Pebbles: User-Configurable Device Network for Robot Navigation

Kentaro Ishii[1,2], Haipeng Mi[1,2], Lei Ma[1,2], Natsuda Laokulrat[1,2],
Masahiko Inami[3,2], and Takeo Igarashi[1,2]

[1] The University of Tokyo, Tokyo, Japan
[2] JST, ERATO, IGARASHI Design Interface Project, Tokyo, Japan
[3] Keio University, Yokohama, Japan
kenta@sakamura-lab.org, mi@iii.u-tokyo.ac.jp,
malei@satolab.itc.u-tokyo.ac.jp, natsuda@logos.t.u-tokyo.ac.jp,
inami@inami.info, takeo@acm.org

Abstract. This study proposes devices suitable for use by non-experts to design robot navigation routes. The user places landmarks, called pebbles, on the floor to tell navigation routes to a robot. Using infrared communication, the pebbles automatically generate navigation routes. The system is designed such that non-expert users can understand the system status to configure the user's target environment without expert assistance. During deployment, the system provides LED and voice feedback. The user can confirm that the devices are appropriately placed for the construction of a desired navigation network. In addition, because there is a device at each destination, our method can name locations by associating a device ID with a particular name. A user study showed that non-expert users were able to understand device usage and construct robot navigation routes.

Keywords: Robot Navigation, Tangible User Interface, Navigation Landmark, Non-Expert User.

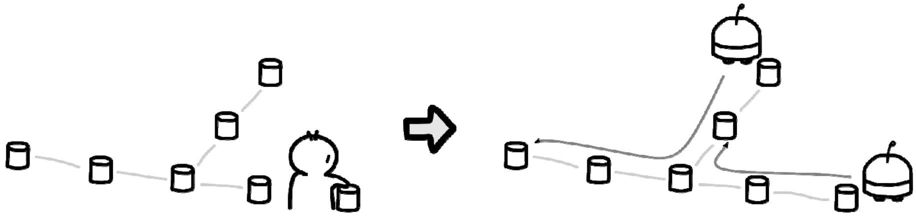


Fig. 1. Pebbles Usage. The user places pebbles at desired locations. Pebbles automatically construct a network topology. After deployment, the robot can travel to any pebble in the network.

1 Introduction

Robots can be used for various applications in both home and office environments. For example, a robot can travel to predetermined locations at a specific time to collect environmental data. Robots can also deliver a physical object, such as a cup of coffee

P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS 8118, pp. 420–436, 2013.
© IFIP International Federation for Information Processing 2013

to a colleague, upon a user's request. If a robot moves an electric fan closer to a person, the fan can work more energy efficiently. However, it is difficult for a robot to know how to reach a specific target location in an unknown environment. Meanwhile, regarding user interaction, it is preferable to specify target locations using natural expressions such as names of the locations. In this study, we propose a method that allows a non-expert end-user to provide navigation information to a robot and to naturally specify locations.

While there are many ways to characterize existing methods to configure an environment for robot navigation, we divide the existing methods into two approaches: automatic mapping by robots and environmental configuration by humans. Automatic mapping by robots, represented by simultaneous localization and mapping (SLAM) [1], is an effort saving method from the user's point of view. The robot acquires map information by automatically exploring the environment, and the robot can plan navigation for a specific location after constructing a map. However, the internal representation of the constructed map is typically a set of coordinates, which may be too complicated for non-expert users to associate with a specific named location, such as "kitchen." We call this name association "labeling." In addition, in such a system, the robot may enter into areas where the user does not want the robot to enter.

On the other hand, there are several implementations that fall into the category of environmental configuration by humans. Physical guides, such as rails for trolley robots and lines for line following robots are the simplest ways to realize such environmental navigation support. A robot can reach target locations by tracing these environmental guides. Most other techniques employ absolute positioning because odometer-based positioning has the fundamental limitation of accumulating positioning errors. Some directly measure a robot's position using environmental sensors [2-3], while some other employ robot-readable devices or tags that work as landmarks that allow the robot to determine its position in the environment [4-5]. For a robot to navigate in these environments, the map of the entire environment is required and is typically provided by the person who manually calibrates and installs sensors, devices, or tags. Combining this manual registration of map information with manual labeling of locations enables a user to naturally ask the robot to reach a specific location. However, existing methods require experts to deploy devices, and therefore the home or office user needs expert assistance to configure locations and labels.

We propose a method that allows non-expert users to configure the environment for robot navigation employing infrared communication devices called pebbles. The robot receives topology information from these devices and moves to locations by tracing infrared signals. We extend previously demonstrated implementation [6], and discuss usage scenario and deployment assistance. We also conduct a user study to see if non-expert users can understand device usage. Fig. 1 illustrates the basic concept of the proposed method. The user only has to place pebbles at desired locations, and the pebbles will automatically construct a network topology. After deployment, the robot can move to any pebble location in this network. Since infrared signals travel in a straight line, if one pebble has unobstructed line-of-sight to another pebble, the robot can be expected to travel between these two pebbles. With our method, destinations are determined by pebbles, and thus it is easy to label destinations by associating a pebble device ID with a name. In our current implementation, the user can easily do this by putting physical labels on buttons on a remote control, for example, “kitchen” or “entrance” (Fig. 5). The user can select a labeled button for the target location. In addition, our method explicitly prevents the robot entering into the area where the user does not want the robot to do, because the robot moves by tracing the constructed network trails. Although our method requires the user to place each pebble so that it is within line-of-sight of neighboring pebbles, each pebble can provide LED feedback, and the robot can provide voice feedback to confirm connection status. None of these configuration processes require expert knowledge.

