveillance With Minimum Information Idleness and Latency Constraints. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4812–4819. [CrossRef]
19. Buchanan, R. Employee Turnover in a Grocery. *Chron* **2010**. Available online: Smallbusiness.chron.com/employee-turnover-grocery-15810.html (accessed on 20 September 2020).
20. Matsuhira, N.; Ozaki, F.; Tokura, S.; Sonoura, T.; Tasaki, T.; Ogawa, H.; Sano, M.; Numata, A.; Hashimoto, N.; Komoriya, K. Development of Robotic Transportation System-Shopping Support System Collaborating with Environmental Cameras and Mobile Robots. In Proceedings of the ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), Munich, Germany, 7–9 June 2010; IEEE: New York, NY, USA, 2010; pp. 893–898.
21. Gross, H.-M.; Boehme, H.; Schroeter, C.; Mueller, S.; Koenig, A.; Einhorn, E.; Martin, C.; Merten, M.; Bley, A. TOOMAS: Interactive Shopping Guide Robots in Everyday Use—Final Implementation and Experiences from Long-Term Field Trials. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; IEEE: New York, NY, USA, 2009; pp. 2005–2012.
22. Cleveland, J.; Thakur, D.; Dames, P.; Phillips, C.; Kientz, T.; Daniilidis, K.; Bergstrom, J.; Kumar, V. Automated System for Semantic Object Labeling With Soft-Object Recognition and Dynamic Programming Segmentation. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 820–833. [CrossRef]
23. Coppola, D. U.S.: Number Supermarket/Grocery Stores 2018. Statista. Available online: https://www.statista.com/statistics/240892/number-of-us-supermarket-stores-by-format/ (accessed on 19 September 2020).
24. Durrant-Whyte, H.; Bailey, T. Simultaneous Localization and Mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]

25. Bailey, T.; Durrant-Whyte, H. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
26. Naudet-Collette, S.; Melbouci, K.; Gay-Bellile, V.; Ait-Aider, O.; Dhome, M. Constrained RGBD-SLAM. *Robotica* **2021**, *39*, 277–290. [CrossRef]
27. Havangi, R. Robust Square-Root Cubature FastSLAM with Genetic Operators. *Robotica* **2021**, *39*, 665–685. [CrossRef]
28. Taheri, H.; Xia, Z.C. SLAM; Definition and Evolution. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104032. [CrossRef]
29. Neto, O.D.A.R.; Lima Filho, A.C.; Nascimento, T.P. A Distributed Approach for the Implementation of Geometric Reconstruction-Based Visual SLAM Systems. *Robotica* **2021**, *39*, 749–771. [CrossRef]
30. Liu, D. A Data Association Algorithm for SLAM Based on Central Difference Joint Compatibility Criterion and Clustering. *Robotica* **2021**, 1–18. [CrossRef]
31. Azzam, R.; Taha, T.; Huang, S.; Zweiri, Y. Feature-Based Visual Simultaneous Localization and Mapping: A Survey. *SN Appl. Sci.* **2020**. [CrossRef]
32. Kowalewski, S.; Maurin, A.L.; Andersen, J.C. Semantic Mapping and Object Detection for Indoor Mobile Robots. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *517*, 012012. [CrossRef]
33. Francis, J.; Drolia, U.; Mankodiya, K.; Martins, R.; Gandhi, R.; Narasimhan, P. MetaBot: Automated and Dynamically Schedulable Robotic Behaviors in Retail Environments. In Proceedings of the 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Washington, DC, USA, 21–23 October 2013; IEEE: New York, NY, USA, 2013; pp. 148–153.
34. Thompson, C.; Khan, H.; Dworakowski, D.; Harrigan, K.; Nejat, G. An Autonomous Shopping Assistance Robot for Grocery Stores. In Proceedings of the Workshop on Robotic Co-workers 4.0: Human Safety and Comfort in Human-Robot Interactive Social Environments 2018 IEEE/RSJ, Madrid, Spain, 1–5 October 2018.
35. Schroeter, C.; Gross, H.-M. A Sensor Independent Approach to RBPF SLAM Map Match SLAM Applied to Visual Mapping. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; IEEE: New York, NY, USA, 2008; pp. 2078–2083.
36. Houben, S.; Droeschel, D.; Behnke, S. Joint 3D Laser and Visual Fiducial Marker Based SLAM for a Micro Aerial Vehicle. In Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden, Germany, 19–21 September 2016; IEEE: New York, NY, USA, 2016; pp. 609–614.
37. Wurm, K.M.; Stachniss, C.; Grisetti, G. Bridging the Gap between Feature and Grid Based SLAM. *Robot. Auton. Syst.* **2010**, *58*, 140–148. [CrossRef]
38. Junior, R.P.; Petry, M.R. Robot Localization Through Optical Character Recognition of Signs. In Proceedings of the 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Porto, Portugal, 24–26 April 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
39. Nieto, J.I.; Guivant, J.E.; Nebot, E.M. The HYbrid Metric Maps (HYMMs): A Novel Map Representation for DenseSLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; IEEE: New York, NY, USA, 2014; pp. 391–396.
