3S-cart: A Lightweight, Interactive Sensor-based Cart for Smart Shopping in Supermarkets

You-Chiun Wang and Chang-Chen Yang

Abstract—Nowadays, shopping has played a key role in our economic activity. It deserves investigation how to provide smart shopping by promptly interacting with customers in supermarkets. The paper proposes a *sensor-based smart shopping cart (3S-cart)* system by using the context-aware ability of sensors to detect the behavior of customers, and respond to them in real time. A prototype of 3S-cart is implemented by encapsulating modularized sensors in a box to be put on shopping carts. Thus, 3S-cart is lightweight and easy to deploy. We also demonstrate two supermarket applications by 3S-cart. In the *sales-promotion application*, each cart checks if its customer has interest in some products and shows sales information at once to increase the purchasing desire. In the *product-navigation application*, a customer asks the system to find an unhindered, shortest path to comfortably obtain the desired product. The paper contributes in exploiting the sensor technology to provide interactive shopping in supermarkets, and addressing the prototyping experience and potential applications of the proposed 3S-cart system.

Index Terms—behavior detection, path calculation, sensor application, shopping cart, supermarket.

1 Introduction

Wireless sensor network (WSN) has drawn a great deal of attention from academic, industrial, and commercial communities. It consists of many embedded devices called *sensors* deployed in regions of interest to collect information from the surroundings. With their context-aware sensing ability, sensors provide an easy and low-priced way to monitor the environment, and support diverse applications from animal control to security surveillance, traffic management, and oceanic exploration [1]. The great progress of WSN has also made it become a core technique in Internet of Things (IoT) [2].

Applying the sensing capability of WSN to supermarkets is an attractive direction. Traditional retailers often use member cards to record the products sold to customers, and analyze their preference accordingly. However, this scheme is offline, since the retailer has no idea what products the customers may have interest (but do not buy) as they are in the supermarket. On the other hand, visual surveillance systems could monitor what customers are doing, but it requires huge computation to analyze customers' behavior and cannot interact with them [3]. Including the intelligence of WSN helps solve the problems by providing online monitoring of the behavior of customers and responding to them in real time. For example, when a customer stops to take a look at some products, WSN can ask the retailer's server to send a notification to the customer about the sales promotion for these products.

With the above motivation, we develop a 3S-cart (*Sensor-based Smart Shopping cart*) system, whose idea is to use a shopping cart to infer its customer's behavior and provide prompt interaction. Specifically, we use an embedded computing platform, called *Arduino* [4], to integrate with multiple sensors on a cart to detect its movement and action from the customer. The cart also has an LCD (liquid crystal display) touch panel to show real-time information (e.g., product or sales data) and allow a customer to send queries (e.g., asking the location of some product and how to reach it). Moreover, we deploy wireless routers on each product shelf, which organize a wireless mesh network to connect all carts with the control server. These routers also help estimate the position of each cart in the supermarket.

Our 3S-cart system has three design goals. First, it has to offer a *lightweight* solution to construct the system. To do so, we adopt off-the-shelf, modularized sensors and encapsulate them in a small box. Such boxes can be easily installed on shopping carts to start up 3S-cart. Moreover, since sensors are modularized, it is convenient to add or update sensors. So, changing the sensing ability of a cart requires a small upgrading cost. Thus, 3S-cart is featured by modularity and scalability. Second, it should be *simple* to use the system. We employ two 'natural' actions of customers in terms of carts to observe their behavior: 1) holding the cart's handle and 2) moving/stopping the cart. In this way, customers can get into the swing of the 3S-cart system shortly. Besides, customers need not bring extra devices like mobile phones to use the system. Third, it can support *cooperation* with other systems. 3S-cart should be able to work alone, but can work together with existing systems such as membership, visual surveillance, and bar-code/QR-code/RFID systems operated in a supermarket. To achieve the goal, 3S-cart provides fast and basic detection of customer behavior. Such detection can be used by other systems to provide more sophisticated analysis. This also helps reduce the computing complexity.

Based on 3S-cart, this paper proposes two applications to demonstrate its practicability. For the *sales-promotion application*, 3S-cart jointly detects the movement of a shopping cart and checks if the customer is holding it, to estimate the customer's preference on products. If the cart finds that the customer has interest in some product, the control server shows product data or sales promotion on its LCD to draw his attention. When a customer walks back and forth or the cart does not move for a long while, the system asks an employee to go to help that customer or retrieve the cart, respectively.

The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, 80424, Taiwan.
E-mail: ycwang@cse.nsysu.edu.tw; zavi.yang@gmail.com

On the other hand, the *product-navigation application* considers how to schedule a comfortable path for a customer to detour obstructions (e.g., product shelves) and beat the crowds. We divide the supermarket into grids to model such environment, and develop an enhanced A* algorithm to find unhindered, shortest paths for customers to walk to get the desired products. The design of both applications is lightweight, and they can meet customer demands in a supermarket.

The contributions of this paper are fourfold: 1) We embed the sensing ability in shopping carts to support real-time interaction with customers to improve their shopping experience. 2) It provides fast, lightweight deployment of the proposed system by using off-the-shelf, modularized sensors. 3) Not only two potential applications are proposed, but also a prototype is demonstrated in a real supermarket. 4) 3S-cart can cooperate with other systems to provide low-cost but sophisticated monitoring of customer behavior.

