



# Robots serve humans in public places—*KeJia* robot as a shopping assistant

Yingfeng Chen, Feng Wu, Wei Shuai and Xiaoping Chen

## Abstract

This paper reports the project of a shopping mall service robot, named *KeJia*, which is designed for customer guidance, providing information and entertainments in a real shopping mall environment. The background, motivations, and requirements of this project are analyzed and presented, which guide the development of the robot system. To develop the robot system, new techniques and improvements of existing methods for mapping, localization, and navigation are proposed to address the challenge of robot motion in a very large, complex, and crowded environment. Moreover, a novel multimodal interaction mechanism is designed by which customers can conveniently interact with the robot either in speech or via a mobile application. The *KeJia* robot was deployed in a large modern shopping mall with the size of 30,000 m<sup>2</sup> and more than 160 shops; field trials were conducted for 40 days where about 530 customers were served. The results of both simulated experiments and field tests confirm the feasibility, stability, and safety of the robot system.

## Keywords

Service robot, quadtree mapping, robot navigation, odometry calibration, human–robot interaction

Topic: Service Robotics

Topic Editor: Marco Ceccarelli

Associate Editor: Erwin-Christian Lovasz

## Introduction and background

With the rapid advance of robotics and artificial intelligence technologies, several well-known robots have been developed with delicate appearances, human-like joints, and intelligent behaviors, such as Willow Garage's PR2,<sup>1</sup> Boston Dynamics' Atlas,<sup>2</sup> and the Meka.<sup>3</sup> Nevertheless, the general public still rarely have the opportunity to personally interact with tangible and advanced robots. Fortunately, *service robots*, a kind of robots aiming to improve the quality of human life, have the potential to fill the gaps between the public and the robots.

A service robot is defined as a robot that could autonomously perform daily services for humans. By virtue of the technology transformation from traditional industrial robots and new explorations toward this direction, many service robot applications appear.<sup>4</sup> Meanwhile, these applications in real-world environments provide occasions for researchers to verify the existing technical approaches. Through analyzing experimental results and learning from the

participants' feedback, researchers could better understand the limitations and weaknesses of current techniques, which ultimately leads to technological advancement.

In this research, the robot is adapted to daily shopping mall scenarios and acts as a shopping assistant, providing guidance service, information service, and entertainments. There is a great motivation to integrate the essential techniques into a public service robot, which have been pursued in *KeJia* project<sup>5</sup> for many years, to verify whether the robot system could work properly in real-world environments and

---

Department of Computer Sciences, University of Science and Technology of China, China

### Corresponding author:

Feng Wu, University of Science and Technology of China, Room 220, Tech. Building, No. 443 Huangshan Road, Hefei 230000, China.  
Email: wufeng02@ustc.edu.cn



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License

(<http://www.creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

furthermore to investigate whether the robot could meet the public expectations for an eligible service robot. The shopping mall is considered as one of the most frequently visited places in routine activities. Generally, the environment of a shopping mall is much more complex and challenging than the laboratory, and it is chosen as the test scenario since it is there where the robot system can be sufficiently exposed to the untrained and nontechnical customers, who could offer valuable suggestions and different views.

## Related work

In the past decades, there are several robot applications that have been successfully deployed in public spaces, for example, museum, exhibition hall, shopping mall, and so on. A pioneering work was carried out by Burgard et al.,<sup>6</sup> in which the *RHINO* robot acted as a guide in the *Deutsches Museum Bonn* for 6 days and gave tours to more than 2000 visitors (Figure 1(a)). They preliminarily integrated mapping, localization, navigation, interaction, and telepresence modules into a complete robot system, making a considerable success and significant impact in robotic research community. Specifically, it primarily aimed at testing the autonomous navigation algorithms in dynamic environments using natural landmarks for robot localization. The *MINERVA* robot,<sup>7</sup> as successor of *RHINO*, with impressive appearance and facial expressions, was developed to enhance the performance in human–robot interaction, which was regarded as the main deficiency of its predecessor. The *HERMES* robot,<sup>8</sup> a prototype of humanoid service robot, also served in a museum for half a year, up to 18 h per day. Due to the robustness and dependability of the system, the robot was able to work stably during long-term operation. In order to improve the human–robot interaction in museums, the *Robotinho* robot<sup>9</sup> addressed the challenge on more intuitive, natural, and human-like interaction by exploiting richer facial expressions and gestures.

In contrast with the aforementioned robots in museums, the work<sup>10</sup> presented *Robox* (Figure 1(b)), a service robot designed for mass exhibition, which was used to guide visitors in the exhibition hall for 159 days and contacted with more than 686,000 people. Besides, the work<sup>11</sup> introduced a service robot in hospital, which was able to accomplish navigation task in hospital environment, as the role of medical supplies transporter.

Shopping is considered to be one of the favorite activities for many people, and therefore, many researchers are interested in this field. In the study of Tomizawa et al.,<sup>12</sup> a shopping robot system was envisaged to help humans accomplish shopping tasks via remote communication. The idea was creative but it still remains at the conceptual stage. Only one part of the system, a suction hand for manipulating fresh foods was implemented. *RoboCart*<sup>13</sup> was designed to help visual impaired people navigate in grocery stores and carry purchased items (Figure 1(c)). The work<sup>14</sup> proposed a shopping assistant robot, providing collision-free guidance and accompanying service. It is noteworthy



**Figure 1.** (a) *RHINO* in museum. (b) *Robox* in exhibition. (c) *RoboCart* in grocery. (d) *TOOMAS* in store.

that the localization function of the robots in the studies by Kulyukin et al.<sup>13</sup> and Shieh et al.<sup>14</sup> was both implemented with the radio frequency identification (RFID) tags, which were installed at various locations in the workspace beforehand. In the study by Kanda et al.,<sup>15</sup> a robot was deployed in a fixed area of a shopping mall without the ability of moving freely, which aimed to entertain customers with natural interaction, such as providing information with speech and giving directions with gestures. Iwamura et al.<sup>16</sup> investigated two design considerations of assistive robots, the experiments were conducted in the shopping scenarios with two types of robots: a humanoid robot and a cart robot. The robot *TOOMAS*<sup>17</sup> (Figure 1(d)) roamed in stores to search the potential customers and guided them to the locations of requested products. The requirements of the shopping scenario were investigated comprehensively and the whole system was implemented systematically, it was even considered as the most practical shopping robots for everyday use so far.

Compared with all the related works mentioned above, the *KeJia* robot system is implemented in more detail and is one of the few works in which field tests had been carried out. In addition, the operating environment is more challenging than most of the existing works, even the work of Gross et al.,<sup>17</sup> in which the robot was exactly deployed in stores and not in shopping malls. Besides, a mobile

application (App) is employed in the system to make the human–robot interaction more convenient.

## Cases summary, requirements, and technical challenges

Service robots in public are expected to play two main roles: tools and partners.<sup>16</sup> In more detail, the robots as tools are expected to offer physical assistance, partially for the disabled and the elder. The works<sup>11,13,14</sup> fall into this category to a certain degree. While robots as partners are expected to interact with people naturally and provide service similar to human companions. The works<sup>6–8,10,17</sup> basically belong to this category. In accordance with the development trend of service robots,<sup>4</sup> a promising service robot in shopping malls should be regarded as a partner instead of simply offering help like in other public facilities.

Indeed, there are strong demands for service robots to provide more convenient service in shopping malls. One very common requirement is to help customers find the locations of their desired shops. The size of a modern shopping mall is becoming larger and larger and it usually contains hundreds of individual commercial tenants. Customers are often confused with dazzling decorations and get lost in the maze-like environment, not to mention finding the way to a certain shop. Sometimes, thumbnail maps and signposts are available, but interestingly, customers still prefer to turn to staff for asking directions.<sup>15</sup> Apart from this, customers may hope to know the shopping information, such as product catalogue, price list, and product discount. Traditionally, the shopping information is often printed on pamphlets and distributed to people, which is inconvenient and not environment friendly. Therefore, obtaining information from a service robot with natural ways might be more attractive for customers. Furthermore, in order to ensure service quality and improve shopping experience, the shopping mall managers have to hire more professional guides, which inevitably increases the labor costs. In this case, robots could be cost-efficient substitutions. Based on the aforementioned requirements in shopping malls, the functions of *KeJia* robot are defined as providing guidance service, information service, and entertainment service. Although every functional capability has been tackled individually in previous research, they still need a significant amount of effort to develop new techniques and methods according to the actual conditions in real shopping malls.

Here, guidance service refers to a robot, moving in front of the users, leading them to their destinations. For robot, it exactly depends on the function of navigation given that the environmental map is known. Specially, robot navigation usually involves the techniques of mapping, localization, path plan, and obstacle avoidance. In the literature, there are two very common solutions to implement robot navigation. The first method needs modifications of the operation environments, making them adapt to the robot application. For instance, in the studies by Kulyukin et al. and Shieh et al.,<sup>13,14</sup> a net of RFID tags is deployed in the operating

area, which facilitates robot to locate itself and follow the routes. The other one is to exploit the onboard sensors exclusively; however, it requires more advanced navigation solutions.<sup>6,7,17</sup> In the shopping mall scenario, the first method is unacceptable in practice because additional facilities or environmental modifications are not allowed by shop mall managers, even if they would not result in substantial influence on original environments. Furthermore, it will dramatically increase deployment cost of the system and even worse the deployment may need to be redone once the environment is significantly changed.

Consequently, the second solution was chosen in this robot system, that is, only the use of the onboard sensors. Though the related algorithms in robotic area are considerably mature, it is nontrivial how to implement a robust and safe navigation system in a realistic shopping mall environment. The greatest challenge is that the size of the operation environment (e.g. more than 30,000 m<sup>2</sup> in this case) is much larger and more complex than most existing works. For mapping, the commonly used algorithms are based on particle filter, which are confronted with prohibitive storage consumption in a large environment. Therefore, a quadtree map presentation is proposed and integrated with *Rao-Blackwellized particle filter* (RBPF) based on simultaneous localization and mapping (SLAM)<sup>18,19</sup> to solve this problem. For localization, the Monte Carlo localization (MCL)<sup>19</sup> is improved according to different environmental conditions, making the localizer less susceptible to dynamic environments. For navigation, the traditional methods can hardly work smoothly in dynamic and intricate environments, thus an improved navigation framework and a local controller are designed to enhance the performance of navigation.

Human–robot interaction is indispensable for a service robot in shopping malls, and customers obtain information and entertainment from the interaction process. Speech-based interaction was adopted in *KeJia* robot as well as in some previous works.<sup>15,16</sup> It is considered as the most ideal way of human–robot interaction because it is also the primary form of communication between humans. Due to the complexity of speech recognition under noise background and natural language understanding, robust speech–base interaction is quite challenging in shopping malls. Therefore, in order to simplify the interaction, a chatting system for handling the failures of normal speech conversation is designed. Additionally, a mobile application is developed, which serves as an alternative interaction interface between the robots and customers.

In general, the main contributions of this work can be summarized as follows:

- The RBPF SLAM algorithm based on quadtree map representation is proposed; it greatly reduces the memory consumption when mapping large environments.
- The MCL is substantially improved by selecting different strategies under different situations, making it more robust in crowded environments.

- A safe navigation system in human populated environments is designed and it achieves a good practical effect in the shopping mall.
- A multimodal interaction mechanism for a robot shopping assistant is designed, which enhances the experience of shopping for customers.
- The whole robot system is well integrated and fully tested in shopping malls. The firsthand feedback from the customers in field trials are collected and the results from this study are encouraging.

