iCane – A Partner for the Visually Impaired

Tsung-Hsiang Chang, Chien-Ju Ho, David Chawei Hsu, Yuan-Hsiang Lee,
Min-Shieh Tsai, Mu-Chun Wang, and Jane Hsu

National Taiwan University, Computer Science and Information Engineering
http://www.csie.ntu.edu.tw
National Taiwan University, Electrical Engineering
b90106@csie.ntu.edu.tw, b90090@csie.ntu.edu.tw,
bengiu.david@gmail.com, b90097@csie.ntu.edu.tw,
b90070@csie.ntu.edu.tw, b90082@csie.ntu.edu.tw,
yjhsu@csie.ntu.edu.tw
http://www.ee.ntu.edu.tw

Abstract. Any modern society should be concerned over the well-being of the visually impaired citizen. It is the responsibility of the society to lessen the inconvenience and anxiety experienced by the handicapped. In addition to helping one to avoid obstacles on the sidewalks, technology can further help in navigating to places. In this paper we attempt to create a supportive environment with timely and useful information to guide the visually impaired to comfortably roam in a city that cares.

1 Introduction

To navigate on the streets today, the visually impaired is required to walk with the aid of *white canes* and *tactile paving*. The standard white canes are used for detecting obstacles, and tactile paving indicates the general configuration (e.g. straight path, intersection etc.) of the street in front. However, it does not provide detailed information about the pedestrian's current *location*. The main idea of the proposed design is to enrich the interaction of the visually impaired with the environment using smart sensors, intelligent software and accessible interface. The *iCane* is designed to provide its user location-based services, e.g. navigational guidance, points-of-interest referral, and mobile commerce, through a friendly voice interface.

iCane equips the standard white cane with an RFID (radio frequency identification) reader 0, which connects to a personal digital assistant with Bluetooth headphones. The reader retrieves location and relevant information, e.g. intersection, elevator and stairs, nearby shops etc., from RFID tags embedded in the environment. The user would give speech command to indicate the desired destination. The navigation service then computes the optimal path to the destination based on the current location and a map database. Navigational advice and other pertinent information are delivered to the user through the headphones at appropriate times.

The features of *iCane* are summarized as follows.

- *iCane* offers timely navigational guidance to the visually impaired.
- *iCane* issues warnings of specific road conditions (e.g. crossroad etc.) ahead.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 393 – 402, 2005.

- *iCane* helps locate nearby services, e.g., shops and government offices.
- *iCane* maintains a personal map to facilitate revisiting points of interest.

The rich information and interaction offered by *iCane* assists the visually impaired to go beyond the boundaries of their homes and familiar environments. *iCane* is a first step toward an obstacle-free space for the needed to help explore and broaden his/her experience and enjoy life in a brand new way.

2 System Overview

The basic design idea is to embed the common white cane with a location sensor. In particular, *iCane* is equipped with an RFID reader, which connects to a Personal Digital Assistant (PDA) with Bluetooth earphones. RFID tags preloaded with information are adequately placed throughout the public space. After sensing the RFID tags on the ground, the RFID reader embedded in *iCane* sends the information to the connected PDA to check for warnings (e.g. crossing points, hazard etc.) as well as additional information of interest, e.g. a virtual signboard for a nearby shop or the menu for a restaurant straight ahead. This way the visually impaired can stroll around the streets freely. In addition, given a destination, the map system is able to compute the best direction to go based on the target location. The instructions are then transmitted to the user via speech through the Bluetooth earphones. A microphone is adopted as the input device, and voice commands from the user are processed through speech recognition software. In summary, *iCane* is designed to provide rich environmental information to the visual impaired through a spoken language interface.