We organize this paper as follows: Section 2 reviews related work. Section 3 proposes our 3S-cart system. Two supermarket applications are given in Section 4. Section 5 discusses the prototyping experience, while Section 6 presents experimental results. Finally, we draw a conclusion in Section 7.

2 Related Work

Supermarkets usually have visual surveillance systems like cameras or closed-circuit television (CCTV) to assure safety, which can be used for the analysis purpose. For example, Zhang et al. [5] develop an object retrieval application to identify specified products by analyzing the images captured from cameras. Popa et al. [6] use CCTV to detect motion patterns such as walking, looking at products, and touching products, to categorize customers by their behavior. In the work [7], a surveillance system is used to assess shopping behavior (e.g., browsing products or putting them in baskets) to measure customers' level of interest. However, such systems need to extract meaningful information from massive video data, which requires significant computation.

Some researchers use *radio frequency identification (RFID)* to observe the preference of customers on products. Both studies [8], [9] associate products with passive RFID tags to detect their movement caused by customers. Thus, the retailer can find out popular products or what products customers pay attention to. However, these systems lack interaction with customers. In the work [10], each customer uses a smart phone with an RFID reader to look for the location of desired product. However, it only guides the customer to the destination product shelf without addressing the dynamic environment (i.e., obstructions and crowds) in a supermarket.

A few studies equip shopping carts with sensors to expand their ability. In the study [11], a shopping cart actively tracks its lost customer via sensors and a localization algorithm. The cart also has two motors to help it move close to the customer. The work [12] attaches a webcam on a cart to guide a customer to the prescribed locations based on the shopping list. To avoid complex image processing, the supermarket is mapped with some colors, and each cart identifies the color to estimate its position. Nevertheless, none of them adopt shopping carts to offer real-time interaction with customers, which motivates us to propose the 3S-cart system.

The work [13] puts sensors on shopping carts and product shelves in a shopping area, and uses 3D ray launching to analyze sensor deployment to reduce interference and energy consumption. It also develops a monitoring application to help each customer know the amount of products in his cart and the information of these products (e.g., expiration date). Apparently, our 3S-cart system has a different goal with that in [13], as we aim at inferring customers' interest on products by analyzing their behavior through sensors on shopping carts. Moreover, we develop a navigation approach for customers to find unhindered, shortest paths to comfortably get their desired products. This issue is also not addressed in [13].

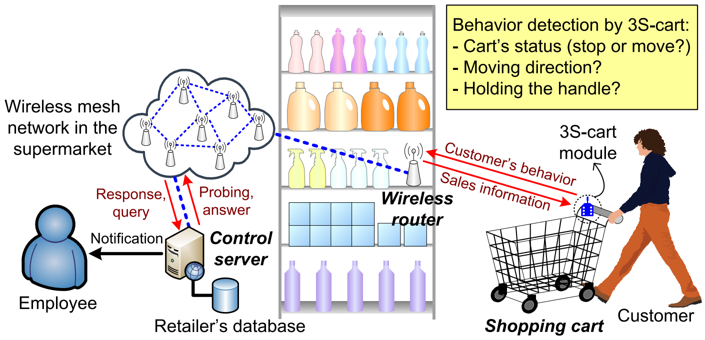


Fig. 1: System architecture of 3S-cart. A shopping cart communicates with the control server, through a wireless router deployed on the product shelf, to report the customer's behavior and receive sales information.

3 3S-Cart for Smart Shopping

In this section, we first present the design and operations of our 3S-cart system, and then discuss the cooperation of 3S-cart with other systems in a supermarket.

3.1 System Design

Fig. 1 shows the 3S-cart architecture, which consists of *wireless routers*, *shopping carts* (with 3S-cart modules), and a *control server*. Since it is common to put the same kind of commodities on a product shelf in supermarkets, we deploy a wireless router on each shelf to associate it with the type of products on that shelf. Once a router detects the coming of a customer (via his cart), we could know what products the customer is looking for. These routers also form a wireless mesh network to connect all devices in the system. Thus, a cart can always keep in touch with the server. Besides, wireless routers are responsible for tracking every cart, which can be achieved by some indoor positioning approaches [14].

A shopping cart acts as the 'interface' to interact with each customer. It carries multiple sensors to monitor the behavior of its customer. In our design, the cart can detect its status (stop or move), the moving direction, and whether the customer is holding its handle. We can infer the customer's behavior by using the combination of sensing data, and the control server gives feedback to the customer via the cart's LCD. For example, when the cart finds that the customer is stopping to check some products, the server can send sales promotion or product data. Moreover, a customer can query the cart about product information (e.g., price or location). The cart then asks the server to send the answer to the customer in real time.

The control server is the decision center and has four missions. First, it monitors the health of each device by sending probing messages. If the server cannot hear any response from a device, it notifies an employee to examine that device. Second, the server is responsible for answering the queries from customers. To do so, it connects to the retailer's database to access product and sales information, as shown in Fig. 1. Third, based on the customer's behavior reported by a shopping cart, the server does actions to respond to the behavior, for instance, sending sales promotion or notifying an employee to assist that customer. Finally, the server keeps track of the geographic distribution of carts and their customers. It finds out 'hotspots' in the supermarket, to which the retailer can make reference for product arrangement.