Our contribution is presenting a user-configurable method for robot navigation, rather than the system’s algorithms, communication technology, or network construction. LED and voice feedback helps the user to construct an appropriate network, and we describe the interaction design between the user and the devices.

2 Related Work

Several positioning techniques have been proposed for indoor environment, where GPS technology provides only limited performance. Radio-based fingerprinting [7] is one of the trends in indoor positioning; in most cases, the system does not require major changes to infrastructure because of the recent widespread presence of wireless LAN [8]. Some other approaches calculate positions by installing sensors in the environment [9-13], including a commercial system [13]. Another popular approach is to utilize cameras fixed in the environment [14]. Other approaches use passive tags to calculate position. Saito et al. implemented an indoor marker-based localization system using coded seamless patterns [15]. Similar indoor localization systems have been subsequently proposed using invisible markers installed on the ceiling [16]. These absolute positioning techniques can be used to determine robot position, but navigating in the environment also requires environmental knowledge, in particular, map information. Typically, registering such map information requires expert knowledge and is not easy to reorganize after registration.

On the other hand, relative positioning by devices emitting beacon signals is another method of robot navigation. Similar to physical guides such as rails and lines, the robot can trace signals to move among the devices. These devices also work to understand environmental structure by detecting if the beacon signals reach others as expected. There have been some work where a robotic system uses such beacon devices to determine the environment [17-20], and we share the basic idea of this previous work. However, the previous work does not discuss if an average user can deploy such devices, which is our focus here. In contrast, a robot places the devices [17], or robots themselves are the devices [18-20].

The automatic mapping techniques represented by SLAM are widely used for robots and autonomous vehicles [1]. Various sensors are used for observing an environment, such as sonar [1,21], laser [22], and vision [23-25]. The robot constructs a map of the environment by exploration, and the robot can use the constructed map to navigate toward a specific location in the map. However, a typical internal representation of the map is a set of coordinates, and it is not trivial for a user to specify a destination or label a location within it. In general, this method requires another representation or visualization of the constructed map so that the user may specify a point in the map. This paper deals with the issue of providing a user-understandable representation, as described in the next paragraph.

In the proposed method, tangible user interface [26] is a key concept for providing a user-understandable representation of the environmental map. Placing a tangible device at each destination bridges the specific location in the user's physical world and the specific node in the system's internal topology map. In our prototype system, the internal representation of the environmental map is a network topology, but the user does not need to know about it. The user only has to specify and label a device at desired locations in the physical world, and the system determines the steps to reach the devices from the network topology.

3 Pebbles

3.1 Device Hardware

A pebble is a self-contained device that is capable of communicating with other pebbles. Pebbles construct a network topology to generate navigation trails for the robot. Each pebble consists of a microcontroller unit, infrared communication unit, and battery set. All components are packed in a plastic cylinder case 10 cm in diameter and 10 cm in height (Fig. 2). Each pebble has a unique ID that distinguishes it within the network. A button labeled with this ID sits on the top for user input.

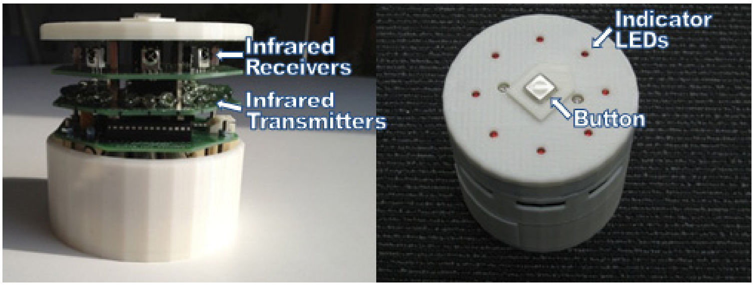


Fig. 2. Pebble Hardware. A pebble has 24 infrared transmitters, 8 receivers, 8 indicator LEDs, and a button. The infrared transmitters and receivers are arranged to cover all directions. The indicator LEDs are used to provide primitive feedback to the user. The button is used to receive user input.

The infrared communication unit has eight infrared transceiver modules; each one contains a red LED indicator, infrared receiver, and three infrared signal transmitters. The modules are placed 45 degrees apart so that all eight modules achieve 360-degree coverage. Infrared signals are modulated using a 38 kHz carrier, a frequency typically used for remote-controlled appliances, to prevent errors caused by noise. The indicator LEDs on the top provide primitive feedback.

3.2 Deploying Pebbles

The robot plans its navigation based on topological information from the pebbles and moves toward a goal according to pebble signals. Therefore, each pebble must communicate with at least one other pebble, and no pebble should be completely isolated. To arrange for each pebble to communicate with its neighbors is the user's responsibility. The system provides LED and voice feedback to help the user to ensure pebble connectivity.

The user can see LEDs on the top of each pebble, each of which indicates its connection status to another pebble in a specific direction (Fig. 3). When transmitting and receiving infrared signals, the indicator LEDs of corresponding channels synchronously blink to indicate adjacent relationships. The user can see if a pebble is visible to its neighbor pebbles. In this way, the user knows if two pebbles are placed too far apart or the path directly between them is blocked by some obstacles. The user can then correct the problem.