The remainder of this article is organized as follows. To start, the hardware and software design of the *KeJia* robot are introduced. Next, an overview of the services provided by the robot is given. Then, the navigation system including mapping, localization, and local planner is presented with emphases on the new proposed improvements. This is followed by a description of the interactive module. After that, the results of simulated experiments and field tests are reported. Finally, conclusions and future works are presented.

## Design of *KeJia* robot

The *KeJia* robot designed for the shopping mall scenario is shown in Figure 2. This prototype is based on the design of the domestic service robot that won the world champion of the RoboCup@Home competition in RoboCup 2014. The following sections will give more details on the hardware and software designs, respectively.

### Hardware architecture

The appearance of *KeJia* is that of a young lady, dressed on a traditional suit. With the height of 165 cm, the robot is comparable to a professional shop assistant. The design philosophy of *KeJia* is that a service robot with anthropomorphic shape encourages people to interact with it in a natural way. Cartoon appearance is also popular in the exterior design of service robots, which is attractive especially for children, but from previous experience, children tend to be the sources of trouble for robots in public, their curiosities often drive them to have unexpected behaviors to robots. Touch screen is a frequently used interaction interface with robots, but it is not adopted in *KeJia* robot. Touch screen often requires that the robot is stationary, which means that people hardly interact with robots when robots are moving. In *KeJia* system, customers could interact with it by speech all the time. Besides, the mobile App provides additional method to interact with robots.

A sound equipment embedded in the base is used to play the voice generated by speech synthesis module. The whole robot is motorized by two differential wheels installed on their middle axis and a castor is mounted on the rear. Although it is not as flexible as omnidirectional wheels, it is able to spin on the spot, this endows *KeJia* a good



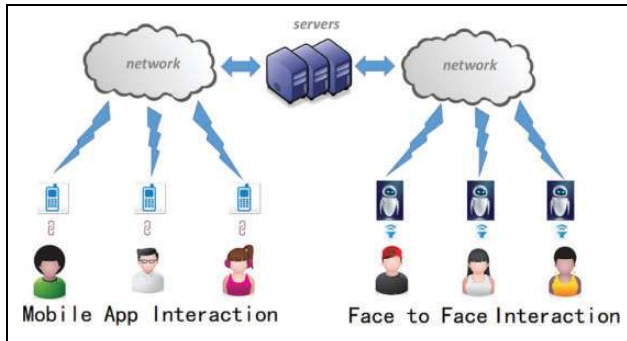
**Figure 2.** (a) *KeJia* robot. (b) Laser scanner. (c) Robot chassis. (d) Differential wheels.

maneuverability and stability when trapped in crowds. The main sensor of the robot is a Hokuyo UTM-30LX (Tokyo, Japan) laser scanner, which feeds the distance data of obstacles around the robot (maximum distance 30 m) to other software modules (e.g. navigation and localization). A pinhole camera and a directional microphone are installed behind the bow tie, which are invisible from outside. The camera is able to capture 30 images every second with a resolution of  $1280 \times 720$ . The directional microphone only captures the voice from a specific orientation; it avoids noises in the input voice as much as possible, therefore the customers are advised to talk in front of the robot.

There is a hidden frame made by aluminum alloy and plastic shells, which constitutes the body of the robot. The head and two arms are fixed on the interior frame. The arms with four degree of freedoms (DOFs) can make some gestures, like greeting people, pointing directions, and so on. The hands are made up of silicone without any degree of freedom. The two DOFs (pan-tilt unit) head can freely turn to a wide range of directions. Every motor on the robot is an individual controller area network (CAN) node; thus, they make up a CAN control network. All the external devices are linked to a universal serial bus (USB) hub centrally, the hub is connected to a laptop computer embedded in the chassis, and the computer processes the data from sensors and performs all online computation.

### Software design

**Overall architecture.** The system mainly consists of three parts: robots, central server, and distributed mobile Apps. Robots could be deployed on different floors in the shopping mall, which are directly faced with customers with the speech interaction. The mobile software can be



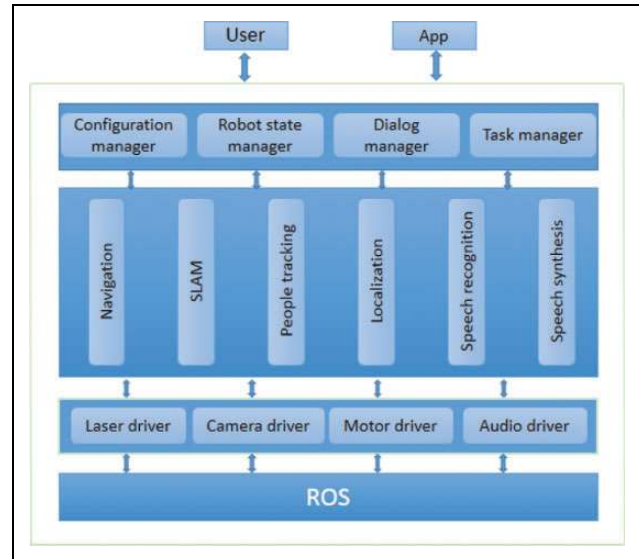
**Figure 3.** Concept map of the proposed system.

downloaded and installed freely on mobile phones, customers obtain running information from them, and they interact with robots through internet. The central server is responsible for maintaining data and building the bridge between robots and mobile Apps. The robots and mobile Apps are allowed to run multiple instances simultaneously, and all of them are connected to a sole central server (see Figure 3).

The data exchanged between the robot and the mobile software are classified into three categories: running information, configuration data, and human–robot interaction data. The running information is about the current states of the robot, including robot position, task state, and hardware state, which are reported periodically by the robot and forwarded from the server to mobile clients. Configuration information is the description of the shopping mall, including environmental map, shops’ positions, shops’ introductions, and so on. Interaction data include the text strings and speech data, which would be processed by the dialog system on the server preliminarily. If the final result of several round interactions turns out to be a task command that is feasible for the robot to carry out, the task will be sent to the robot.

There are two advantages of this structure. Firstly, the configuration data of shopping malls are constantly changing due to the alterations of shops; it would be convenient to modify the central stored data on the server, what the robots need to do is to regularly update the latest data from the server. Secondly, interaction via mobile phone makes the robot reach its greatest potential at the operation time, while other customers can still keep chatting with robot during its navigation period, which increases the customers’ engagement with the robot.

**Modules on robot.** *KeJia* adopts a flexible four layered software architecture (see Figure 4) to meet the requirements of an integrated robot system, such as reliability, extensibility, maintainability, and customizability. The lowest layer is the robot operating system (ROS),<sup>20</sup> which provides a set of robotic software libraries and reliable communication mechanism for modular nodes. The second layer mainly contains hardware drivers. The laser and camera drivers are in charge of packaging the raw sensor data to standard



**Figure 4.** Software structure of the modules on robot.

formats of ROS messages and publish them to upper layers; the motor and audio drivers interpret the command messages from upper layers to hardware executors. The next layer is the most important one in the structure; all the proper skills of a classic robot are placed here, such as mapping, localization, navigation, people tracking, speech recognition, and speech synthesis, which directly decide which functionalities could be implemented in the upper applications. The highest level includes task manager, configuration manager, dialog manager, and robot state manager. The dialog manager module attempts to understand users’ intentions by speech interaction and translates them to some explicit tasks. Configuration manager guarantees the consistency of the configuration data between local and remote, which will synchronize the data when remote configuration data is modified. Robot state manager monitors the states of both hardware and software constantly, which will raise an alarm when exceptions occur. In addition, it reports state information to server. Task manager is responsible for task scheduling and behavior switching, which receives tasks from server or dialog manager and decides what to perform next.

By using such a layered structure, for adding any new application or new skill, only the corresponding layers need to be changed rather than the whole system. Moreover, individual modules can be assigned to different programmers who just develop the desired functionality according to the preestablished interfaces without caring the implementation details of others.

**Modules on server.** The server is an important part in the whole robot system, which includes three basic modules: permission manager, information manager, and dialog manager. The dialog manager on the server is an interaction module designed for the mobile App, which processes the

speech and texts from the mobile clients. If the input is voice segments, it firstly translates them to texts by a build-in speech recognition module. The texts will be put into the speech dialog manager, which will keep on the conversation with response texts. The information manager loads configuration data from database and waits for the data requests from the robot. When a request is received, the data connection will be established and configuration data will be sent to robot. Meanwhile, it also pushes the robot state information to the online mobile clients. The permission manager decides whether a mobile client could get the control right of the robot. If the client is a normal user, the request will be ignored when the current task is running. But if the mobile client is identified as the shopping mall manager, it would be entitled to interrupt the current task and get control of the robot immediately.

## Overview of robot services

Prior to the introductions of navigation and interaction techniques implemented in the current version of the robot in the following sections, a brief overview of the already realized service functionalities is given. Before starting to work, the shopping mall manager powers the robot on and controls it to the initiation location with a joystick. The initiation location is marked with rectangular line on the ground, which facilitates the startup of the localization module with known coordinates. Once all modules start properly without any errors, the robot will enter into the patrol pattern, moving along the regular route that is defined by a series of fixed waypoints. The route could be modified by choosing different waypoints by a special tool. When a predefined waypoint is arrived, the robot stops and waits for potential customers. Customers can stand in front of the robot and attract the attention of the robot by speech, like saying "hello" in Chinese. Once the robot perceives that someone tries to start an interaction, it begins to confirm the caller's position and turns the head to that direction. Naturally, the robot will continue the conversation until the customer raises the request to a desired shop and then the robot will begin guidance to the desired location. Otherwise, if customers do not need a guidance, the conversion will end up by saying "goodbye." During the guidance, robot moves in front of the people with an appropriate speed (0.3–0.6 m/s), which is comfortable for customers to keep pace with the robot; meanwhile, it will give an introduction of the shop. When arriving at the destination, the robot will notice the customer of the arrival and it points to the direction of the shop. If there is no further request, it will end up this guidance successfully and convert back to the patrol mode to the nearest waypoint.

On the mobile clients, the customers could connect to the network in the shopping mall and download the mobile App by scanning two-dimensional barcode. The robot pose is shown on the information screen overlaying with the

environmental map; shops are also displayed on the map and can be searched by name. On the chatting screen, customers could type words in the input box or press voice button to communicate with the dialog manager on server. If the robot is occupied in guidance task, it will be prohibited for the mobile App to issue any commands to the robot. In spite of this, customers could still obtain enough prompt of how to reach the shop by virtue of the annotations in the map. The shopping manager could login the mobile App and monitor the states of all robots. Before getting off work, all the running tasks will be terminated by mobile phone. If the manager is nearby the robot, he could control the robot back to the charging area with joystick manually; otherwise, the homeward commands will be sent to robot by mobile App.

In general, the robot performs guidance tasks and interacts with customers autonomously, once it starts up normally. All the operations on the robot are self-explanatory with voice cues. In most cases, robot's behaviors are cohere with customers' expectations, so it is not necessary to give any antecedent instruction or teaching to customers.

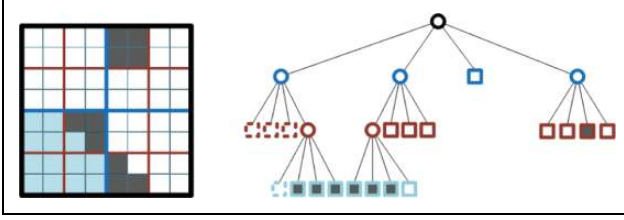
## Navigation system

Navigation system is the core functional module, which provides robot the guidance ability in complex and large environments. It is not an individual module, but a collection of related softwares, including mapping, localization, path planning, and local controller. In this section, the details of the improvements in each submodule will be presented under their backgrounds.