We remark that the 3S-cart system actually organizes a *hybrid WSN* [15] in the supermarket. Specifically, each shopping cart acts as a mobile sensor, which 'roam' (by its customer) in the WSN and generate sensing data. On the other hand, wireless routers serve as static sensors to support connectivity and message exchange for the whole network. Finally, the control server can be viewed as a sink node to collect data from carts and send commands to the WSN.

3.2 System Operations

In 3S-cart, we define four types of system operations below.

Behavior analysis: This operation involves the collection of sensing data from a shopping cart and the decision making by the control server. We take a product-searching example in Fig. 1. Suppose that a customer wants to buy a detergent. He is holding the cart and walking towards the product shelf with detergents. Then, the cart finds that the customer stops for a while, and thus sends the cart's status to the server via the wireless router on the shelf. The server infers that the customer may have interest in searching detergents, and queries the retailer's database for the sales information of detergents. It then sends such information to the cart, which will be shown on the cart's LCD. This helps the customer in choosing products and thereby arouses his purchasing desire. We will further discuss this operation in Section 4.1.

Query and answer: The customer queries the information of some product (e.g., its location) through the cart's LCD. Then, this query will be relayed by the nearest wireless router and transmitted to the control server. The server then calculates the answer for the query (e.g., finding the shortest path to reach the product's location) and sends it back to the cart. In Section 4.2, we will discuss how to calculate such a path.

Asking for help: Each customer can ask for assistance in two ways. First, the customer *actively* uses the LCD panel on the shopping cart to seek an employee. Since the control server knows where the cart is, it notifies the closest employee to go to the cart's location. Alternatively, when the customer is walking and pushing the cart back and forth, the server will *passively* find that the customer is visiting the same product shelves. Thus, it can ask an employee to help that customer.

Equipment examination: This is a background operation for system maintenance. The control server periodically sends a probing message to each device in the system to check its health. If a device does not respond after several tries, the server marks the device as *suspicious* and can ask an employee to check it. On the other hand, when a shopping cart detects that nobody uses it for a threshold time, it can notify the server to ask an employee to retrieve the cart.

3.3 Cooperation with Other Systems

3S-cart is a stand-alone system to provide fast detection and response to customer behavior. Through its detection result, 3S-cart can collaborate with other systems to support more sophisticated monitoring or personalized service for customers. Below, we discuss how 3S-cart cooperates with existing systems in a supermarket.

Membership systems: A customer usually uses the shopping cart anonymously. However, since most supermarkets have a membership system, we can allow a customer to input the member number in a cart (via the LCD touch panel or a card scanner). Then, the 3S-cart system creates a *one-to-one relationship* between the customer and that cart. In this way, the control server can obtain the customer's profile from the retailer's database, and provide him with member-only or exclusive offers such as special discounts or reward points.

Visual surveillance systems: Cameras or CCTV support more accurate monitoring of customer behavior, but they have to spend huge computation to extract meaningful information from extensive video data. By cooperating with 3S-cart, we can develop a *lightweight, event-driven surveillance system*. Specifically, when a shopping cart detects that its customer may have interest in some products (discussed in Section 4.1), the control server will ask nearby cameras to give a close-up view to that customer[1]. Therefore, the system can provide a finer analysis of the customer's preference in time, without dealing with much irrelevant video data.

Bar-code, QR-code, and RFID systems: Normally, products have bar-codes, QR-codes, or RFID tags used to identify them. To team up with such systems, we can extend 3S-cart's capability by adding a code scanner or an RFID reader to each shopping cart. Thus, once a customer wants to know the detailed information of certain product, he can scan the code or tag by the cart. In this case, the cart can ask the control server to search the corresponding information from the retailer's database and then answer to the customer.

4 Supermarket Applications

Based on the 3S-cart system, we propose two applications in supermarkets. One is to detect the preference of a customer and send him sales information at once to attract his attention. The other is to guide a customer to comfortably walk to the destination product shelf to get the desired product.

4.1 Sales-promotion Application

When a customer has interest in some kind of products, if the retailer can *immediately* tells the customer about the product's introduction or even sales promotion, we can help the customer choose products and improve his shopping experience. However, most supermarkets use billboards or employees to announce such information, which consumes much manpower. With the help of 3S-cart, we can *automatically* detect the potential preference of each customer based on his behavior on a shopping cart, and give the customer his interesting information in real time.

To let customers easily use the system, we adopt two natural actions of a customer in terms of the shopping cart to observe his behavior: 1) holding the cart's handle and 2) moving or stopping the cart. Based on the combination of these two actions, we infer nine possible cases of customer behavior, which is summarized in Table 1. Initially, each cart is in the *working state*, where all of its components are turned on.

Case 1: The customer is holding the cart by both hands, and it is static for a short period T_1 (e.g., 30 seconds). The

1. It can refer to the work [16] for the issue of how to coordinate a subset of cameras to focus on a given object or person.

TABLE 1: Nine cases to analyze a customer's behavior, where $T_1 < T_2 \ll T_3$.

Case	Holding the cart	The cart's status	Analysis result	Standby state
1	By both hands	Static for T_1 time	The customer has moderate interest in some product.	
2	By one hand	Static for T_1 time	The customer has significant interest in some product.	
3	Not holding	Static for T_1 time	The customer temporarily leaves the cart.	✓
4	Holding	Static for T_2 time	The customer may do other things (e.g., chatting).	✓
5	Not holding	Static for T_3 time	The cart is unused and should be retrieved.	✓
6	By one hand	Moving	The customer is browsing some products.	
7	By both hands	Visiting the same shelves	The customer needs assistance.	
8	By both hands	Not visiting the same shelves	The customer has no interest in nearby products.	
9	Not holding	Moving	The cart is not used normally.	✓

customer may be watching some product, which means that he could have moderate interest. Thus, we can show the product's introduction on the cart's LCD to draw his attention.