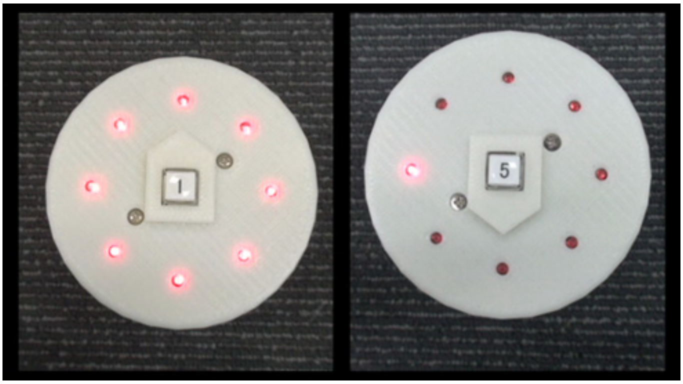


Fig. 3. LED Feedback. When transmitting and receiving infrared signals, the indicator LEDs of corresponding channels synchronously blink to indicate adjacent relationships to the user. The LEDs of the transmitting pebble (left) indicate the transmission of the signal to all directions. The LEDs of the receiving pebble (right) indicate the reception of the signal in the direction of the transmitting pebble.

In addition to the LED feedback, the system also provides voice feedback. The robot enumerates the IDs of pebbles in the network, for example, "I can locate number one, number three, and number five." The user then knows which pebbles are not visible to the robot and can identify which connection is not working as expected. If the robot does not receive any signals from pebbles, it says, "I cannot see any pebble," to indicate that the user should rearrange it.

Utilizing LED and voice feedback, the user places pebbles at desired goals and their transit points. If the user places pebbles as in Fig. 4, for example, the robot can go to all rooms except the top middle room. Since only routes connecting two arbitrary neighbor pebbles are considered possible trails, the user can explicitly keep the robot from entering the top middle room.

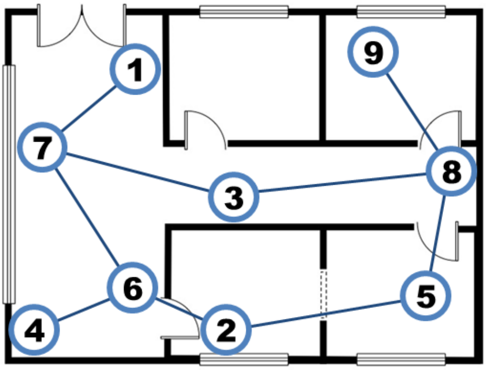


Fig. 4. Sample Deployment. The robot can go to all rooms except the top middle room. In this case, the user explicitly keeps the robot from entering the top middle room.

When the user changes the layout of the rooms or introduces a new object that obstructs a route of the robot, the user has to rearrange the pebbles near the change. Pebbles periodically transmit signals and automatically detect topology changes. Therefore, the user can reconfigure the network by adjusting pebbles, confirming that a pebble communicates with its neighbors as it did initially. This is useful, for example, when the user places an obstacle, such as a piece of luggage, in the room temporarily and wants to specify a path that avoids it.

Locations are labeled by attaching physical labels on the buttons of the remote control (Fig. 5). After labeling, the user can select the target location of the robot by the name of the label. We consider three kinds of labeling: static, semi-static, and dynamic targets. Static targets are the places whose name will not be changed after initial

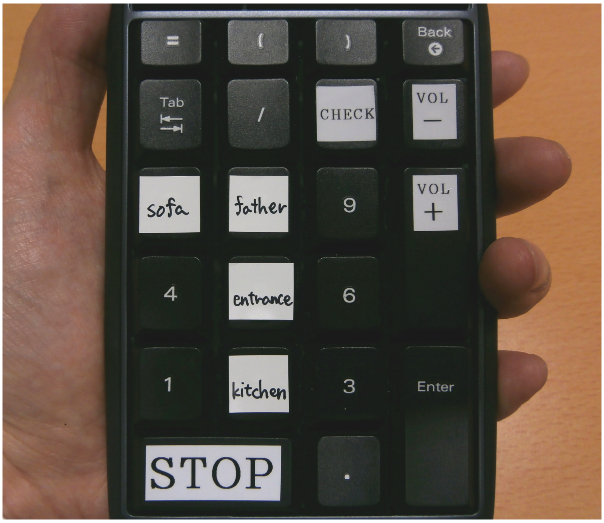


Fig. 5. Labeling by Physical Labels. The user attaches physical labels on a remote control to name locations. The labels of the targets can be static like “kitchen” and “entrance,” semi-static like “sofa,” or dynamic like “father”.

installation, such as “kitchen,” “entrance,” and “meeting room.” Semi-static targets are usually fixed locations, but might be relocated occasionally, such as “dining table,” “sofa,” and “the desk of Ms. XXX.” Because the user might make a change to the layout of a room in home or office, the location associated with the table, sofa, or desk may be changed after the installation. The user can easily reconfigure the navigation network to fit the new layout by relocating the associated pebble. The user can also carry a pebble around to define dynamic targets, such as “father” and “Mr. YYY.” For instance, users could carry a pebble with them to continuously indicate their current locations. When one user wants to send a robot to another user, the sending user can simply specify the goal by pressing the button associated with the receiving user. Note that the sending user does not need to know where the receiving user is in this scenario. Pebbles can deal with all three scenarios in the same manner. In particular, the ability for non-experts to deal with semi-static and dynamic targets is the feature that differentiates our method from existing methods.

3.3 Specifying a Goal

After the navigation routes have been appropriately designed, the user can instruct the robot to move to any pebble within the network. A basic way to specify a goal is by pressing a button on the remote control (Fig. 6 (left)). The application services of the robot will request the user to specify a goal when required to proceed with the task. The user can also specify a goal by pressing the button on the pebble (Fig. 6 (right)). This is useful when the user want to call the robot or describe a series of target locations by traveling to the locations. If the button of the pebble is pressed, LEDs on the pebble blink more slowly than when transmitting and receiving the signals, which is described Section 3.2. The robot will reset the button press state after the task is completed, or the user can cancel the button press state by pressing the button again.