### Mapping of large environments

There are already many mature algorithms to the SLAM problem, most of them take the grid map as the representation of environments. However, the memory exhaustion problem emerges along with the increasing enlargement of mapped environments, especially for particle filter family,<sup>18,19,21</sup> in which every particle is generally associated with an individual map. The size of the shopping mall is usually more than 10,000 m<sup>2</sup>, which makes the situation worse, therefore, a quadtree map representation is introduced to settle this issue.

*Quadtree representation.* A quadtree is a tree data structure in which each internal node has exactly four children, each child indicates an equal size region divided from parent area. It is often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. In the context of robotic mapping, as usual, the cell area of a map is measured as three states, that is, occupied, free, and unknown. In real-world situation, the range sensors in two dimension are usually just able to detect the surface profile of obstacles and the interiors are assumed to be unknown. In addition to this, the mapped environments are



**Figure 5.** Grid map (left) and quadtree (right). Squares are leaf nodes and circles are internal nodes. Squares (dotted line border) represent unknown areas which actually are nulls, squares (filled with grey color) represent occupied areas, and squares (solid line border, not filled) represent free areas.

generally spacious enough for the robot to move, particularly in a shopping mall. Either of the two cases result in plenty of group grid cells with the same state of unknown or free, which provides the opportunity to reach the full potential of the quadtree representation.

As shown in Figure 5, an example of a quadtree is compared with a traditional grid map. The nodes stand for different areas by their positions in the quadtree. The nodes at the same depth have the same area size, the areas represented by the nodes that are at adjacent depth have double relationship. A region containing inhomogeneous states will be recursively divided into smaller quarter areas until the children’s size reaches a predefined resolution. All the leaf nodes in the tree keep a simple Boolean value indicating whether the area is occupied. If some nodes are non-existent in tree, it means that these areas are still in an unknown state so far. Compared with the grid map, a leaf node that is not at the lowest level can be on behalf of at least four or more grids that are in same state, and the unknown areas do not take up any node in the tree.

**Quadtree access acceleration.** The quadtree is an ideal map representation for mapping as described above, but as a kind of tree data structures, it cannot get rid of the intrinsic shortage—low access efficiency. The time cost of visiting a node in the quadtree usually contains two parts: downward searching from root and branch selecting at intermediate nodes. The time cost of downward searching is often with low proportion of total time and varies with the height of tree. The branch selecting operation is to compare the intended node’s position to the midplane position of current node, which is often time-consuming. To accelerate the speed of node access, a locational code mechanism is usually adopted, such as Morton code or Z-order code.<sup>22</sup> Inspired by this method, a kind of new coding scheme between node’s position and access key in pointer-based quadtree is designed, which can substantially decrease the access time in quadtree.

In the proposed method, access key of a node in quadtree is represented as a triple  $(K_x, K_y, \text{depth})$ . Since the map only has two dimensions, the  $K_x$  and  $K_y$  are the sub-components of the access key in each dimension, respectively, and the depth indicates the height of the node in quadtree. The number of valid bits in its access key is equal

to the depth and encoded from left to right in order. The conversion between access key and node’s position in one dimension is shown in Figure 6. Taking the leaf node A as example, given its access key  $K_x = 00000001$ , the size of the whole space  $[0,1)$  (denoted as  $|S|$ , the left boundary of the space is written as  $S_{\text{begin}}$ ). Now the  $x$  position (denoted as  $P_x$ ) of node A can get from equation (1),  $P_x = 1/2^2 \times (1 + 0.5) + 0 = 0.375$ . The node A is in depth 2, hence it represents the interval  $[0.25, 0.5)$ , which is occupied. Equation (1) can also be utilized to encode the access key form position reversely.

$$P_{x,y} = |S|/2^{\text{depth}} \times (K_{x,y} + 0.5) + S_{\text{begin}} \quad (1)$$

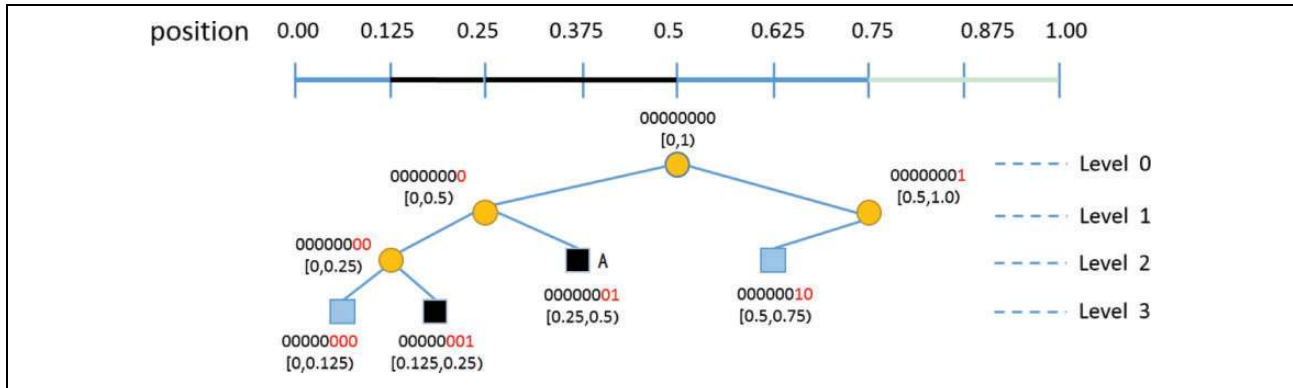
Here, the properties of the proposed access key are analyzed. Firstly, the mapping from every node’s position to its access key is unique, which guarantees the availability of the coding scheme. Secondly, exploiting corresponding bits in  $K_x$  and  $K_y$ , the right branch can be selected directly without any additional comparison. The  $K_{x,y}^i$  is the  $i$ th bit in  $K_x$  or  $K_y$ , the number of total combinations of  $K_x^i$  and  $K_y^i$  is four and each one corresponds exactly to a branch in the quadtree. Locating a node in the quadtree needs to go downward depth times, at  $i$ th time, the branch is chosen according to  $b_i \in \{0, 1, 2, 3\}$  ( $b_i = 2 \times K_x^i + K_y^i$ ), which is the core of this method. For example, given the completed access key of a node in quadtree ( $K_x = 00000010$ ,  $K_y = 00000001$ , depth = 3). From equation (1), the area it represents is the square area  $\{(x, y) | x \in [0.25, 0.375), y \in [0.125, 0.25)\}$ . To visit this node, the sequence of chosen branches are branch 0, branch 2, and branch 1 in the quadtree. Besides the access key can be derived from its parent node by equation (2), this feature can simplify the encoding process.

$$\begin{cases} K_{x,y}(\text{left\_child}) = 2 \times K_{x,y}(\text{parent}) \\ K_{x,y}(\text{right\_child}) = 2 \times K_{x,y}(\text{parent}) + 1 \\ \text{Depth}(\text{children}) = \text{Depth}(\text{parent}) + 1 \end{cases} \quad (2)$$

The robotic SLAM problem can be modeled mathematically as how to estimate the joint posterior distribution of the map  $m$  and the pose  $x_{1:t} = x_1, \dots, x_t$  of the robot, given its sensor data  $z_{1:t} = z_1, \dots, z_t$  and control data  $u_{1:t-1} = u_1, \dots, u_{t-1}$ . This problem is further factorized to equation (3) in the study by Murphy,<sup>23</sup> which enables us to use particle filter to track potential trajectories of the robot and each particle maintains its own map created along the trajectory

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(m | x_{1:t}, z_{1:t}) \times p(x_{1:t} | z_{1:t}, u_{1:t-1}) \quad (3)$$

Integrating with quadtree will not impact the procedures of RBPF, which includes four steps: sampling, importance weighting, resampling, and map updating. Due to the merit of quadtree representation, particles require less memory when they are associated with quadtree maps. According to



**Figure 6.** The whole space is  $[0, 1]$  with 0.125 resolution, squares are leaf nodes and circles are internal nodes. Black squares represent occupied ranges and blue squares represent free ranges. The access key is coding with 8 bits (maximum) and the valid bits of key are marked with red color on the right. The ranges represented by the node is associated with the access key.

this principle, the quadtree SLAM is implemented based on the open source project Gmapping.<sup>24</sup>

## Localization in complex environment

Accurate localization is the prerequisite of safe and credible navigation behavior. Robot localization has been addressed in many studies.<sup>19,25,26</sup> Kalman filter is first proposed and performs well with landmark maps in many cases, but it is hard to handle the global localization problem. This problem can be solved by Markov localization, which tracks the belief of robot's position over all possible space. However, Markov localization is limited by the size of discrete states (usually the grids in map) and not applicable for large environments with high resolution. The Monte Carlo localization is the state-of-art method for robot localization, which could deal with various noises and concentrate the computation in the most possible areas. However, it performs badly in dynamic environments, where sensor data is often corrupted.

In the real shopping mall, dynamic environments are due to the shop decoration, temporary stalls (Figure 7(d)), advertising board replacement, infrastructure improvement (Figure 7(b)), and so on. All of these belong to gradual changes because they change slowly with time. The other kind of dynamic change is the flow of crowds, which often make the environments change quickly and hence they will be called rapid changes. To cope with these situations, the MCL is adopted as the fundamental method with strategy selection in different dynamic conditions, making it more adaptive and stable in highly dynamic shopping malls. Meanwhile, odometry calibration is applied to ensure the robot will not completely lose its position rapidly in exceptional cases.

### Odometry calibration

The typical and thorny dynamics in shopping malls is the dense streams of people, which often blocks most of the laser beams, making the sensor data severely polluted. Worse is that sometimes people like to stand around the

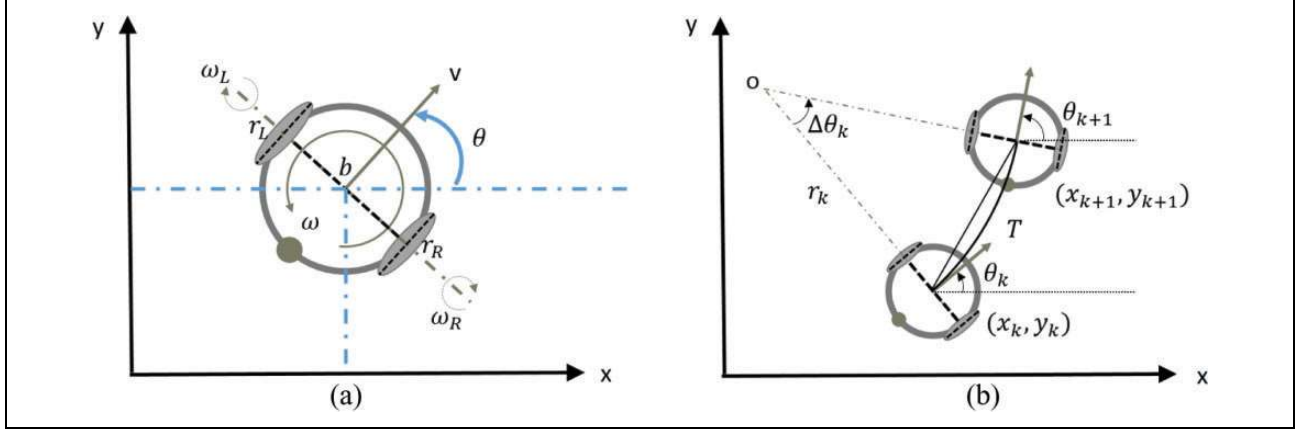


**Figure 7.** (a) Typical shopping mall scenarios with crowds. (b) Newly added resting chair. (c) Glass wall. (d) Temporary stall.

robot and watch other customers interacting with robot; the laser data are totally inconsistent with the pre-built map (Figure 7(a)). On such occasions, the most reliable perception comes from the internal data, that is, odometry. The importance of odometry has been already stressed in many works.<sup>27-29</sup> It is considered as the most cheap, stable, and self-sustaining positioning method. Even in many latest robotic techniques, it plays an indispensable role as a priori knowledge. But odometry often suffers from kinds of systematic and nonsystematic errors, resulting in a significant decrease in performance.<sup>27</sup> Thus, odometry calibration is elaborately designed and carried out to guarantee the accuracy of blind localization. The base of *KeJia* is a differential drive structure (Figure 8(a)), and the odometry of such structure can be calculated from the encoder data of two wheels without any numerical approximation (Figure 8(b)).