Case 2: The customer is holding the cart by only one hand, and it is static for T_1 time. There is a high possibility that the customer is taking and checking some product, which indicates that he has significant interest. Thus, we can send him sales promotion or special activity to arouse his purchasing desire.

Case 3: The cart is not held and static for T_1 time. Since the customer may temporarily leave to do other things (e.g., searching nearby products or checking prices), the association between customer and cart is broken. We thus view this case as neutral, and let the cart enter the *standby state*, where it turns off LCD to conserve energy. Once the customer comes back (which can be detected when the customer holds the cart again), the cart returns to the working state. Consequently, the cart turns on its LCD and shows the catalogue or advertisement of products for the customer's reference.

Case 4: The customer is holding the cart (by one or both hands), and it keeps static for a threshold time T_2 (e.g., 3 minutes). We infer that the customer does not use the system, as he is doing something such as chatting with a person. Thus, the cart goes to the standby state by turning off its LCD for power saving. When the customer starts moving the cart, it switches to the working state and turns on LCD accordingly.

Case 5: The cart is not held and static for a long period T_3 (e.g., 30 minutes). We treat the cart as *unused*, so the control server notifies an employee to retrieve that cart. The employee should turn off the 3S-cart module on the cart or change it to the *staff mode*, to avoid confusion on the server side.

Case 6: The customer is moving the cart by only one hand. He may be browsing some products along the aisle, which means that the customer has moderate interest. Thus, we can notify the customer of product data to attract his attention.

Case 7: The customer is moving the cart by both hands, and he has visited the same product shelves multiple times. This case implies that the customer is searching for some products and may require assistance. Consequently, the control server will ask a nearby employee to go to help that customer.

Case 8: The customer is moving the cart by both hands, but he does not visit the same product shelves. Apparently, it means that the customer is just walking through the aisle between these shelves and has no interest in nearby products. Thus, the cart can show the catalogue or advertisement to help the customer look for his interesting products.

Case 9: The cart is not held but moved. This case may occur, for example, when a child is playing the cart or the cart is hit by another one. Since the cart is not used normally, it enters the standby state. However, when somebody is holding the cart, the cart returns to the working state by turning on its LCD to provide service.

When the shopping cart detects that a customer has interest in some products (i.e., cases 1, 2, and 6), the control server will update the corresponding statistics for these products in the database. Thus, the retailer can refer to the statistics to adjust the sales policy or product arrangement in the future.

We remark that there are two advantages of using the above nine cases to estimate the behavior of a customer. First, it only observes the common and basic actions of the customer conducting on the shopping cart (i.e., holding the cart's handle and moving it). Thus, it adds no overhead for customers to use the 3S-cart system. Second, the solution is lightweight and easy to implement. Consequently, the retailer can deploy the system with a low cost to fast and generally collect customers' preference. However, such estimation may not be always accurate. To solve this problem, we can combine 3S-cart with a visual surveillance system. As discussed in Section 3.3, when the cart estimates that its customer may have interest in some product, the control server can ask nearby cameras to focus on that cart to provide more detailed analysis. Alternatively, we can combine 3S-cart with an RFID system. In particular, each product is associated with an RFID tag to detect its movement (the detail can be found in both studies [8], [9]). Since a product item may not be necessarily moved by a customer (e.g., an employee rearranges product items), the control server can identify whether the customer really has interest in some product when both case 2 occurs and one item of that product is moved. This helps significantly increase the monitoring accuracy.

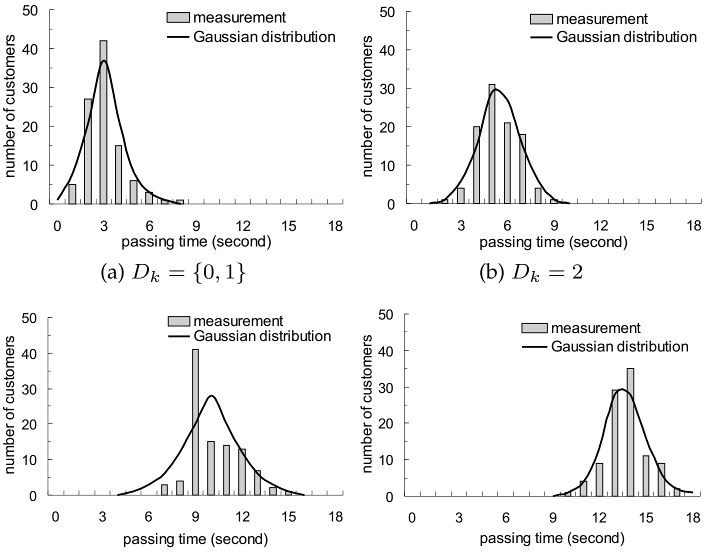
4.2 Product-navigation Application

Most supermarkets are operated by self-service. Customers spontaneously search for the products that they want to buy, and then proceed to the checkout counters. However, without any assistance, finding the products in a large supermarket will be time-consuming [17]. Even though a customer is familiar with the supermarket's layout, he has no idea of the distribution of shopping carts and persons in the supermarket, and thus may be stuck in a crowd. Consequently, it could greatly improve the customer's shopping experience if the system can recommend him a 'comfortable' walking path to the destination which bypasses obstructions and crowds.