40. Lemus, R.; Díaz, S.; Gutiérrez, C.; Rodríguez, D.; Escobar, F. SLAM-R Algorithm of Simultaneous Localization and Mapping Using RFID for Obstacle Location and Recognition. *J. Appl. Res. Technol.* **2014**, *12*, 551–559. [CrossRef]
41. Sim, R.; Little, J.J. Autonomous Vision-Based Exploration and Mapping Using Hybrid Maps and Rao-Blackwellised Particle Filters. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; IEEE: New York, NY, USA, 2006; pp. 2082–2089.
42. Schulz, R.; Talbot, B.; Lam, O.; Dayoub, F.; Corke, P.; Upcroft, B.; Wyeth, G. Robot Navigation Using Human Cues: A Robot Navigation System for Symbolic Goal-Directed Exploration. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: New York, NY, USA, 2015; pp. 1100–1105.
43. Case, C.; Suresh, B.; Coates, A.; Ng, A.Y. Autonomous Sign Reading for Semantic Mapping. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: New York, NY, USA, 2011; pp. 3297–3303.
44. Wang, H.; Finn, C.; Paull, L.; Kaess, M.; Rosenholtz, R.; Teller, S.; Leonard, J. Bridging Text Spotting and SLAM with Junction Features. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015; IEEE: New York, NY, USA, 2015; pp. 3701–3708.
45. Li, B.; Zou, D.; Sartori, D.; Pei, L.; Yu, W. TextSLAM: Visual SLAM with Planar Text Features. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: New York, NY, USA, 2020; pp. 2102–2108.
46. Han, S.; Xi, Z. Dynamic Scene Semantics SLAM Based on Semantic Segmentation. *IEEE Access* **2020**, *8*, 43563–43570. [CrossRef]
47. Bavle, H.; De La Puente, P.; How, J.P.; Campoy, P. VPS-SLAM: Visual Planar Semantic SLAM for Aerial Robotic Systems. *IEEE Access* **2020**, *8*, 60704–60718. [CrossRef]
48. Chen, X.; Milioto, A.; Palazzolo, E.; Giguere, P.; Behley, J.; Stachniss, C. SuMa++: Efficient LiDAR-based Semantic SLAM. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: New York, NY, USA, 2019; pp. 4530–4537.
49. Zhao, X.; Zuo, T.; Hu, X. OFM-SLAM: A Visual Semantic SLAM for Dynamic Indoor Environments. *Math. Probl. Eng.* **2021**, *2021*, 1–16. [CrossRef]

50. Biederman, I. Recognition-by-Components: A Theory of Human Image Understanding. *Psychol. Rev.* **1987**, *94*, 115–147. [CrossRef]
51. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
52. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017; pp. 2980–2988.
53. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 1492–1500.
54. Shi, B.; Bai, X.; Yao, C. An End-To-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 2298–2304. [CrossRef] [PubMed]
55. Hahnel, D.; Triebel, R.; Burgard, W.; Thrun, S. Map Building with Mobile Robots in Dynamic Environments. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; IEEE: New York, NY, USA, 2013; pp. 1557–1563.
56. Leigh, A.; Pineau, J.; Olmedo, N.; Zhang, H. Person Tracking and Following with 2D Laser Scanners. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; IEEE: New York, NY, USA, 2015; pp. 726–733.
57. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Tran. Robot.* **2007**, *23*, 34–46. [CrossRef]
58. Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, USA, 10–11 July 1997; IEEE: New York, NY, USA, 1997; pp. 146–151.
59. Move_Base. ROS.org. Available online: Wiki.ros.org/move_base (accessed on 29 September 2020).
60. Wang, J.; Olson, E. AprilTag 2: Efficient and Robust Fiducial Detection. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; IEEE: New York, NY, USA, 2016; pp. 4193–4198.
61. Tonioni, A.; Di Stefano, L. Domain Invariant Hierarchical Embedding For Grocery Products Recognition. *Comput. Vis. Image Underst.* **2019**, *182*, 81–92. [CrossRef]
62. Winlock, T.; Christiansen, E.; Belongie, S. Toward Real-Time Grocery Detection for the Visually Impaired. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Workshops, San Francisco, CA, USA, 13–18 June 2010; IEEE: New York, NY, USA, 2010; pp. 49–56.
63. Leo, M.; Carcagni, P.; Distante, C. A Systematic Investigation on End-to-End Deep Recognition of Grocery Products in the Wild. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; IEEE: New York, NY, USA, 2021; pp. 7234–7241.
64. Square Spiral Search (SSS) Algorithm for Cooperative Robots: Mars Exploration. *Int. J. Res. Stud. Comput. Sci. Eng.* **2020**, *7*. [CrossRef]
65. Mu, B.; Giamou, M.; Paull, L.; Agha-mohammadi, A.; Leonard, J.; How, J. Information-Based Active SLAM via Topological Feature Graphs. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; IEEE: New York, NY, USA, 2016; pp. 5583–5590.
66. Vaughan, R. Massively Multi-Robot Simulation in Stage. *Swarm Intell.* **2008**, *2*, 189–208. [CrossRef]