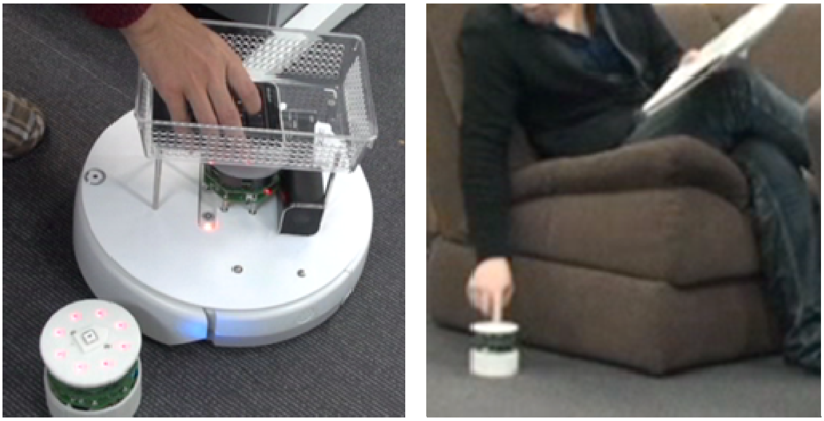


Fig. 6. Two Ways to Specify a Goal. The user can specify a goal of the robot by pressing a button on the remote control attached to the robot (left). The user can also specify a goal by pressing the button of the pebble (right).

3.4 Robot Navigation

The robot navigates by tracing infrared signals from pebbles. Therefore, the robot has the same hardware as a pebble for receiving infrared signals (Fig. 7). Our prototype employs the robotic platform iRobot Create, the developer's version of iRobot Roomba vacuum cleaner. The robot has a sound speaker for voice feedback and a roof rack for delivering objects (Fig. 7).

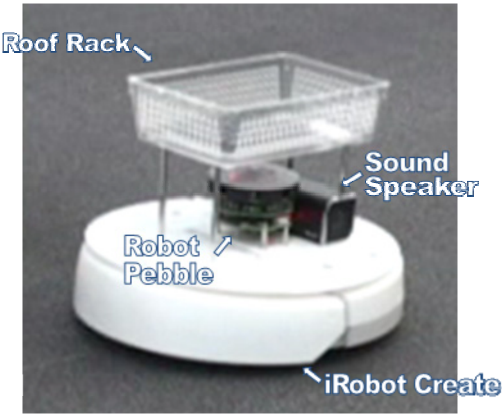


Fig. 7. Robot Platform. The robot has the same hardware as a pebble for receiving infrared signals. The robot also has a sound speaker for voice feedback and a roof rack for delivering objects.

The closer a neighbor pebble is placed, the more receivers out of eight receive signals. Using this characteristic, the robot roughly estimates its proximity to the target pebble by monitoring the number of receivers receiving signals from the target pebble. The robot has a bumper switch to detect collision and will change its direction if it detects obstacles, including pebbles. Because the bumper switch is weaker than a pebble's static friction, the robot does not push the pebble to a new location when the robot bumps into it. In this way, the robot in the current implementation can navigate past objects on the floor. Note that obstacle avoidance is beyond the topic of this paper, and if the robot is equipped with some mechanisms to detect an obstacle before collision such as a laser range finder, the robot could avoid obstacles without collision. The robot may happen to go out of line-of-sight of the network of pebbles. In such a case, the robot moves around in the local area to search for infrared signals as well as searches for another path to the final goal. If it finds the current target pebble, the robot resumes the navigation. If it finds a new path to the final goal, the robot starts a new navigation. If it cannot find the way, it stops navigation and says, "Can you help me? I'm lost."

Based on the constructed network topology, we apply Dijkstra's algorithm [27] with a uniform edge cost for finding the shortest path. All pebbles directly visible to the robot are tested as a starting point of a path. After finding the shortest path, the robot moves to each pebble in the path one by one. For example, if the shortest path is 7-3-2-5, the robot first moves toward pebble No.7. After the robot reaches pebble No.7, the robot then moves toward pebble No.3, and the robot repeats this process to finally arrive at pebble No.5.

One fundamental limitation with the current implementation is confusion caused by highly reflective surfaces such as metal-coated walls (Fig. 8(a)). We observed that the robot receives infrared signals from all directions if the robot is in a corner made of these walls and there is a pebble nearby (Fig. 8(b)). In such a case, the robot does not understand the direction of the target pebble and cannot move forward using a tracing strategy. We implement a short random walk in the local area when receiving the target pebble's signals from all directions until some of channels do not receive the signals. Although this problem can happen in an environment with highly reflective walls, it is the case that the robot happens to move into a corner, and we observed that the robot reached the final goal in most navigation trials. We confirmed that navigation can be done when a pebble is near a highly reflective wall (Fig. 8(c)). In this case, the angle estimation of the robot has slight errors, but the robot can move in a similar direction and can try the next step of angle estimation. We also confirmed that normal wallpaper-covered walls do not cause this problem.

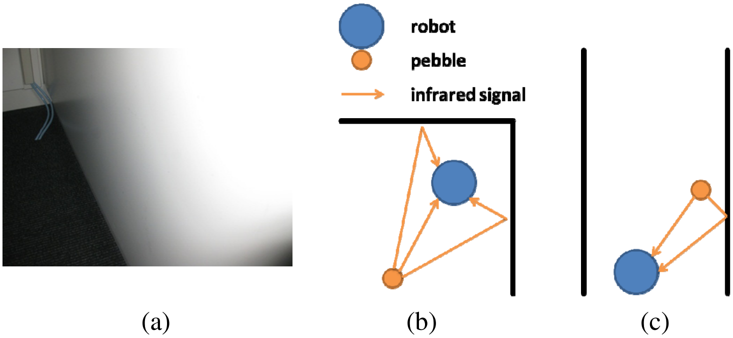


Fig. 8. Possible Limitation. If there are highly reflective walls like (a) in the environment, the robot may receive the same signal from all directions and be confused (b). On the other hand, the robot can deal with the case that a pebble is near a highly reflective wall (c).