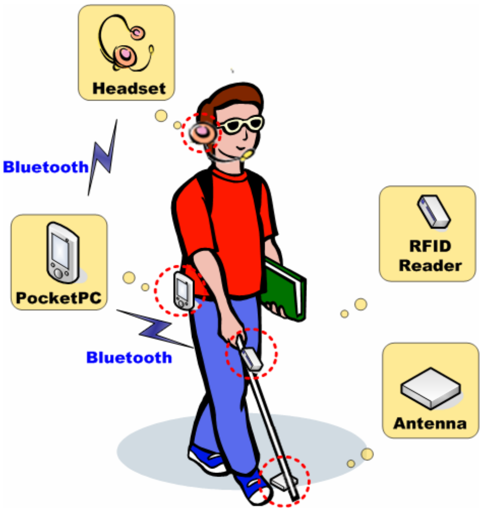


Fig. 1. The *iCane* System Overview

3 Hardware Implementation

Since the visually impaired use white canes frequently, designing white canes must be done with care. There are several issues which must be considered. First of all, the weight of the cane cannot be too heavy since the visually impaired needs to use the cane over a long distance. Second, the user should be able to carry the white cane conveniently thus it should be foldable to reduce size. Third, since the cane is designed for daily use, it must be durable. Last, the cane should be able to extend tactility of our visually impaired users. However, there are tradeoffs between *iCane* and original white cane. For example, to increase the functionality of original white cane, we might place sensors on it and thus increase the weight. Currently we adopt an existing white cane and attach our sensor and antenna to it. In order to increase the accuracy of detecting RFID tags, we place the antenna on the bottom of white cane to sense tags on the ground. For preventing damage from the environment, such as rain and accident collision, the reader is located on the handle near their hands and connected to the antenna by extended wire coiled around the body of white cane. All these components are protected in waterproof pads.

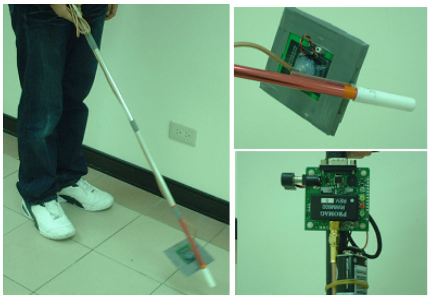


Fig. 2. On the left hand side of the figure is the *iCane* overview; on the top right of the figure is our RFID antenna; on the bottom right of the figure is our RFID reader

Location Sensor. There are several existing locating systems and one of them is GPS, which has been used for many years. But the typical error of 6 to 12 meters is not accurate enough for the visually impaired. To achieve high accuracy using GPS, the cost becomes prohibitedly high. Besides, indoor GPS doesn't perform well. An alternative locating system is the WiFi-based positioning. To make it accurate, it also requires several access points placed in a small region. For this reason WiFi technology is much more suitable in the indoor environment.

Due to the limitations in the GPS and WiFi-based positioning, we decided to adopt RFID technology. There are some advantages of adopting RFID. First, the accuracy depends on how dense we pave the tags. And the price of a passive tag is relatively low – this can reduce the cost of building infrastructure. Second, it can work in both indoor and outdoor environment, which expands the service area of the *iCane*.

RFID receiver contains RFID reader and antenna. We adopt SkyeTek EA1 to be the antenna and the RFID reader module we use is a GigaTek's product (RWM600). The RFID reader supports RS232 protocol for us to combine it with an RS232 – Bluetooth adapter. Thus the communication link between PDA and RFID reader is through Bluetooth wireless channel. The RFID reader uses a 9V battery as power supply and can operate continuously about 6 hours.

Computing Device. We choose the PDA as our computing device. The HP iPaq 5550 PocketPC is selected as the platform for *iCane*. It is connected to the RFID reader and the headset via Bluetooth technology.

Input and Output Device. We provide input and output devices for the user to communicate with the *iCane*. First, to provide an accessible interface for the visually impaired, we choose the Bluetooth headset as our user interface. This enables the users to talk to the *iCane* directly, making the application intuitive. The users can also receive instructions from *iCane* through voice guidance. Second, Sometimes people may be in a quiet place where they don't want to make noises; or they may not want others to hear where they want to go. So we provide the buttons on PDA as another input device.

4 Software Implementation

4.1 System Architecture

We design the system following the object-oriented paradigm. The system is divided into 4 layers: input, core system, subsystems, and output. Each object of the 4 layers is responsible to its own work. And except the core system, other objects can be substituted by dummy objects. This kind of architecture makes us able to build a complete system even in initial developing period, and we can make sure that the workflow of the system is correct. In the following stages, we replace these dummy objects by implemented components one by one.

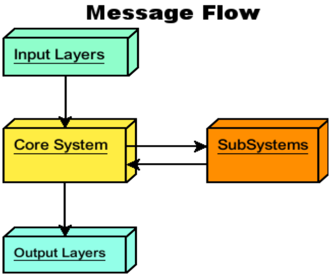


Fig. 3. Software Message Flow

The functionality of each layer is described below:

- **Input Layers:** Communicate with input devices. Wrap input data into *InputData* structure and forward them to the Core System. Input Layer contains LocationSenser, speech recognizer, hardware buttons.