To do so, our 3S-cart system divides the supermarket into grids to model the obstructions such as product shelves. Given the initial grid (i.e., the customer's location) and the terminal grid (i.e., the location of desired product), we employ the A* algorithm [18] to find the shortest path between them, which computes a *cost* for each grid k by

$$S(k) = C(k) + H(k), \tag{1}$$

where $C(k)$ is the actual cost from the initial grid to grid k, and $H(k)$ is the predicted cost from grid k to the terminal grid.



(a) $D_k = \{0, 1\}$ (b) $D_k = 2$ (c) $D_k = 3$ (d) $D_k = 4$

Fig. 2: Our measurement of walking time for 100 customers to pass through one grid in the Carrefour supermarket.

From the initial grid, A* iteratively selects the adjacent grid with the smallest cost to move, and updates $C(k)$ and $H(k)$ of some grids accordingly, until reaches the terminal grid.

The A* algorithm can avoid obstructions when selecting a path. However, it assumes only *static* obstructions and thus cannot be directly applied to the supermarket environment, where people may dynamically crowd some regions. In this case, A* would find a path through some *overcrowded* regions. To conquer this problem, we modify A* by considering two factors in each grid: 1) the density of shopping carts and 2) the average walking speed of customers. Specifically, we set the grid length to allow two carts to parallelly pass. Let D_k be the density of carts in grid k, where $D_k \leq 4$. Let us denote by $V_{\text{avg}}(D_k)$ the average walking speed of customers in a grid with density D_k. To consider the supermarket environment, we replace Eq. (1) by the new cost function:

$$S(k) = C(k) \times \max\{1, D_k\} + \frac{H(k)}{V_{\text{avg}}(D_k)}. \quad (2)$$

The original A* algorithm assumes that the cost to pass through each single grid is fixed. In fact, it incurs a higher cost if the customer passes through a grid with more carts. That is why we multiply $C(k)$ by the density factor D_k. However, it is possible that grid k is empty (i.e., $D_k = 0$), so we use the term $\max\{1, D_k\}$ instead of D_k in Eq. (2). On the other hand, A* estimates the value of $H(k)$ by considering a constant moving speed. Actually, $H(k)$ should be reduced if the customer walks faster, and vice versa. Thus, we divide $H(k)$ by the speed factor $V_{\text{avg}}(D_k)$ in Eq. (2).

In practice, it is not easy to estimate the walking speed of customers. Instead, we compute the average time for a customer to pass through one grid with density D_k by

$$T_{\text{avg}}(D_k) = \frac{\text{Unit grid length}}{V_{\text{avg}}(D_k)} \triangleq \frac{1}{V_{avg}(D_k)}. \quad (3)$$

By integrating Eq. (3) to Eq. (2), we can derive the final cost function as follows:

$$S(k) = C(k) \times \max\{1, D_k\} + H(k) \times T_{\text{avg}}(D_k). \quad (4)$$

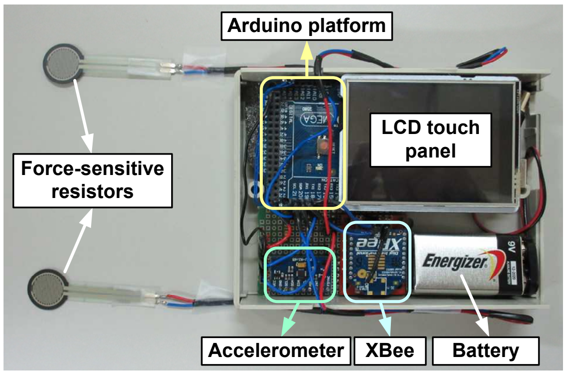


Fig. 3: The prototype of our 3S-cart module on a shopping cart. Attached to the Arduino platform are force-sensitive resistors, an accelerometer, and an XBee device. An LCD touch panel provides interaction with the customer.

The remaining issue is how to find $T_{\text{avg}}(D_k)$ in Eq. (4). To do so, we can analyze the video data captured from the cameras or CCTV installed in a supermarket. For example, Fig. 2 shows our measurement from the Carrefour supermarket (the Love-River Branch in Taiwan). Interestingly, the walking time of customers can be approximated by a Gaussian distribution in most situations (except for $D_k = 3$). The results of $D_k = 0$ and $D_k = 1$ are the same, because a customer can freely pass the gird containing just one shopping cart. In this case, the average walking time $T_{\text{avg}}(D_k)$ is 3.58 seconds. When D_k is 2 and 3, the average walking time will be 5.92 and 10.66 seconds, respectively. When there are four carts in a grid, we can have $T_{\text{avg}}(D_k) = 15.22$ seconds.

Since the distribution of customers will dynamically change as time goes by, we set a tolerable threshold δ (e.g., 50%) to determine whether to 'recompute' the walking path of a customer to react to such change. Let $\mathcal{G}_p$ be the set of grids that constitute the remaining path, and $\mathcal{G}_c$ be the subset of grids in $\mathcal{G}_p$ whose cart density D_k increases. If $|\mathcal{G}_c|/|\mathcal{G}_p| > \delta$, it means that more than half of the current path may become congested. Therefore, our 3S-cart system will adaptively find a new path for the customer to detour these new crowds.