4 Sample Deployment

We took a pebble system to a demonstration house and conducted a trial deployment and navigation in a relatively realistic environment. We took a first-time user from our group and asked her to deploy the pebble system. The goal of the deployment was to get a robot navigate between a kitchen and a studio. Fig. 9 shows part of the

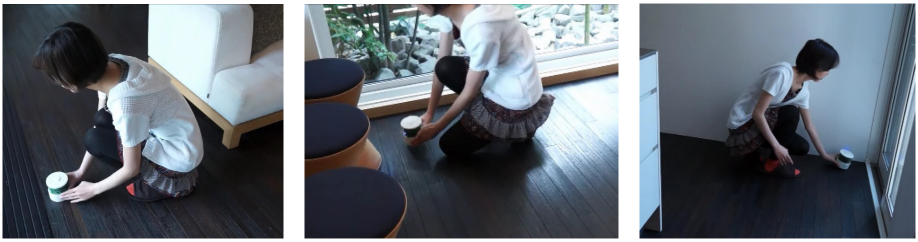


Fig. 9. Deployment Process. We took a pebble system and a first-time user to a demonstration house and asked her to design navigation routes in this environment.

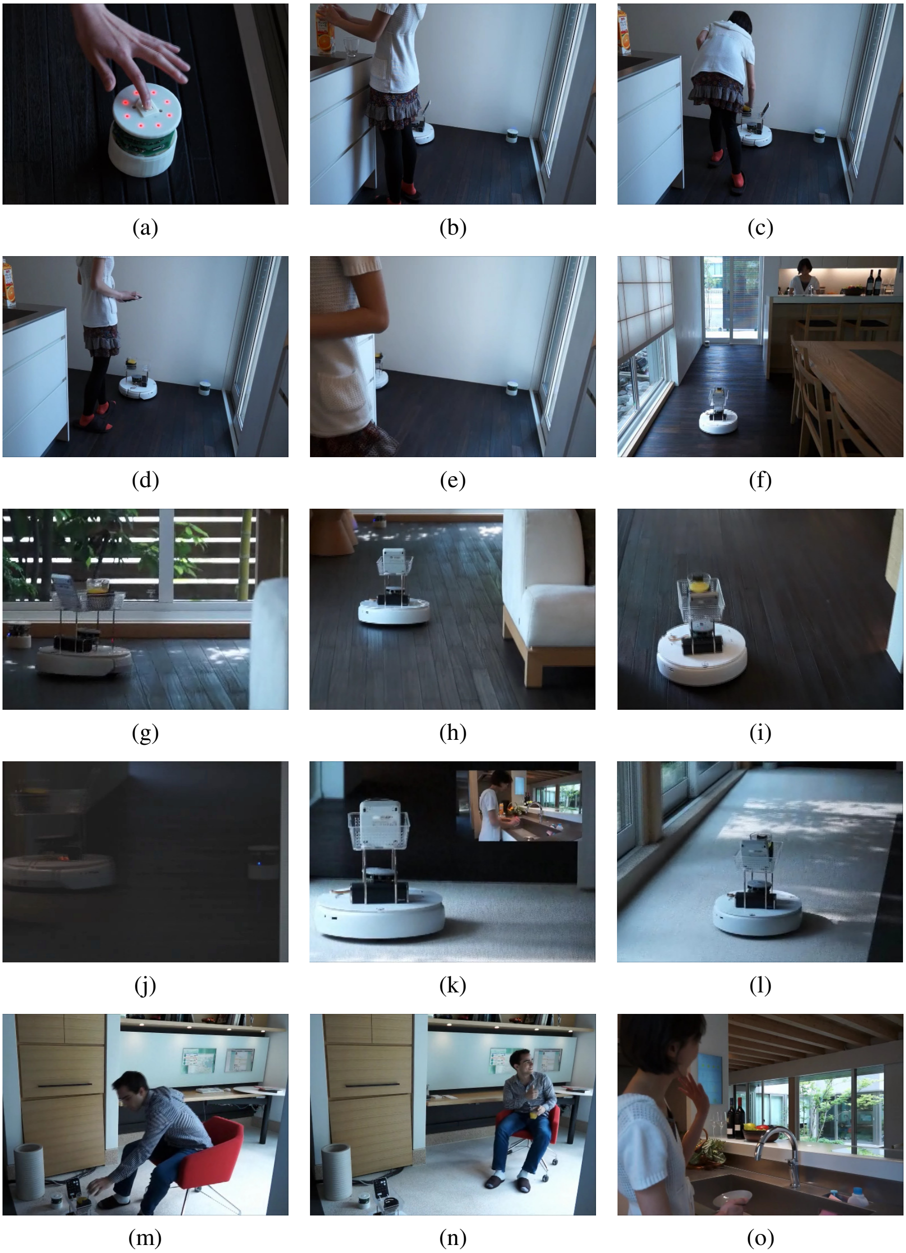

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o)

Fig. 10. Navigation Procedure. The user in a kitchen called a robot by tapping a pebble button (a). The user poured juice into a glass, while the robot was moving toward the pebble (b-c). The user directed the robot to travel to a studio using a remote control (d). The robot autonomously traveled out of the kitchen (e-f), through a living room (g-i), a passage (j-l), and the studio (m). The other user received the glass of juice (n-o). In this way, users were able to pass physical things between remote locations by employing the robot.