- **Core System:** Forward data from Input Layer to subsystems. Wrap computed data into *OutputData* structure and send them to adequate output layer component.
- **Subsystems:** Currently there are two subsystems, and their responsibilities are listed below.
  - *User Tracker:* Track the recent user's position history and predict the user's position.
  - *Map System:* Store map data of the real world. Provide services such as path planning.
- **Output Layers:** Provide output ability to the system. Make the system able to transfer messages to users. Output Layer contains Text-To-Speech, text logger.

4.2 Software Components

Implementation details of primary components are described as follow:

LocationSenser: *GigatekReader* Adapter. We use the GigaTek RFID Reader as the main location sensor in *iCane*. The class *GigatekReader* implements the protocol that communicates with the GigaTek RFID Reader. Each time reader's tag-in event occurs, the *GigatekReader* sends out commands according to the requirements to read contents in tags. The tag ID is then converted to the coordinates by *TagDB*. The coordinates and data is wrapped into *InputData* structure and is put into the internal queue for the *CoreSystem* to take out later on.

TagDB. Tag Database stores data of all tags in the environment and provides an interface which enables others to query tag ID and its corresponding coordinates. In our first design, we stored the real world coordinates in each tag and read them out when needed (e.g., intersection, corner), and the system gave instructions that tell the user where to go. However, it takes extra time to read out the content written in tags, and the way the visually impaired use their cane may cause them to lose important information, such as intersection. According to the experiments we have done, even if we placed 3 Blister paving (15 tags per 30*30 brick) contiguously, the missing rate is still up to 20% at a relatively slow walking speed (1.5km/hr). This made us change our design. We considered storing all mapping data including tag ID and its corresponding coordinates in the PDA. Taking the map of Taipei City (where the streets are highly concentrated) for example in 2003, the total length of all streets in Taipei City was 1,536,713 meters. Suppose that we place one tactile paving every 30cm, and the data of one tactile paving consist of tag ID (8 bytes) and its coordinates (8bytes), so the total size needed for one tactile paving is 16 bytes. We also considered that there are two pavements, one on each side of the street. We are able to store the map data of all streets in Taipei City within 16×(153,671,300÷30)×2 bytes (about 156MB). Consider today's storage devices, this size is quite acceptable. So we decided to store the whole mapping data of a city in the Tag Database.

As for implementation, because the size of RAM in PDA is limited, the whole mapping data cannot be loaded into memory. So we sort the mapping data according to tag ID in a file. TagDB class does binary search in the file to find out the corresponding coordinates in the real world.

Tag Encoding. There are several kinds of tactile paving with different functionality. Directional paving and Blister paving are most commonly seen. In our system, we distinguish the two by tag ID. The reason why we do this is because the tag-in event occurs immediately after the RFID reader senses a tag, and it will read in tag ID at the same time. But if we want to read the data stored in tags' memory, it requires additional command such as select ID and read ID, which requires additional time. And reading data from tags cannot be done when the visually impaired swing their canes fast. In order not to slow down user's walking speed, tags are classified into two types. By encoding the tag ID, we can decide whether to read the data in tags just after tags are sensed.

Table 1. Tag Encoding Table. Tags are classified by the Least Significant Nibble (LSN) of tag ID and there are 10 types of tags

LSN	Function
**00	Directional paving
**10	Virtual Signboard
0001	Blister paving - intersection
0011	Blister paving – corner
0101	Blister paving – stairs
0111	Blister paving - escalator
1001	Blister paving – elevator
1011	Blister paving – ascending road
1101	Blister paving – descending road
1111	Blister paving - obstacle

CoreSystem. The *CoreSystem* examines whether the tag-in event has happened by polling the RFID reader object 20 times per second. If there is a tag-in event, The *CoreSystem* takes out the *InputData* and forwards it to *UserTracker* and *MapSystem* for further computation. It also reads out the content of a virtual signboard. And the computed data is wrapped into *OutputData* structure and sent to Output Layer.

i. MapSystem

Our map system deals with two main tasks. First, it maintains the real world's map and other environmental information. Second, it provides path-planning service that help users in navigation.

In the first part, we map addresses and other location names into geographical coordinates. We provide a query interfaces makes possible the conversion between human readable names and absolute coordinates. Our map structure is simply defined as follow: pairs of coordinates represent roads in the real world, and the map is constructed by all those roads; and a database which contains the mapping between addresses and coordinates is also maintained.