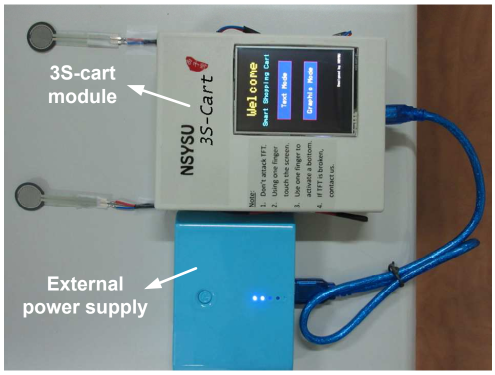
We remark that the A* algorithm is widely used in the field of path-finding and graph traversal. It is an extension of the famous Dijkstra's shortest-path algorithm but can achieve better performance by adopting heuristics to guide the search [19]. A* is easy to implement and incurs less computation overhead. That is why we choose to modify it to find an obstruction-free, shortest path for each customer. By introducing both factors D_k and $T_{\text{avg}}(D_k)$ in Eq. (4), we can tailor A* to fit for the supermarket environment. Experimental results in Section 6 will also show that our modification significantly reduces the overall walking time of a customer to reach his destination.

5 Implementation Details

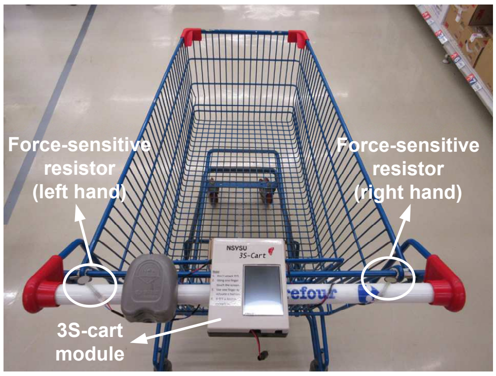
We adopt off-the-shelf, modularized sensors to implement a 3S-cart prototype, which provides easy deployment of the system. In our implementation, XBee [20] is used as a wireless router. It is a ZigBee-compliant device that supports low-power communication in 2.4GHz band. Fig. 3 presents the 3S-cart module installed on a shopping cart, which uses Arduino as the processing platform. Arduino has an ATmega2560 micro-controller with 16MHz clock speed and 256KB flash memory. It uses digital pins to connect with XBee to communicate with wireless routers, and analog pins to connect with an LCD touch panel to interact with a customer. We use TFT touch shield [21] to be the panel, which supports 240×320 resolution and 18bit colors on a 2.8inch diagonal LCD display. Moreover, we adopt two types of sensors to analyze a customer's behavior. Sparkfun FSR-402 [22], a force-sensitive resistor, checks if the customer is holding the cart's handle by detecting the change of its voltage. InvenSense MPU-9250 [23], an accelerometer, estimates the gravitational acceleration along X, Y, and Z axes, which helps derive the cart's moving direction in 3D space. Our module also provides scalability, since it is easy to add extra devices (e.g., bar/QR-code scanner or RFID reader) by connecting them with the Arduino's pins.

As mentioned earlier in Section 3.1, shopping carts and wireless routers form a hybrid WSN via ZigBee communication. We employ the *ad hoc on-demand distance vector (AODV)* protocol for message exchange, which is the mandatory routing protocol specified in ZigBee [24]. Besides, wireless routers will identify the position of every cart in the supermarket. This is achieved by collecting the *received signal strength (RSS)* from each router to estimate the distance between a cart and the router. Then, we adopt the triangulation technique [25] to calculate the cart's position. Moreover, since the LCD touch panel consumes a large portion of energy in the 3S-cart module (discussed in Section 6.2), the cart will switch to the standby state to turn off LCD when it finds that the customer does not use the system. Specifically, based on the readings from the force-sensitive resistors and accelerometer, Arduino can consult Table 1 to determine whether to send a signal to LCD to turn it off.

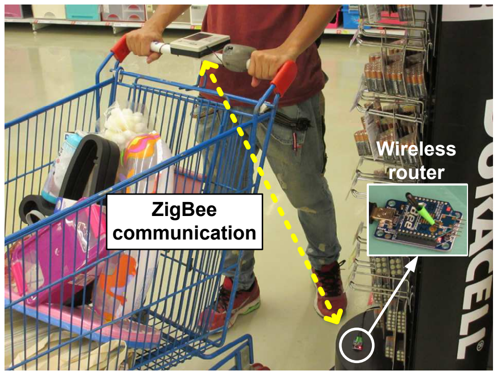
We test our 3S-cart prototype in the Carrefour supermarket, which is demonstrated in Fig. 4. We encapsulate all components of the 3S-cart module in a small plastic box, as Fig. 4(a) shows, which has two advantages. First, our module can connect to an external power supply to substantially extend its usage time. Second, it is simple to install our module on a shopping cart, as presented in Fig. 4(b), which helps the retailer easily build up our system in a supermarket. When a customer is going shopping with a cart containing our 3S-cart module, it can search the wireless router deployed on a nearby product shelf and establish a ZigBee link with the router for communication, as Fig. 4(c) illustrates. Then, according to the customer's behavior, the cart will show product data on its LCD panel. Besides, the customer can query the cart where to find his desired product. Through the enhanced A* algorithm in Section 4.2, our 3S-cart system can compute a comfortable path for the customer to walk to his destination by detouring obstructions and most crowds. However, due to the LCD display size, we guide the customer by showing an *arrow* to indicate the moving direction, as shown in Fig. 4(d). When the system recomputes a new walking path for that customer (e.g., due to the movement of crowds), the cart can adaptively adjust the arrow to guide the customer to the new path.