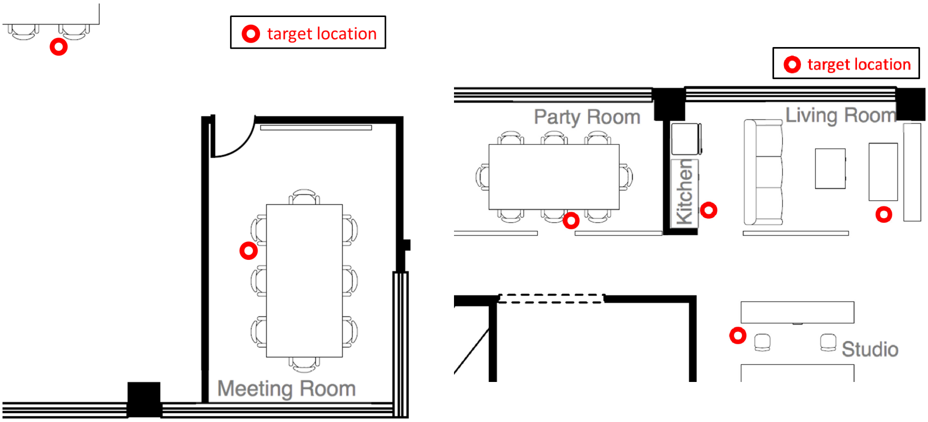
deployment process, which is just placing each pebble so that neighboring pebbles are within line-of-sight. A total of seven pebbles were placed in a kitchen, a living room, a passage, and a studio.

After deployment, we asked her to try using the robot. We gave the scenario: "A wife in the kitchen wants to deliver a glass of juice to her husband in the studio." Fig. 10 shows stills from the video record of the navigation procedure. She called the robot and directed it to deliver a glass of juice to the studio using functions derived by the pebble system. The robot successfully navigated and delivered a glass of juice, indicating that the pebble system can provide environmental knowledge for robot navigation even in an unfamiliar environment.

5 User Study

5.1 Goals and Methods

We designed a user study to test if the pebble system is understandable and usable for end-users to configure an environment. The user study involves two tasks. The first task (Task 1) is to investigate the clarity of our system. The participants are given only a user guide describing the system. Task 1 uses a room and a passage (Fig. 11(a)). The two target locations do not have a direct view between each other, and thus the participant needs to place pebbles at transit points. After Task 1, the experimenter asks about the system to determine the understanding of the participant and demonstrates full function of the system. The experimenter also accepts questions from the participant. The second task (Task 2) is intended to test the pebble system in a more complicated environment with four different room spaces and a passage (Fig. 11(b)).



(a) Task 1

(b) Task 2

Fig. 11. Experimental Configuration. We use a room and a passage for Task 1, and four rooms and a passage for Task 2.

In both studies, the participant is asked to think aloud so that the experimenter can observe the participant's thoughts. The experimenter places the robot in the environment and hands out nine pebbles that are switched off.

5.2 Detailed Procedure

In Task 1, the participant is instructed to read the user manual. After the participant finishes reading the manual, the participant is asked to configure the environment where the robot can navigate toward two target locations using pebbles. The user manual mentions how to use LED and voice feedback, but it does not instruct or encourage the participant to use the feedback. After deployment, the participant is asked to direct the robot to go to two targets using two different methods: by the remote control and by a pebble button. The participant is allowed to adjust the positions of pebbles in case the robot cannot arrive at the target. If the participant has difficulty completing the task, the experimenter regards it as the participant could not understand the system, and provides necessary help.

In Task 2, the participant is asked to design a network for the robot with four target locations. The participant is allowed to revise their deployment. After deployment, the experimenter checks to see if the network is constructed so that the robot can go to all four of the locations. After that, the participant is asked to move the robot between some of the target locations, given some scenario, such as "A wife in the kitchen wants to deliver snacks and coffee to her husband in the living room."

After Task 2, we conduct a semi-structured interview with the participant to ask opinions about their experiences using the system and how the participant thinks the system should be improved.

5.3 Participants

Eight non-expert participants (five female and three male with ages ranging from 26 to 62; 26 year old female (denoted as 26f), 28 year old male (denoted as 28m), 29m, 29m, 30f, 31f, 50f, and 62f) were invited to participate. The participants received 3000 yen, roughly 25 dollars, for taking part in the user study.

5.4 Results

Task 1. Six out of eight participants correctly deployed pebbles at appropriate places to generate navigation trails and successfully navigated the robot to the specified targets. One participant (28m) did not understand that the pebbles connected as a network. He attempted to move the robot step by step using only one pebble. He placed one pebble at a transit point to move the robot and then took the pebble to the next transit point. Another participant (26f) did not notice the visibility restriction, and the pebble network was disconnected between the inside and outside of the meeting room.

We observed a participant (29m) who tried to use pebbles in an unexpected manner. He wanted to call the robot, so he placed a pebble on the table and pressed the button as if he was pressing a service bell. However, because elevated communication is not supported by the current implementation, the pebbles on the floor did not receive the user request and the robot failed to come. Another participant (31f) attempted to place pebbles in inconspicuous places, e.g., under chairs or at corners. She argued that navigation landmarks should not disturb users' daily activities and they should not be placed at the center of the corridors.

Task2. We confirmed using the voice feedback that all participants completed an appropriate landmark deployment without the experimenter's help. During the deployment, all participants used the voice feedback feature to confirm navigation trails. The participants reported that it was not much more difficult to plan more complicated navigation trails than the ones given in Task 1. After the deployment, eight participants tested a total of 21 navigation trails. In 17 of the tested navigation trails, the robot successfully navigated to the specified place without the participant's help. In the other four navigation trails, the participants first experienced unsuccessful navigations caused by loss of the target pebble location. After adjusting the pebbles, the robot completed the navigation in all four trials.