Ignoring the nonsystematic errors in odometry (such as wheel slippage, uneven floor, castor wheels, etc.), the following equation can be obtained:





**Figure 8.** (a) Differential drive structure with relevant variables. (b) Computation of odometry between two successive poses.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = C \times \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad C = \begin{bmatrix} \frac{r_R}{2} & \frac{r_L}{2} \\ \frac{r_R}{b} & \frac{-r_L}{b} \end{bmatrix} \quad (4)$$

In equation (4), the  $v$  and  $\omega$  indicate the translational and angular velocities of robot, respectively; the  $\omega_L$  ( $\omega_R$ ) and the  $r_L$  ( $r_R$ ) are the rotate speed and the radii of left (right) wheel, respectively;  $b$  is the distance between two wheels; and the  $C$  is the parameter matrix. The odometry of the robot is calculated by discrete-time integration of equation (4), namely

$$\begin{cases} \Delta X_k = X_{k+1} - X_k = T v_k \cos\left(\theta_k + \frac{T w_k}{2}\right) \\ \Delta Y_k = Y_{k+1} - Y_k = T v_k \sin\left(\theta_k + \frac{T w_k}{2}\right) \\ \Delta \theta_k = \theta_{k+1} - \theta_k = T w_k \end{cases} \quad (5)$$

The aim of odometry calibration is to determine the parameters (i.e.,  $r_L$ ,  $r_R$ , and  $b$  in matrix  $C$ ) of the kinematic model, and the deviations of these parameters cause most of the systematic errors. Equation (5) shows that once the accurate measurements of robot's movements as well as the encoder data are known, the relevant parameters will be figured out by data fitting techniques. Via proper formula manipulation, the linearity of the parameter matrix  $C$  can be exploited, converting the calibration to a least-squares estimation problem<sup>30</sup> as following

$$\begin{bmatrix} \theta_{N_1,1} - \theta_{0,1} \\ \vdots \\ \theta_{N_p,p} - \theta_{0,p} \end{bmatrix} = \begin{bmatrix} \Phi_{\theta,1} \\ \vdots \\ \Phi_{\theta,p} \end{bmatrix} \times \begin{bmatrix} C_{2,1} \\ C_{2,2} \end{bmatrix} \quad (6)$$

$$\Phi_{\theta,p} = T \left[ \sum_{i=0}^{N_p-1} w_{R,i} \sum_{i=0}^{N_p-1} w_{L,i} \right]$$

In equation (6),  $\theta_{0,p}$  and  $\theta_{N_p,p}$  are the robot's directions at 0 and  $N_p$  moments in the  $p$ th trajectory sample, and the  $p$ th

trajectory contains the encoder data of  $0, 1, \dots, N_p$  moments (totally  $N_p + 1$  moments in this trajectory, namely,  $N_p$  equal intervals with time  $T$ ). The  $\Phi_{\theta,p}$  is the overall angular change of the  $p$ th trajectory sample, which is the sum of the minor changes in all  $N_p$  intervals  $\sum_{i=0}^{N_p-1} \Delta\theta_i$ . Therefore, a deterministic regressor for this problem could be established by sampling  $p$  trajectories. In the same way, the rest of the parameters in  $C$  would be identified through equation (8).

Thus far, the calibration procedure is elicited from equations (6) and (8), but it needs to obtain the precise measurements of robot's actual movements, which is an unavoidable problem in practical terms. In the realization, a global motion capture system (MoCap) is utilized to obtain the ground truth of robot's movements, which could provide high accuracy and real-time measurements. As far as we know, this may be the first work to introduce such a measuring equipment into robotic calibration.

### Strategy selection

The principle of the strategy selection is that the localizer tends to trust the odometry rather than particle filter in highly dynamic cases, because the odometry do not depend on observations and thus diverges slowly, but wrong observations may misguide the particle filter to an unrecoverable situation quickly. In order to quantify the degree of dynamics in environments, the polluted rate of laser data at position  $p$  (assuming  $p$  is the true position of robot) is defined as  $\sigma_p$

$$\sigma_p = \sum_{i=0}^N \Gamma(|M_p(i) - E_p(i)|/E_p(i) < \gamma) / \text{Num} \quad (7)$$

In equation (7), Num is the total number of laser beams and the  $M_p(i)$  is the measured distance of the  $i$ th beam in laser data, while the  $E_p(i)$  is the expected distance of the  $i$ th beam in laser data when given a position  $p$  and map.  $\gamma$  is usually set to  $[0.1, 0.25]$ , which indicates maximum

permissible range error.  $\Gamma$  is a Boolean function and  $\Gamma(x) = 1$  if  $x$  is true otherwise  $\Gamma(x) = 0$ .

When  $\sigma_p$  is below a threshold value (e.g., 60%), it is considered that the robot is entering exceptional cases. In this condition, the update phase in MCL is skipped since the matching score calculated from highly corrupted observations would be nonsense. The prediction phase is executed as usual and more particles are generated to depict the gradually diverging distribution of the robot position. Once the robot gets out of the dilemma, the MCL procedures would return to normal and particles would tend to converge to the true position.

The localization on *KeJia* is improved based on the implementation of the adaptive particle filters technique.<sup>31</sup> The original implementation has two shortcomings: (1) The pose is given without confident degree of the outcomes. As a result, the robot is unsure of whether it should believe the localization result in navigation, which may lead the robot to dangerous situation. Obviously,  $\sigma_p$  could be the indicator of localization confidence. (2) The particles are generated based on the probabilistic motion model, and the best particle is chosen to represent the most likely pose of robot. Even so, the best particle may not represent the exact pose due to the random sampling error.

Therefore, the process of scan match is taken on the best particle to refine the last result.

### Navigation control

The early works<sup>32–34</sup> have proposed many methods to achieve safe navigation in dynamic environments, the common idea is to decompose the navigation module into two layers, global planner and local controller. The global planner aims to find an end-to-end path from current position to the goal according to some criterions, like avoiding collision and decreasing distance traveled. The local controller is to determine immediate action for robot actuators, which reacts to dynamic changes and ensures the safety. The recent works<sup>35–39</sup> are prone to develop human-friendly navigation under the coexistence with humans, which tries to make robots behave more natural within their abilities. The navigation of *KeJia* continues to adopt the layered framework and emphasizes on handling of the rapid changes caused by customers. The rapid changes affect both the global path planner and the local controller, and the proposed method could enhance the efficiency of path planning and the natural of behaviors.

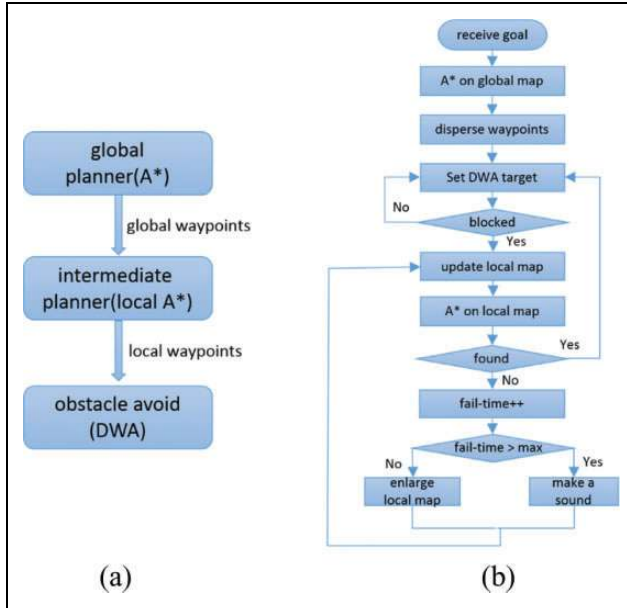
$$\begin{bmatrix} X_{N_p,1} - X_{0,1} \\ Y_{N_p,1} - Y_{0,1} \\ \vdots \\ X_{N_p,p} - X_{0,p} \\ Y_{N_p,p} - Y_{0,p} \end{bmatrix} = \begin{bmatrix} \Phi_{xy,1} \\ \vdots \\ \Phi_{xy,p} \end{bmatrix} \times \begin{bmatrix} C_{1,1} \\ C_{1,2} \end{bmatrix} \quad \Phi_{xy,p} = T \times \begin{bmatrix} \sum_{i=0}^{N_p-1} w_{R,i} \cos(\theta_i + Tw_i/2) & \sum_{i=0}^{N-1} w_{L,i} \cos(\theta_i + Tw_i/2) \\ \sum_{i=0}^{N_p-1} w_{R,i} \sin(\theta_i + Tw_i/2) & \sum_{i=0}^{N-1} w_{L,i} \sin(\theta_i + Tw_i/2) \end{bmatrix} \quad (8)$$

**Path planning under frequent blocking.** In shopping malls, many narrow crisscross passageways make the environment a complicated network, and the passageways are frequently blocked by customers at some time, which changes the connectivity of the environment. In traditional two-layered navigation, once the local planner cannot find a valid command when blocked by crowds, global replanning will be triggered. But this idea is not fully applicable because of the following reasons: (1) The robot may move back and forth between two frequently blocked passageways without progress. (2) Refinding a global path on the whole map is time-consuming. (3) Making a long detour is sometimes more expensive than just waiting for a while.

Making a further analysis, the global planner aims to find a latest feasible path with the dynamic changes in the whole map, the path may oscillate severely, and the length of the path is not considered, but the local controller just follows the global path, thus leading to unreasonable behaviors. In order to eliminate this disharmony between the global and local planner, an intermediate layer is

employed (Figure 9(a)) to limit the path searching within an appropriate range and avoid unnecessary global replanning.

The flow chart of the navigation is shown in the Figure 9(b). Once a goal is receiving, the path from the robot's position to goal is computed. Next, a series of ordered way points are generated from the global path and dispatched to the improved local controller described below. If the local controller fails to find suitable control commands following the blocked path, the intermediate planner will take charge instead of global planner. It tries to search a nearby path by continuously enlarging the range of local planning map until maximum limitation is reached (usually  $30 \times 30 \text{ m}^2$ ). Most often, if there is an alternative path around, robot will turn to follow that path. But if the blocked path is the only one which must be passed, the intermediate planner will send a signal to the global planner to do path searching in the whole map. The length of the new global path will be compared with the remainder of the blocked path, if the former is longer than a threshold, robot will give

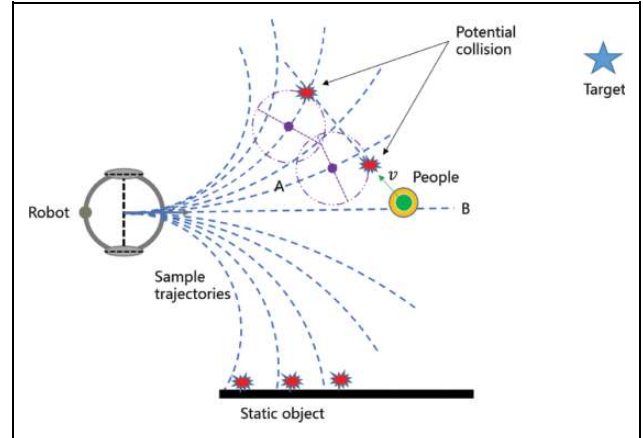


**Figure 9.** (a) Three layers of navigation. (b) Flow chart of navigation.

up the new path and demand the crowds to keep out of the way by making a sound. This strategy makes the navigation more efficient and less disturbing as possible.