(a) package all components together in a small box



(b) install the 3S-cart module on a shopping cart



(c) the cart communicates with a wireless router

(d) LCD shows product data and moving direction

Fig. 4: Our 3S-cart implementation in the Carrefour supermarket.

6 Experimental Results

This section conducts experiments on the 3S-cart system to evaluate its performance. We first assess the walking path calculated by the product-navigation application in Section 4.2. Then, we measure the amount of energy consumed by each component of the 3S-cart module in Section 5.

6.1 Walking Paths of Customers

Based on the layout of Carrefour supermarket, we divide it into 27×15 grids and randomly select the locations of customers and their destinations. There will be 100 to 600

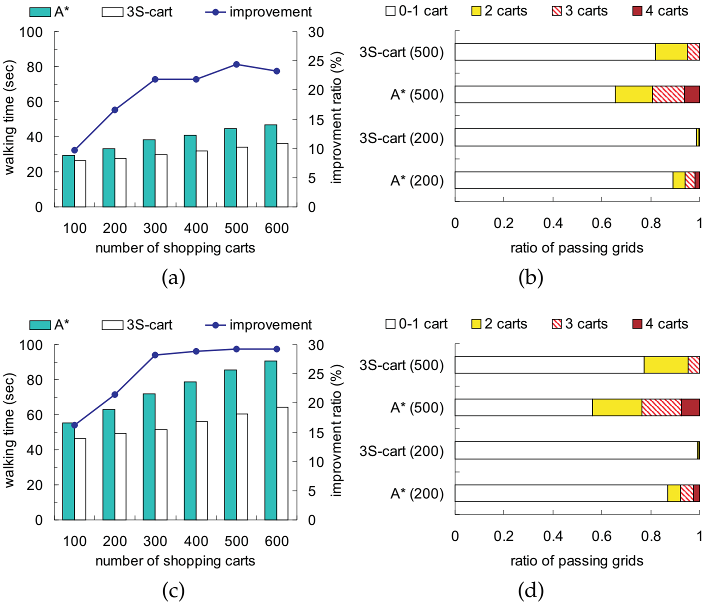


Fig. 5: Performance evaluation of the product-navigation application: (a) the average walking time of customers in the SM scenario, (b) the ratio of grids constituting a customer's path in the SM scenario, (c) the average walking time of customers in the LM scenario, and (d) the ratio of grids constituting a customer's path in the LM scenario.

TABLE 2: Energy consumed by each component in the 3S-cart module.

Component	Working state	Standby state
Resistors	0 ∼ 0.05W	0W
XBee	0.288W	0.288W
Accelerometer	0.012W	0.012W
LCD	1.1W	0W
Total	1.4 ∼ 1.45W	0.3W

shopping carts in the supermarket. Two scenarios are considered to evaluate the walking paths of customers. In the *short-movement (SM) scenario*, the distance between a customer and his destination is no larger than eight grids. In the *long-movement (LM) scenario*, this distance ranges from 15 to 20 grids. We compare 3S-cart with the A* algorithm, and measure the average walking time of customers and the ratio of grids which constitute a customer's path (with 200 and 500 carts).

Fig. 5 presents the experiment results. In the SM scenario, 3S-cart saves around 10% ∼ 25% of average walking time than the A* algorithm, as shown in Fig. 5(a). The reason is that 3S-cart assigns a higher cost for the grids with more shopping carts. In this way, it can avoid adding those congested grids to the walking path. As Fig. 5(b) illustrates, A* will include more grids with $D_k \geq 3$ if they constitute the shortest path between the customer's location and his destination. Nevertheless, it greatly increases the walking time for that customer, thereby degrading his shopping experience. On the other hand, 3S-cart reduces 15% ∼ 30% of average walking time comparing with A* in the LM scenario, as shown in Fig. 5(c). Similarly, from Fig. 5(d), we can observe that A* includes more congested grids in the LM scenario, especially when 500 carts roam in the supermarket. These results verify that 3S-cart can adaptively schedule a comfortable walking path for each customer to detour most crowds in the supermarket, as compared with the traditional A* algorithm.

6.2 Energy Consumption of 3S-cart Module

In this experiment, we measure the energy consumption of each individual component of the 3S-cart module in Fig. 3. Since the Arduino platform connects all other components and also provides operating power to them, it is difficult to measure the amount of energy consumed by Arduino alone. Thus, Table 2 gives the energy consumption of each component, except for Ardunio[2].

The force-sensitive resistors consume energy only when the customer is pressing them. In this case, they spend at most 0.05W. Both the XBee device and accelerometer will keep operating in our system, which spend 0.288W and 0.012W per hour, respectively. From Table 2, it is obvious that the LCD touch panel consumes the most part of energy in the 3S-cart module, where it requires 1.1W for operation (i.e., occupying around 78.57% energy consumption of the whole module). That is why we let the module switch to the standby state when it detects that the customer is not using it (referring to Section 4.1). In this way, the overall energy consumption of the 3S-cart module will decrease from 1.4W to only 0.3W.