We observed the participants' tendency to reposition the navigation landmarks several times to find more efficient routes, for example, when trying to reduce the number of pebble landmarks. We observed a participant (28m) who tried to extend the navigation network to an additional room that was not required in Task 2. These observations provide evidence to support our claims that a pebble system has the advantage of being reconfigurable, which provides users more freedom when designing navigation routes.

Interview. We conducted an interview with each participant. All participants were asked their opinions about the feedback. Overall, we obtained positive comments regarding the feedback that assists deployment, and seven out of eight participants gave a more positive evaluation of the voice feedback. One participant (30f) answered that she never watched the LED indicators; instead, she always checked the possible trails by listening to the voice feedback. We conclude that LED feedback could be improved in future revisions.

Two participants gave negative feedback for the entire system. One (31f) argued, "it seems too costly to maintain." In the current implementation, pebbles operate on battery power and require battery replacement after a certain period of time. Since most pebbles, with the exception of dynamic targets, are placed at fixed locations for a long time, we plan to revise the hardware so that power may be supplied from an outlet. The other (26f) did not want to have to "think where to place them to construct a connected network." We, however, think that deploying devices appropriately should remain the user's responsibility. A user can have all the advantages of using pebbles, such as explicitly marking areas to avoid and labeling locations, by the small work of ensuring a direct line-of-sight between each pebble, which is difficult for the system to do automatically. Some people may prefer an effort-saving method, but others may prefer a more customizable method such as the proposed method. Our proposed method provides yet another way to realize environment configuration for robots.

Most of the participants (26f, 28m, 30f, 31f, 50f, and 62f) mentioned, “it would be better if a pebble was smaller.” In the current implementation, infrared communication reaches to at least 5m, and this may be more than needed for the home and office environment. We thus think that with careful performance considerations, the device could be made smaller. We think the participants would be satisfied if we eventually made them small enough to be attached to home or office furniture.

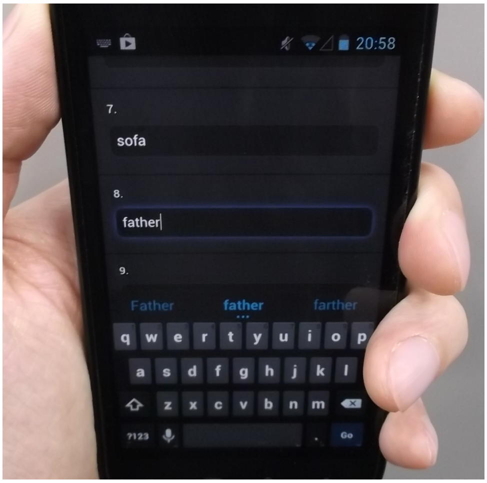
One participant (29m) commented that he would like to “call the robot using a mobile phone.” This is possible with our current implementation. When a pebble is near the user, the user just tells the system the ID of the pebble using a mobile phone. If the user is taking a pebble as a dynamic target as described in Section 3.2, any place could be the robot’s goal. As an alternative approach, a mobile phone that comes with infrared transmitters could work as a pebble to call the robot.

6 Discussion and Future Work

The proposed method provides a simple way for manually configuring an environment for robots, in particular, describing the structure of the environment and naming specific locations. This is, in a manner of speaking, the technique of “Simultaneous Labeling and Mapping.” Our method is one option for users having preferences as described in this paper, and we do not aim to completely replace existing methods with the proposed method. If a user does not care about the drawbacks of existing methods for mapping the environment, those techniques can be used to achieve more sophisticated navigation. For example, in the case where the user can receive daily expert assistance, more sensors can be installed to achieve finer robot localization. In the case where the user does not mind the robot exploring the environment, automatic mapping helps the robot to perform more efficient navigation. Conversely, labeling is fundamentally a manual process, and not well fitted for autonomous mapping. Therefore, the proposed technique can be used to add the labels to locations in the already mapped environment.

Although we proposed a simple way of mapping and labeling, non-expert but proactive users may prefer more flexible solutions. Seeking out different possibilities and providing an adequate degree of flexibility for each user’s preference is a continuing topic of interest for us. One direction is to use text data for labeling. Inputting text may be a little more complicated than attaching physical labels on buttons, but it still does not require expert knowledge. If the system stores the text of the location names, they can be used to specify locations in various forms such as typing, drop-down menu, and voice input. We are now developing a smartphone interface for label registration based on text data (Fig. 12).

Another direction is to conduct a longitudinal study that could possibly reveal more practical limitations than our user study, as well as uncover further research challenges. Ideas presented in this paper may be elaborated by an additional user study of the system.



7. sofa

8. father

9\.

Fig. 12. Labeling by Text Data. We are developing a smartphone interface to label locations. Labels based on text data can be seamlessly used to specify locations by typing, drop-down menu, or voice input.

7 Conclusion

We proposed a tangible user interface for designing navigation routes for robots. Our method is designed for non-expert users, and solves problems associated with existing methods, such as the technical complexity of labeling, invasion in unwanted areas, and complications for reconfiguration. Our user study showed that the proposed devices are easy to use and provide users with flexibility where planning customized navigation routes in simulated home and office environments.

We believe that robots, as autonomous mobility entities, improve convenience in home and office environments by providing more functionality such as monitoring and delivery, but it is difficult to realize completely autonomous robots. In this paper, we demonstrated that a little help from the user improves the functionality of the robot as well as the naturalness in specifying locations. We also assert that, besides improving performance and functionality, improving usability and other human factors is an important topic when using robots with humans.