**Human-aware local controller.** Local controller is a fundamental component in the navigation of mobile robots, which focuses on collision avoidance and motion control. The problem is mostly solved by reactive algorithms, such as vector field histogram (VFH),<sup>40</sup> dynamic window approach (DWA),<sup>41</sup> and velocity obstacle.<sup>42</sup> The VFH method is simple and easily, but it depends much on parameter tuning. The DWA method could achieve good effects on obstacle avoidance and was used for a period of time in the system. Although the DWA reacts to the dynamics quickly, it lacks the prediction of the environment, particularly the moving people, which often results in overly aggressive or conservative behaviors. The improved DWA method integrates the predictions of human motion into the local controller and further mends the path generated by the global planner.

The human detection approach adopted in navigation is described in the study by Arras et al.,<sup>43</sup> which utilizes AdaBoost to train a classifier based on the features extracted from laser clusters of legs. The tracking and prediction of people is based on the constant velocity model, considering the fact the human motion (speed and orientation) is constant or changes very little in short time, which has already been exploited in several works.<sup>44-46</sup> As introduced in the study by Fox et al.,<sup>41</sup> the DWA method is a local reactive avoidance controller that searches for optimal action (translational speed  $v$  and angular speed  $\omega$ ) in the velocity space, meanwhile, the dynamic constrains of the robot are taken into account to reduce the sample space. Trajectories corresponding to different velocities can be represented by a sequence of circular arcs with different curvatures. In order



**Figure 10.** Illustration of the improved DWA method. DWA: dynamic window approach.

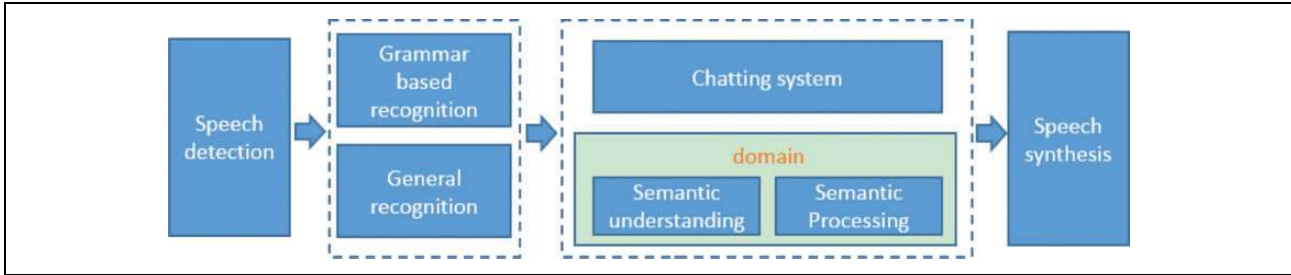
to speed up the velocity sampling for fast reactive response, it assumes that the velocity of robot in all future intervals is constant. The sampled trajectories are evaluated by an objective function  $G(v, \omega)$ , which incorporates the target heading, clearance, and velocity as criteria

$$G(v, \omega) = \sigma \left( \alpha \times \text{heading}(v, \omega) + \beta \times \text{dist}(v, \omega) + \gamma \times \text{velocity}(v, \omega) \right) \quad (9)$$

The heading  $(v, \omega)$  is a measure of the closeness between current orientation and goal direction, the  $\text{dist}(v, \omega)$  is the distance to the closest obstacle along the trajectory, and the  $\text{velocity}(v, \omega)$  represents the forward velocity of the robot with dynamics constraints. (Details can be found in the study by Fox et al.<sup>41</sup>) The improved DWA method is illustrated in Figure 10, the speed of the moving people can be estimated from the tracking module and the predicted trajectory is assumed as a line, thus the potential collisions of sampled trajectories are calculated. The  $\text{dist}(v, \omega)$  now is evaluated by the collision distance of static obstacles as well as the moving obstacles. Under the circumstance shown in Figure 10, the trajectory A is most likely selected in original DWA since it is directly toward the target. While with the improved method, the trajectory B is inclined to be chosen since the left trajectories may bump into the moving people. Although the trajectory A would not lead to immediate collisions, it may lead the robot to an embarrassing situation. Obviously, the trajectory B is more optimal and natural. In order to keep in harmony with the global path planner, the areas of moving people will be cleared and the areas of potential collisions will be marked as occupied in map.

## Multimodal interaction

In this section, the multimodal interaction mechanism with customers is presented, including speech dialog system, human detection, and mobile application.



**Figure 11.** Processing flow of speech dialog manager.

### Speech dialog system

Speech dialog system is the main interface for the customers to interact with the robot. On one hand, it maintains the conversation with customers, and on the other hand, it is expected to understand the customers' intentions. It consists of four parts: (1) Speech detection: it is used to capture the valid part in the voice by detecting the begin and end points of speech. (2) Speech recognition: it is classified into two categories, that is, general recognition and recognition based on rules. It is responsible for turn the input speech to corresponding text. (3) Dialog manager: it analyzes the semantic of the input text and gives out the response text, and meanwhile, a proper task will be generated based on the semantic. (4) Speech synthesis: it converts normal language text into speech and then plays the voice.

The processing procedures are shown in Figure 11. The voice segments are put into the speech recognition module, in most cases, the general recognition is good enough to give an accurate text. In order to satisfy some special requirements, the speech recognition module also supports custom rules, and the rules will be matched prior to general recognition. The recognized text will then be sent to the core dialog manager, which will prejudge whether the text falls in domain. If not, it will be sent to the chatting system, which is a black box. But when the text is in domain, it will be analyzed based on the semantic rules, which are described by predefined rule files. If the text is matched with a rule, the corresponding action (defined in rule files) will be sent to the task manager. The whole dialog system is developed and integrated based on commercial software.

### Human detection

Information gathered from human detection is used by several components in the system. It helps to verify the presence of customers when conversation begins, and robot also could keep facing to customers during interaction if the poses of customers are given. The human detection is combined with face detection and leg detection. The face detection can be divided into three procedures: (1) Preprocess: In order to meet the real-time demand, the static background is removed firstly, and then it focuses on the interest region where the face is detected at last time. (2) Detection: Cascade classifier based on Haar feature is an

effective object detection method proposed by Viola and Jones,<sup>47</sup> and it is also adopted in the face detection. (3) Tracking: The tracking process performs when a new face is detected, the pan tile servo tries to keep the recognized cluster in the middle of the camera view. The human detection by legs from laser data is already described in previous section. Although the face detection and leg detection are preliminary, the integration of them results in good effects.

### Mobile application

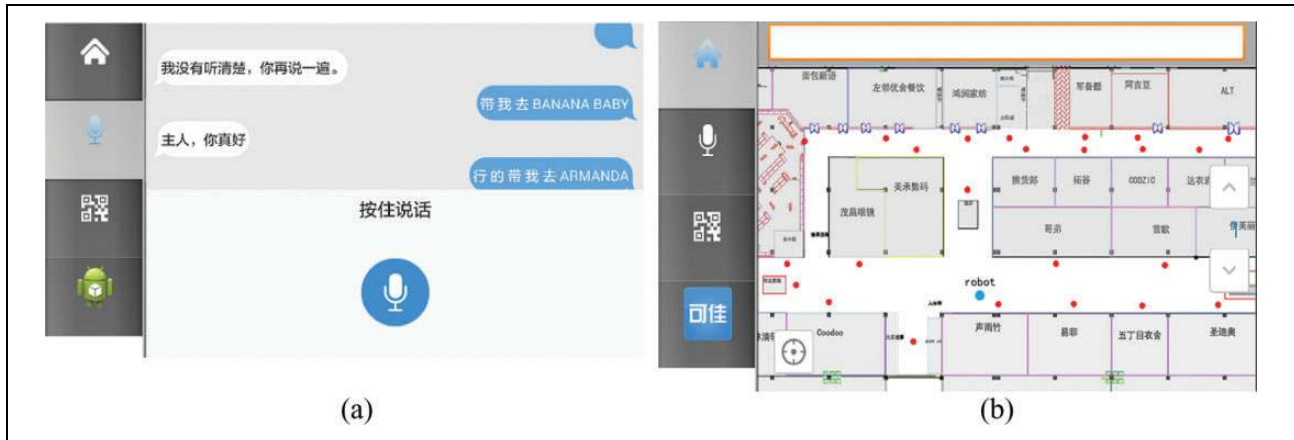
The mobile App in the system is a new interaction style, which is rarely mentioned in previous works. There are mainly two tabbed panels in the mobile App. One is the chatting page (Figure 12(a)), which provides the interface for customers to communicate with the dialog manager on the server. The other one is the display page, as shown in Figure 12(b); the map, the robot position (the blue dot), and available destinations (red dots) are shown on screen, so that customers can select the destinations and browse information by touching screen.

### Experiments and field trials

The field trials last about 40 days from December 8, 2014, including 10 days of preparative debugging work, the demonstration video can be watched on YouTube.<sup>48</sup> The robot system was deployed in GOOCOO shopping mall, which is the largest and most prosperous one in Hefei city, China. It worked on the first floor with size of more than 30,000 m<sup>2</sup> (241 × 140 m<sup>2</sup>), gathering nearly 160 commercial tenants as shown in the floor plan (Figure 13).

### Mapping experiment

The first step to deploy the robot system was to build the environmental map. In order to collect the laser data and odometry data for mapping, the robot was controlled to stroll along all passageways in the shopping mall with a wireless joystick. The route of data collecting is drawn with blue lines and red arrows in Figure 13 (about 1.3 km long), including 147,800 frames of laser data. The final map built by the proposed quadtree mapping is shown in Figure 14; the mapped size is 236.6 × 97.45 m<sup>2</sup> with the resolution of 0.05 m. The method was compared with original Gmapping



**Figure 12.** (a) Chatting page of mobile app. (b) Display page of mobile app.



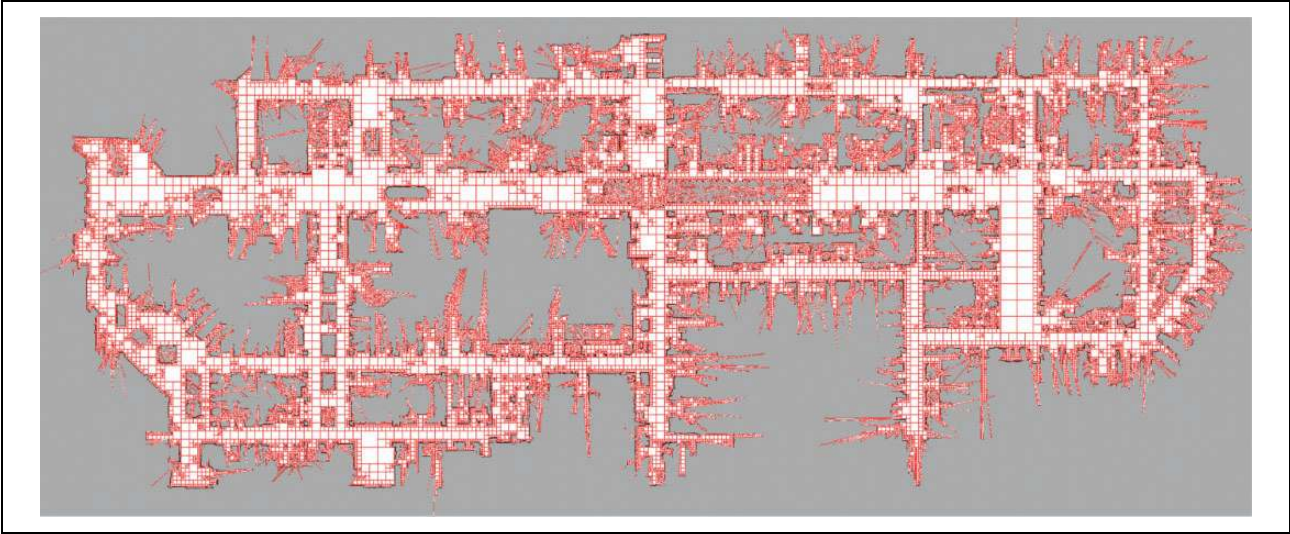
**Figure 13.** The floor plan of the shopping mall is shown in this picture, and the characters are the names of the shops. The blue lines with red arrows are the routes of mapping, the red star is the start point, and the purple star is the end point.