We also use a single 9V battery to measure the total operating time of the 3S-cart module. Specifically, if the module always keeps in the working state, the lifetime will be 83 minutes. However, if we allow the module to periodically enter the standby state for power saving, it can significantly extend the lifetime to 386 minutes. This experiment demonstrates the necessity of using the standby state.

7 Conclusion

This paper designs a 3S-cart system to integrate the WSN technology with shopping carts to support smart shopping. It uses natural actions of customers on carts to infer their behavior and provides real-time interaction to improve the shopping experience. 3S-cart can also cooperate with other systems such as membership, visual surveillance, bar/QR-code, and RFID. Two applications, sales-promotion and product-navigation, are developed to show the practicability of 3S-cart. Furthermore, we realize the proposed system by implementing a prototype and deploying it in the Carrefour supermarket.

References

[1] Y.C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, vol. 47, no. 1, pp. 12:1–12:36, 2014.
[2] L.D. Xu, W. He, and S. Li, "Internet of Things in industries: a survey," *IEEE Trans. Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
[3] Y.C. Tseng, Y.C. Wang, K. Y. Cheng, and Y. Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.
[4] Arduino. [Online]. Available: http://www.arduino.cc/
[5] Y. Zhang, L. Wang, R. Hartley, and H. Li, "Handling significant scale difference for object retrieval in a supermarket," *Proc. Digital Image Computing: Techniques and Applications*, 2009, pp. 468–471.
[6] M. Popa, L. Rothkrantz, Z. Yang, P. Wiggers, R. Braspenning, and C. Shan, "Analysis of shopping behavior based on surveillance system," *Proc. IEEE Int'l Conf. Systems Man and Cybernetics*, 2010, pp. 2512–2519.
[7] M. Popa, L. Rothkrantz, C. Shan, and P. Wiggers, "Assessment of customers' level of interest," *Proc. IEEE Int'l Conf. Image Processing*, 2012, pp. 41–44.
[8] J. Han, H. Ding, C. Qian, D. Ma, W. Xi, Z. Wang, Z. Jiang, and L. Shangguan, "CBID: a customer behavior identification system using passive tags," *Proc. IEEE Int'l Conf. Network Protocols*, 2014, pp. 47–58.

2. In fact, the energy consumption of each component includes a part of energy spent by Ardunio.

[9] T. Liu, L. Yang, X.Y. Li, H. Huang, and Y. Liu, "TagBooth: deep shopping data acquisition powered by RFID tags," *Proc. IEEE INFOCOM*, 2015, pp. 1670–1678.

[10] Y. Guo, L. Yang, B. Li, T. Liu, and Y. Liu, "RollCaller: user-friendly indoor navigation system using human-item spatial relation," *Proc. IEEE INFOCOM*, 2014, pp. 2840–2848.

[11] S. Gai, E.J. Jung, and B.J. Yi, "Localization algorithm based on ZigBee wireless sensor network with application to an active shopping cart," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems* , 2014, pp. 4571–4576.

[12] S.R. Rupanagudi, F. Jabeen, V.R. Savarni, S. Adinarayana, V.K. Bharadwaj, R. Karishma, and V.G. Bhat, "A novel video processing based cost effective smart trolley system for supermarkets using FPGA," *Proc. Int'l Conf. Comm., Information & Computing Technology*, 2015, pp. 1–6.

[13] P. Lopez-Iturri, L. Azpilicueta, J.J. Astrain, E. Aguirre, E. Salinero, J. Villadangos, and F. Falcone, "Implementation of wireless sensor network architecture for interactive shopping carts to enable context-aware commercial areas," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5416–5425, 2016.

[14] D. Dardari, P. Closas, and P.M. Djuric, "Indoor tracking: theory, methods, and technologies," *IEEE Trans. Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[15] Y.C. Wang, F.J. Wu, and Y.C. Tseng, "Mobility management algorithms and applications for mobile sensor networks," *Wireless Comm. and Mobile Computing*, vol. 12, no. 1, pp. 7–21, 2012.

[16] Y.C. Tseng, P.Y. Chen, and W.T. Chen, "k-angle object coverage problem in a wireless sensor network," *IEEE Sensors J.*, vol. 12, no. 12, pp. 340–3416, 2012.

[17] D. Hicks, K. Mannix, H.M. Bowles, and B.J. Gao, "SmartMart: IoT-based in-store mapping for mobile devices," *Proc. IEEE Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing*, 2013, pp. 616–621.

[18] W. Zeng and R.L. Church, "Finding shortest paths on real road networks: the case for A*," *Int'l J. Geographical Information Science*, vol. 23, no. 4, pp. 531–543, 2009.

[19] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of Large and Complex Networks*, Springer, 2009.

[20] XBee. [Online]. Available: http://www.digi.com/lp/xbee/

[21] TFT tocuh shield. [Online]. Available: http://www.adafruit.com/

[22] Sparkfun. [Online]. Available: https://www.sparkfun.com/

[23] InvenSense. [Online]. Available: http://store.invensense.com/

[24] ZigBee Alliance. [Online]. Available: http://www.zigbee.org/

[25] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.