Acknowledgments. We thank Akihiko Ishida and Greg Saul for their valuable suggestion in the initial stage of this work and their help in implementing an earlier version of the proposed system. We also thank Youichi Kamiyama for his help in implementing the proposed system. We also thank Thien-Anh Tran and Gary Mialaret for their help in testing the proposed system and taking a video at the demonstration house.

References

1. Leonard, J.J., Durrant-Whyte, H.F.: Simultaneous Map Building and Localization for an Autonomous Mobile Robot. In: IEEE/RSJ International Workshop on Intelligent Robots and Systems, vol. 3, pp. 1442–1447 (1991)
2. Shiomi, M., Sakamoto, D., Kanda, T., Ishi, C.T., Ishiguro, H., Hagita, N.: A Semi-autonomous Communication Robot - A Field Trial at a Train Station -. In: ACM/IEEE Annual Conference on Human-Robot Interaction, pp. 303–310 (2008)
3. Ishii, K., Takeoka, Y., Inami, M., Igarashi, T.: Drag-and-Drop Interface for Registration-Free Object Delivery. In: IEEE International Symposium on Robot and Human Interactive Communication, pp. 228–233 (2010)
4. Park, S., Hashimoto, S.: Indoor localization for autonomous mobile robot based on passive RFID. In: IEEE International Conference on Robotics and Biomimetics, pp. 1856–1861 (2009)
5. Park, S., Hashimoto, S.: Autonomous Mobile Robot Navigation Using Passive RFID in Indoor Environment. IEEE Transactions on Industrial Electronics 56(7), 2366–2373 (2009)
6. Mi, H., Ishii, K., Ma, L., Laokulrat, N., Inami, M., Igarashi, T.: Pebbles: An Interactive Configuration Tool for Indoor Robot Navigation. In: Annual ACM Symposium on User Interface Software and Technology, Demonstrations, pp. 11–12 (2012)
7. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-based User Location and Tracking System. In: The Conference on Computer Communications, Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 775–784 (2000)
8. Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In: Annual International Conference on Mobile Computing and Networking, pp. 70–84 (2004)
9. Nishida, Y., Aizawa, H., Hori, T., Hoffman, N.H., Kanade, T., Kakikura, M.: 3D Ultrasonic Tagging System for Observing Human Activity. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 785–791 (2003)
10. Chang, H., Choi, J., Kim, M.: Experimental research of probabilistic localization of service robots using range image data and indoor GPS system. In: IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1021–1027 (2006)
11. Niwa, H., Kodaka, K., Sakamoto, Y., Otake, M., Kawaguchi, S., Fujii, K., Kanemori, Y., Sugano, S.: GPS-based Indoor Positioning system with Multi-Channel Pseudolite. In: IEEE International Conference on Robotics and Automation, pp. 905–910 (2008)
12. Gonzalez, J., Blanco, J.L., Galindo, C., Ortiz-de-Galisteo, A., Fernaindez-Madrigal, J.A., Moreno, F.A., Martinez, J.L.: Combination of UWB and GPS for indoor-outdoor vehicle localization. In: IEEE International Symposium on Intelligent Signal Processing, pp. 1–6 (2007)
13. Ubisense, `http://www.ubisense.net`
14. Ishiguro, H.: Distributed Vision System: A Perceptual Information Infrastructure for Robot Navigation. In: International Joint Conference on Artificial Intelligence, pp. 36–41 (1997)
15. Saito, S., Hiyama, A., Tanikawa, T., Hirose, M.: Indoor Marker-based Localization Using Coded Seamless Pattern for Interior Decoration. In: IEEE Virtual Reality Conference, pp. 67–74 (2007)
16. Nakazato, Y., Kanbara, M., Yokoya, N.: Localization System for Large Indoor Environments Using Invisible Markers. In: ACM Symposium on Virtual Reality Software and Technology, pp. 295–296 (2008)

17. Miyama, S., Imai, M., Anzai, Y.: Rescue Robot under Disaster Situation: Position Acquisition with Omni-directional Sensor. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 4, pp. 3132–3137 (2003)
18. Pugh, J., Martinoli, A.: Relative Localization and Communication Module for Small-Scale Multi-Robot Systems. In: IEEE International Conference on Robotics and Automation, pp. 188–193 (2006)
19. Pugh, J., Raemy, X., Favre, C., Falconi, R., Martinoli, A.: A Fast Onboard Relative Positioning Module for Multirobot Systems. IEEE Transactions on Mechatronics 14(2), 151–162 (2009)
20. Roberts, J.F., Stirling, T.S., Zufferey, J.C., Floreano, D.: 2.5D Infrared Range and Bearing System for Collective Robotics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3659–3664 (2009)
21. Yap, T.N., Shelton, C.R.: SLAM in Large Indoor Environments with Low-Cost, Noisy, and Sparse Sonars. In: IEEE International Conference on Robotics and Automation, pp. 1395–1401 (2009)
22. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: National Conference on Artificial Intelligence, pp. 593–598 (2002)
23. Davison, A.J., Murray, D.W.: Mobile robot localisation using active vision. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 809–825. Springer, Heidelberg (1998)
24. Hwang, S.Y., Song, J.B.: Monocular Vision-Based SLAM in Indoor Environment Using Corner, Lamp, and Door Features From Upward-Looking Camera. IEEE Transactions on Industrial Electronics 58(10), 4804–4812 (2011)
25. Pirker, K., Ruther, M., Bischof, H., Schweighofer, G., Mayer, H.: An Omnidirectional Time-of-Flight Camera and its Application to Indoor SLAM. In: International Conference on Control, Automation, Robotics and Vision, pp. 988–993 (2010)
26. Ishii, H., Ullmer, B.: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In: ACM SIGCHI Conference on Human Factors in Computer Systems, pp. 234–241 (1997)
27. Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1(1), 269–271 (1959)