(slam based on grid map) on the same collected data offline. Both the experiments used 30 particles to estimate the map, new observations were inserted to the map only when robot moved more than 1.5 m or rotated more than  $60^\circ$ . The average memory required per particle, the number of grid cells and leaf nodes, disk space of the map, and time spent are listed in Table 1. In general, the memory and disk consumption has a significant reduction, but the time spent increases to 40%, which is mostly spent on the procedures of map matching and updating. In the application, the increased time is acceptable since online mapping is not necessary in *KeJia* system.

### Odometry calibration and localization

The odometry calibration was conducted in laboratory, where a passive optical MoCap was setup. The MoCap system consists of 12 cameras equipped with infrared light-emitting diodes around the camera lens (Figure 15).

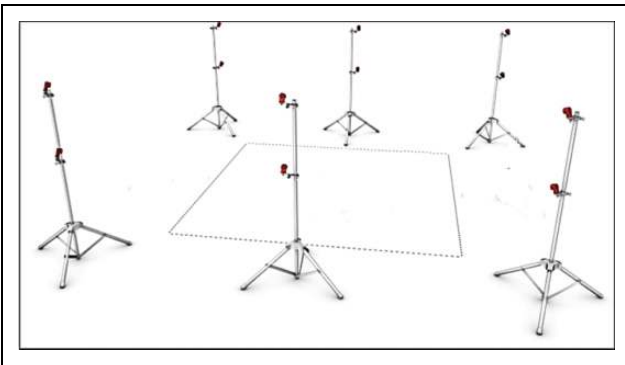
The reflective markers are fixed on the measured objects, and the images of the centers of the markers are matched from various camera views using triangulation to compute their frame-to-frame positions in 3-D space. The robot was installed with a mark set on the base and driven to follow the predetermined routes. In order to eliminate the possible compensation existed in different kinds of trajectories, two sampling modes were designed to find the actual values of the odometry parameters, that is, moving in clockwise and counterclockwise circles (Figure 16). In each sampling mode, 10 trajectories were collected for the follow-up calibration system, and every trajectory was about 6 m long. Meanwhile, the encoder data was logged with the frequency of 40 Hz. Thus, three data sets were prepared: data set A includes 10 clockwise trajectories, data set B includes 10 anticlockwise trajectories, and data set C is assembled by selecting 5 trajectories from A and B, respectively. The results of the calibration performed on three data sets are listed in Table 2; the three sets of parameters calibrated are



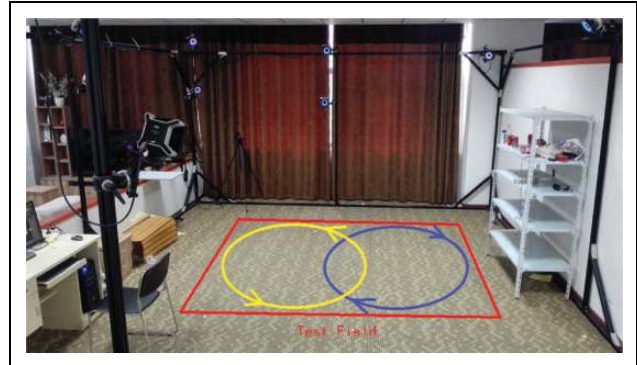
**Figure 14.** The quadtree map is drawn with square grids for visualization. The map has not been processed further; some measures aim at filtering speckles or decreasing burrs can be taken to make the maps more compact and aesthetic.

**Table 1.** Comparison of two mapping methods.

Method	Average memory (MB)	Number of cells	Disk space	Time cost (min)
Grid map	35.12	13,452,288	45.3 MB	32
Quadtree	10.08	6,252,223	287.1 KB	45



**Figure 15.** The diagram of MoCap system. MoCap: motion capture system.



**Figure 16.** The deployed MoCap system for odometry calibration. MoCap: motion capture system.

almost near to each other and all different with nominal value (NV). To verify these parameters, robot was driven to follow a closed trajectory with length of 100 m, and the start point was regarded as reference point. The odometry was calculated with the three sets of parameters and the NV, and the errors are also presented in Table 2, which are generally restrained under 2.5% after calibration. It is hard to say which data set is better, but they are all superior to the uncalibrated NV.

In order to test the performance of the improved MCL method in dynamic environments, simulated experiments were conducted with Gazebo.<sup>49</sup> The scenario simulated was a crowded passageway (the pedestrians were static, see

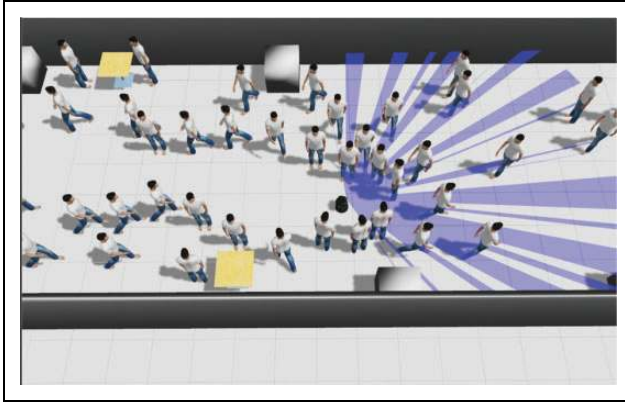
Figure 17), a TurtleBot mounted with a laser was driven to pass through the crowds (Figure 18). The same route would be followed by robot many times in the same scenario, but with different numbers of pedestrians within it, the pedestrians were added in new positions and kept the existing scenario unchanged.

Figure 19(a) shows the corruption rate  $\sigma_p$  of three comparative experiments, it is obvious that the more crowded in the environment, the smaller of the  $\sigma_p$  will be, proving that the proposed corruption rate could fittingly depict the dynamics of the environment. Figure 19(b) shows that the success rate of localization continually reduces with the increasing number of pedestrians.

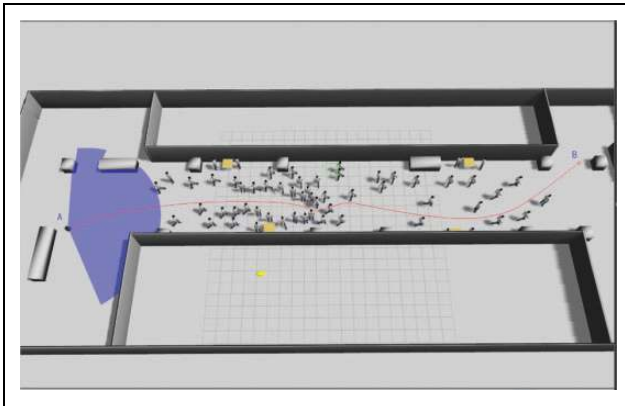
**Table 2.** Calibration on three datasets and nominal value.

Dataset	$r_L$ (cm)	$r_R$ (cm)	$b$ (cm)	Distance error of RP (m)	Angle error of RP ( $^\circ$ )
NV	10.00	10.00	45.00	6.53	71.3
A	9.74	9.68	46.57	2.49	25.6
B	9.78	9.70	46.34	2.03	30.6
C	9.77	9.69	46.42	2.32	27.3

RP: reference point. NV: nominal value.



**Figure 17.** The laser data are corrupted severely by curious crowds.



**Figure 18.** A simulated scenario containing 60 pedestrians.

Meanwhile, it also proves that the proposed method is more robust than original MCL method, even in overcrowded condition, it still achieves a success rate more than 50%.

### Navigation

In the practical navigation, the robot is only expected to move in the passageways, hence some areas in the map are blocked with black padding for constrained path planning (Figure 20), which are usually the interior of shops that are forbidden for the robot. For localization, the original map is used to make the laser data matched as good as possible. Figure 20(a) shows a global path based on static map from start point (brown) to end point (red), and Figure 20(b)

shows the intermediate planner within the windows of local map ( $3.5 \times 3.5 \text{ m}^2$  square box), the path planned in this level is not exactly the same with the global path because the new surroundings have been taken into account, and it is the key that the system can handle with the dynamic changes in environments.

The proposed local controller had also been tested in Gazebo under various simulated situations, and the proposed method was proved to be safer and more efficient in general. Figure 21 shows a typical case where the robot pursues its goal in the presence of two moving people. The pictures in top row show the robot's path obtained by original DWA method at four moments, while the pictures in the bottom row are captured from the simulation of the proposed method. Obviously, the path in the top one is tortuous and unsmooth, exposing the shortsightedness of most reactive controllers. Unlike with the original DWA method, the path generated by the proposed method is more similar to human behavior, straight to the goal without any unnecessary avoidance.

### Field tests

During the operational period (not including the debugging period), the robot served about 530 customers who had kept a dialog of more than half a minute; 289 guidance tasks were understood correctly, and totally about 150 complete tours were performed successfully. The results of the field tests will be presented from the aspects of safety, reliability, human-robot interaction, and customers' feedback, respectively.

**Safety.** The robot has never caused a dangerous situation or an accident, which threaten the safety of customers and facilities. No collisions occurred when robot was in high speed, and some slight scratches happened but were typically provoked by customers themselves in the following situations: (1) Some customers tried to stop the robot by standing in front of it; usually, they would dare not to do that when the robot was moving quickly. Therefore, this case frequently occurred when robot was rotating slowly to find openings in traps, at this time, the customers cannot be stably perceived by the front laser. (2) Some customers tried to draw the attention of the robot by waving their arms in front of the robot's eyes; since no vision information was used in local avoidance, this behavior may cause collision, but in fact it rarely led to substantial danger due to the

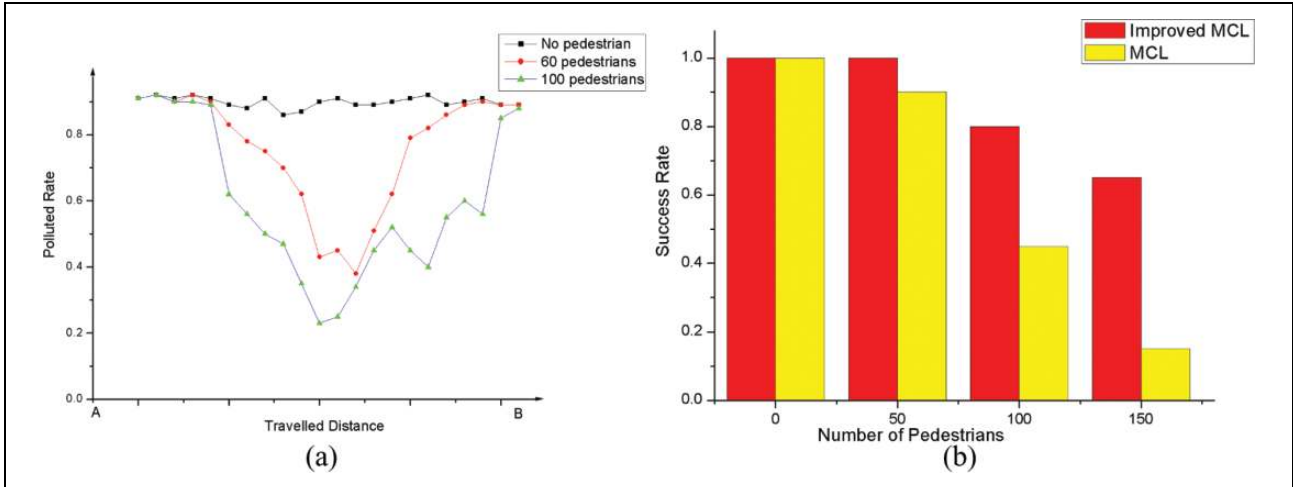


Figure 19. (a) Corruption rate of laser data along the path. (b) Success rate of localization varies with the number of pedestrians.

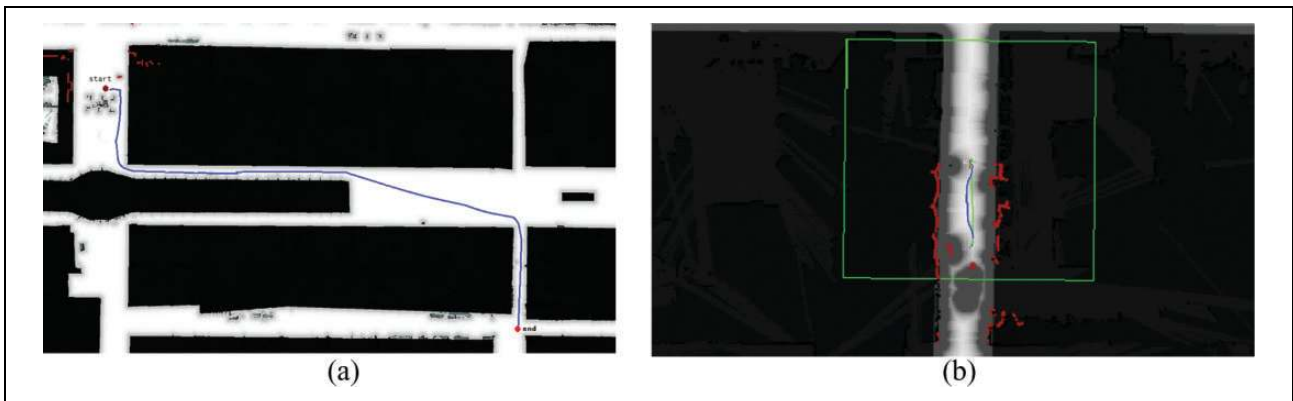


Figure 20. (a) Path planning in global map. (b) Intermediate planner in local map.

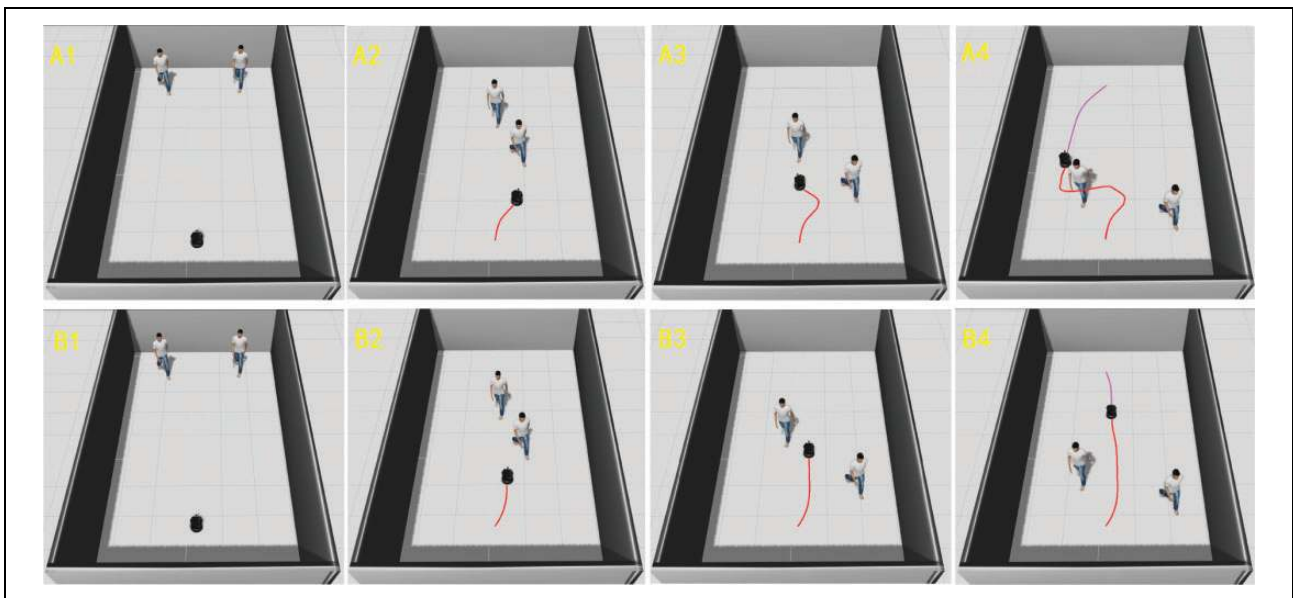


Figure 21. Comparison of two local planners in a typical case involving moving people. The pictures labeled by A1-A4 and B1-B4 are the sequential snapshots of the two compared methods at four same moments.



**Table 3.** The reasons of navigation failures.

Reason	Localization failure	Give up by customers	Blocked by crowds	Unknown reasons
Percentage (%)	10.8	44.6	18	26.6

dexterousness of human arm. (3) The height of the installed laser is 28 cm, so the robot cannot perceive small objects (such as bottles and cardboard boxes) and often pushes them away on the ground, but the design of the robot chassis (an enclosed box near the ground) can avoid wheel crushing.

**Reliability.** The robot worked less than 2 h in the first week because of the high failure rate (especially the hardware), so a lot of time was spent to troubleshoot it. But at the later stage, the situation was much improved, robot could work about 4 h per day with a full charge, and the success rate was continually rising with days and lastly stabilized at about 75%. The localization module with laser was quite stable, and the proposed method was effective in dynamic environments. However, the failures occurred because of the following reasons: (1) customers pushed or rotated the robot when it stopped, (2) the loss of laser data caused by the drop of hardware connection, and (3) odometry jump caused by scratches with fixed objects. Usually, the failures in localization lead to the failures in navigation. In addition, navigation is extremely difficult in dense crowds, and robots were often blocked. What impressed us most was that on New Year's day, a number of customers proliferated and the robot never succeeded to pass through the passageway near to the entrance of the shopping mall. In general, the success rate of the navigation task in the test filed depends on the density of customers in environments, and the robot would give up tasks rather than taking excessive risk. The reasons of the recorded navigation failures (totally 89 times) are listed in Table 3, and it should be noted that 44.6% tasks are given up halfway by customers in the guidance task, which may be explained from two aspects: (1) Most of the customers just want to have a try without true intentions to the goals, and they are impatient with the long journeys and attracted by other things on the way. (2) The behaviors of the robot in the process of guidance are too monotonous to keep customers interested for long time. Besides, 26.6% failures are categorized for unknown reasons, which may imply that some latent defects exist in the system.

**Human-robot interaction.** Customers in the shopping mall are free to choose one of the two interactive methods: speech or mobile App. In face-to-face mode, customers' sentences are recognized and then passed to dialog manager module, once the intentions are explicit the robot will begin a tour guidance. Customers can also talk with *KeJia* by pressing the microphone icon and the text displays on the screen, and the shops' positions are drawn on the map with red dots. In general, the mobile App provides a

**Table 4.** Customers' feeling to the robot.

Common	Fear	Disappointment	Surprise
18%	21%	15%	46%

**Table 5.** Customers' attitude to the robot.

Options	A	B	C	D
Percentage (%)	46	30	16	6

straightforward and effective graphical interface for users. The customers seem to be more interested in chatting with the robot by mobile client, and they intentionally speak some sentences that can hardly be handled by *KeJia* and expect it to respond with funny jokes.

**Feedback from customers.** By exploiting the customers' comments in the mobile App, a general analysis of the customers' feedback was carried out. Customers' feeling to the robot was judged based on their comments and divided into four classes: common, disappointment, surprise, and fear. As shown in Table 4, most of the customers feel fresh and surprise with the robot. While about 15% of customers are disappointed with the robot, the reason may be that the robot never meets their expectations of an intelligent robot like in science fiction movies. It is noteworthy that some customers feel horror with the robot, because the appearance of *KeJia* is lifelike but its expression is dull. The customers' attitudes to the function of the robot are shown in Table 5. The options A, B, C, and D in the table represent that the robot now is useful but need improvements, the robot now is useful and good enough, the robot now is useless but promising, and the robot is useless and unpractical. In general, the customer's feedback is positive, but there is still room for improvement.

## Conclusions

From the practical operation results of *KeJia* robot, firstly the reliability and effectiveness of the proposed robotic techniques are proven, including layered software architecture, mapping in large environments, localization, and navigation in dynamic environments. Secondly, the way to interaction with the robot by mobile phone is explored, and it becomes the most often used and accepted way by the customers in actual running. Lastly, from the customers' feedback, it turns out that the general public have intense interests in service robots, which is encouraging for

us. Overall, we hope this work can provide some experience and enlightenments for similar robotic applications.

However, there are still gaps between existing techniques and expectations. In order to build a more intelligent service robot, several aspects should be improved in our future work. For robotic mapping, the 2-D maps are insufficient to contain enough environmental information since 3-D mapping techniques are more promising. Meanwhile, how to maintain a long-term consistent map should also be considered. The success rate of localization depends much on the dynamics in the environment, and therefore some other sensors (such as gyro and camera) can be integrated to improve the robustness of localization. The research of robot navigation in crowds is not limited to the motion control, and the perception of environments and the understanding of people's intentions are both the prerequisites for social human-robot navigation, all of these capabilities should be enhanced further.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Hi-Tech Project of China under grant 2008AA01Z150, the Natural Science Foundations of China under grants 60745002 and 61175057, as well as USTC 985 project. Besides, Feng Wu is supported in part by the Natural Science Foundations of China under grants 61603368, the Youth Innovation Promotion Association, CAS (no. CX0110000012), and Anhui Provincial Natural Science Foundation under grants 1608085QF134.

### References

1. Overview of the PR2 robot, <http://www.willowgarage.com/pages/pr2/overview> (2010, accessed 6 December 2016).
2. Atlas – the agile anthropomorphic robot, [http://www.bostondynamics.com/robot\\_Atlas.html](http://www.bostondynamics.com/robot_Atlas.html) (2014, accessed 6 December 2016).
3. Guizzo E. Meka robotics announces mobile manipulator with kinect and ROS, <http://spectrum.ieee.org/automaton/robotics/humanoids> (2011, accessed 6 December 2016).
4. Leite I, Martinho C and Paiva A. Social robots for long-term interaction: a survey. *Int J Soc Robot* 2013; 5(2): 291–308. DOI:10.1007/s12369-013-0178-y.
5. Chen X, Jin G, Ji J, et al. KeJia project: towards integrated intelligence for service robots. *RoboCup@Home League Team Descript Istanbul*, Turkey, July 2011, Berlin: Springer, pp. 3–9.
6. Burgard W, Cremers AB, Fox D, et al. Experiences with an interactive museum tour-guide robot. *Artif Intell* 1999; 114(1): 3–55. DOI:10.1016/S0004-3702(99)00070-3.
7. Thrun S, Bennewitz M, Burgard W, et al. Minerva: a second-generation museum tour-guide robot. In: *Proceedings of 1999 IEEE international conference on robotics and automation*, vol. 3, Detroit, Michigan, 10–15 May 1999, pp. 1999–2005, New York: IEEE. DOI:10.1109/ROBOT.1999.770401.
8. Bischoff R and Graefe V. Hermes—an intelligent humanoid robot designed and tested for dependability. In: *Experimental robotics VIII ISER 2002*, Sant'Angelod'Ischia, Italy, 8–11 July 2002, pp. 64–74. Berlin: Springer. DOI:10.1007/3-540-36268-1\_4.
9. Faber F, Bennewitz M, Eppner C, et al. The humanoid museum tour guide robotinho. In: *RO-MAN 2009—the 18th IEEE international symposium on robot and human interactive communication*, Japan, 27 September–2 October 2009, pp. 891–896. New York: IEEE. DOI:10.1109/ROMAN.2009.5326326.
10. Siegwart R, Kai OA, Bouabdallah S, et al. Robox at expo.02: a large-scale installation of personal robots. *Robot Auton Syst* 2003; 42(3–4): 203–222. DOI:10.1016/S0921-8890(02)00376-7.
11. Shieh MY, Hsieh J and Cheng C. Design of an intelligent hospital service robot and its applications. In: *2004 IEEE international conference on systems, man and cybernetics*, vol. 5, Netherlands, 10–13 October 2004. pp. 4377–4382, New York: IEEE. DOI:10.1109/ICSMC.2004.1401220.
12. Tomizawa T, Ohya A and Yuta S. Remote shopping robot system - development of a hand mechanism for grasping fresh foods in a supermarket. In: *2006 IEEE/RSJ international conference on intelligent robots and systems*, Beijing, China, 9–15 October 2006, pp. 4953–4958, New York: IEEE. DOI:10.1109/IROS.2006.282457.
13. Kulyukin V, Gharpure C and Nicholson J. RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired. In: *2005 IEEE/RSJ international conference on intelligent robots and systems*, Edmonton, Alberta, Canada, 2–6 August 2005, pp. 2845–2850, New York: IEEE. DOI:10.1109/IROS.2005.1545107.
14. Shieh MY, Chan YH, Lin ZX, et al. Design and implementation of a vision-based shopping assistant robot. In: *2006 IEEE international conference on systems, man and cybernetics*, vol. 6, Taipei, Taiwan, 8–11 October 2006, pp. 4493–4498, New York: IEEE. DOI:10.1109/ICSMC.2006.384852.
15. Kanda T, Shiomi M, Miyashita Z, et al. An affective guide robot in a shopping mall. In: *Proceedings of the 4th ACM/IEEE international conference on human robot interaction* (eds Scheutz M, Michaud F, Hinds PJ et al.), La Jolla, California, USA, 9–13 March 2009, pp. 173–180, New York: ACM. DOI:10.1145/1514095.1514127.
16. Iwamura Y, Shiomi M, Kanda T, et al. Do elderly people prefer a conversational humanoid as a shopping assistant partner in supermarkets? In: Billard A, Kahn PH Jr, Adams

- JA, et al (eds) *Proceedings of the 6th International Conference on Human Robot Interaction, HRI 2011*, Lausanne, Switzerland, 6–9 March 2011, pp. 449–456, New York: ACM. DOI:10.1145/1957656.1957816.
17. Gross HM, Boehme H, Schroeter C, et al. TOOMAS: interactive shopping guide robots in everyday use – final implementation and experiences from long-term field trials. In: *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 11–15 October 2009, pp. 2005–2012, New York: IEEE. DOI:10.1109/IROS.2009.5354497.
  18. Grisetti G, Stachniss C and Burgard W. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In: *Proceedings of the 2005 IEEE international conference on robotics and automation (ICRA 2005)*, Barcelona, Spain, 18–22 April 2005, pp. 2432–2437, New York: IEEE. DOI:10.1109/ROBOT.2005.1570477
  19. Thrun S, Burgard W and Fox D. *Probabilistic robotics*. Cambridge: MIT Press, 2005. DOI:10.1145/504729.504754.
  20. Quigley M, Conley K, Gerkey B, et al. ROS: an open-source robot operating system. In: *ICRA 2009 workshop on open source software*, Kobe, Japan, 12–17 May 2009, vol. 3, p. 5, New York: IEEE.
  21. Hahnel D, Burgard W, Fox D, et al. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: *Proceedings of 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003)*, vol. 1, Las Vegas, Nevada, 27 October–1 November 2003, pp. 206–211, New York: IEEE. DOI:10.1109/IROS.2003.1250629.
  22. Morton GM. *A computer oriented geodetic data base and a new technique in file sequencing*. New York: International Business Machines Company, 1966.
  23. Murphy KP. Bayesian map learning in dynamic environments. In: *Proceedings of the 12th international conference on neural information processing systems (NIPS'99)* (eds Solla SA, Leen TK and Müller K), Denver, CO, 29 November–4 December 1999, pp. 1015–1021, Cambridge: MIT Press.
  24. Giorgio G, Cyrill S and Wolfram B. Gmapping project, <http://www.openslam.org/gmapping.html> (2009, accessed 7 December 2014).
  25. Gutmann JS, Burgard W, Fox D, et al. An experimental comparison of localization methods. In: *Proceedings of 1998 IEEE/RSJ international conference on intelligent robots and systems*, vol. 2, Victoria, BC, Canada, 13–17 October 1998, pp. 736–743, New York: IEEE. DOI:10.1109/IROS.1998.727280.
  26. Betke M and Gurvits L. Mobile robot localization using landmarks. *IEEE Trans Robot Autom* 1997; 13(2): 251–263. DOI: 10.1109/70.563647.
  27. Borenstein J and Feng L. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans Robot Autom* 1996; 12(6): 869–880. DOI:10.1109/70.544770.
  28. Martinelli A, Tomatis N, Tapus A, et al. Simultaneous localization and odometry calibration for mobile robot. In: *2003 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, vol. 2, Las Vegas, Nevada, USA, 27 October–1 November 2003, pp. 1499–1504, New York: IEEE. DOI:10.1109/IROS.2003.1248856.
  29. Censi A, Franchi A, Marchionni L, et al. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Trans Robot* 2013; 29(2): 475–492. DOI:10.1109/TRO.2012.2226380.
  30. Antonelli G, Chiaverini S and Fusco G. A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation. *IEEE Trans Robot* 2005; 21(5): 994–1004. DOI:10.1109/TRO.2005.851382.
  31. Fox D. Adapting the sample size in particle filters through KLD-sampling. *Int J Robot Res* 2003; 22(12): 985–1003. DOI:10.1177/0278364903022012001.
  32. Yamauchi B and Beer R. Spatial learning for navigation in dynamic environments. *IEEE Trans Syst Man Cybern B Cybern* 1996; 26(3): 496–505. DOI:10.1109/3477.499799.
  33. Roy N, Burgard W, Fox D, et al. Coastal navigation—mobile robot navigation with uncertainty in dynamic environments. In: *Proceedings of 1999 IEEE international conference on robotics and automation*, vol. 1, Detroit, Michigan, 10–15 May 1999, pp. 35–40, New York: IEEE. DOI:10.1109/ROBOT.1999.769927.
  34. Arras KO, Philippsen R, Tomatis N, et al. A navigation framework for multiple mobile robots and its application at the expo.02 exhibition. In: *Proceedings of IEEE international conference on robotics and automation (ICRA'03)*, vol. 2, Taipei, Taiwan, 14–19 September 2003, pp. 1992–1999, New York: IEEE. DOI:10.1109/ROBOT.2003.1241886.
  35. Lam CP, Chou CT, Chiang KH, et al. Human-centered robot navigation towards a harmoniously human robot coexisting environment. *IEEE Trans Robot* 2011; 27(1): 99–112. DOI: 10.1109/TRO.2010.2076851.
  36. Stachniss C, Kai OA and Burgard W. Socially inspired motion planning for mobile robots in populated environments. In: *2008 international conference on cognitive systems (CogSys)*, Karlsruhe, Germany, 2–4 April 2008, pp. 85–90.
  37. Pandey AK and Alami R. A framework for adapting social conventions in a mobile robot motion in human-centered environment. In: *International conference on Advanced robotics (ICAR 2009)*, Munich, Germany, 22–26 June 2009, pp. 1–8, New York: IEEE.
  38. Trautman P and Krause A. Unfreezing the robot: navigation in dense, interacting crowds. In: *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Taipei, Taiwan, 18–22 October 2010, pp. 797–803, New York: IEEE. DOI:10.1109/IROS.2010.5654369.
  39. Henry P, Vollmer C, Ferris B, et al. Learning to navigate through crowded environments. In: *2010 IEEE international conference on robotics and automation (ICRA)*, Anchorage,

- Alaska, USA, 3–7 May 2010, pp. 981–986, New York: IEEE. DOI:10.1109/ROBOT.2010.5509772
40. Borenstein J and Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans Robot Autom* 1991; 7(3): 278–288. DOI:10.1109/70.88137.
41. Fox D, Burgard W and Thrun S. The dynamic window approach to collision avoidance. *IEEE Robot Automat Mag* 1997; 4(1): 23–33. DOI:10.1109/100.580977.
42. Wilkie D, Van Den Berg J and Manocha D. Generalized velocity obstacles. In: *2009 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, St. Louis, MO, USA, 11–15 October 2009, pp. 5573–5578, New York: IEEE. DOI:10.1109/IROS.2009.5354175.
43. Arras KO, Mozos ÓM and Burgard W. Using boosted features for the detection of people in 2D range data. In: *2007 IEEE international conference on robotics and automation*, Roma, Italy, 10–14 April 2007, pp. 3402–3407, New York: IEEE. DOI:10.1109/ROBOT.2007.363998.
44. Pellegrini S, Ess A, Schindler K, et al. You’ll never walk alone: modeling social behavior for multi-target tracking. In: *2009 IEEE 12th international conference on Computer vision*, Kyoto, Japan, 27 September– 4 October 2009, pp. 261–268, New York: IEEE. DOI:10.1109/ICCV.2009.5459260.
45. Granata C and Bidaud P. A framework for the design of person following behaviors for social mobile robots. In: *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Vilamoura, Algarve, Portugal, 7–12 October 2012, pp. 4652–4659, New York: IEEE. DOI:10.1109/IROS.2012.6385976.
46. Chung SY and Huang HP. Predictive navigation by understanding human motion patterns. *Int J Adv Robot Syst* 2011; 8(1): 52–64.
47. Viola P and Jones M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR 2001)*, vol. 1, Kauai, HI, USA, 8–14 December 2001, pp. 511–518. DOI:10.1109/CVPR.2001.990517.
48. Chen Y, Wu F, Shuai W, et al. KeJia – guide robot in shopping mall, <https://youtu.be/qdHI5ESGKFw> (2016, accessed 6 December 2016).
49. Koenig N and Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *Proceedings of 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS 2004)*, vol. 3, Sendai, Japan, 28 September–2 October 2004, pp. 2149–2154, New York: IEEE.