For path planning, given two locations (current location and destination), our map system finds out the shortest path for the user. Since the entire map data is stored in the system, coordinates of the two locations are available. We can than achieve path finding by the A* search algorithm, which narrows the search area with a heuristic function. Once the user sets the destination, the map system finds out the shortest path from current location. And then the map system will keep monitoring if the user is on this shortest path. If the user moves out of the planned path, the map system automatically computes a new shortest path immediately.

ii. UserTracker

The *UserTracker* consists of a location history which records the recent location information of the user. We use the location history to approximate the user's walking speed and predict the current location of the user. If the user is closing to the Blister paving, The *UserTracker* can remind the user to slow down for getting important road condition information.

User Interface (*Speech Recognizer*, PDA Buttons, *TTS*). The user interface consists of two parts: input and output. The output is presented in the form of speech, while the input part consists of speech and PDA buttons. The interaction between *iCane* and users can be accomplished by speech, buttons or both.

In order to achieve an easy-to-use interface, the input method through PDA control buttons is designed with simplicity in mind. There are two direction buttons (for menu selection) and two command buttons (for confirm and cancel). It functions like a cell phone when you input through PDA buttons. You can choose the function with direction buttons and then press the confirm button to execute it.

Also, speech interface is provided. Through speech recognition, user can read out specific words to act out the desired function without pressing any buttons. To process speech recognition, we use the NeuVoice Audient Pocket PC as the developing SDK. As for the output, all output messages including virtual signboard, prompting messages, and instruction messages are delivered to the user by the Text-To-Speech module.

5 Experiment Results

To assure that every component works correctly, unit test class for each main component is built. Besides, test cases which simulate real environment are also built through RFID emulator. By using the Text Logger, we could examine that if the system output is the same as we have anticipated.

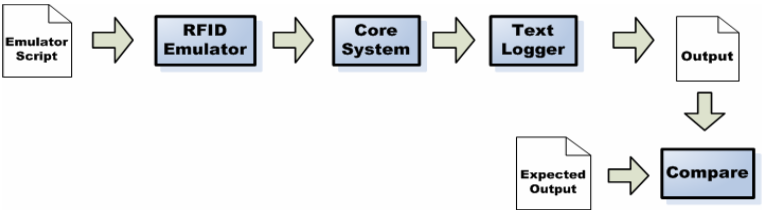
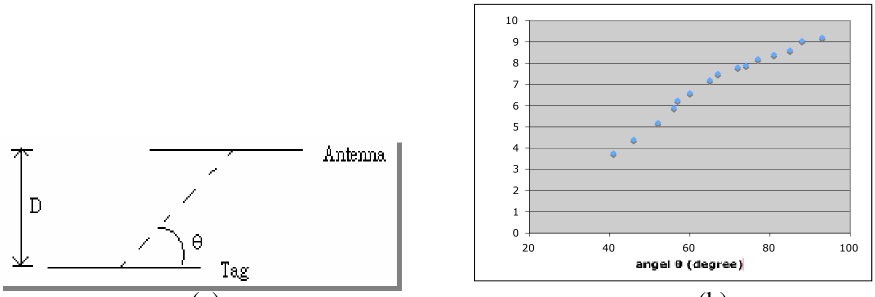


Fig. 4. Software Testing Automation

RFID Emulator. In order to let the system work without real RFID reader in early development, we designed an RFID emulator which simulates behaviors of real RFID reader. The benefit is that we can build up a complete system that is workable in a short time. Besides, test scripts which simulate real environment could be designed through the emulator, and automatic testing to the whole system could be done.

Assistant Tool for Paving Tags. To support infrastructure, a large number of tags need to be paved in the environment, and the tag ID and its corresponding coordinates should also be recorded. To construct a simulating environment efficiently, an assistant tool has also been developed. The assistant tool simply asks the RFID reader to read in tags on the street sequentially, automatically computes the corresponding coordinates of each tag, and saves them in the Tag Database. With the aid of this tool, we can place the RFID reader on a cart. After walking along all the streets with the cart, the mapping of tag ID and coordinates is then computed.

RFID Sensing Profile. The sensing range of RFID technology is limited, and varies with the angel between the antenna and the line connecting the antenna and the tag. We define θ to be the angle between the line connecting the centers of the tags and the antenna, and D the vertical distance between the centers of the tags and the antenna, as shown in Fig. 5(a).



(a) (b)

Fig. 5. (a) Definition of D and θ (b) Detectable Distance and Angel

If we measure D under various θ, we can get their relationship as shown in Fig. 5(b). D increases as θ increase and reaches the maximum when θ = 90°.

As the above experiment shows, the tag should be near by the antenna when the cane contacts the ground (no longer then 5cm) in order to be sensed. Besides, the length of the tag is 9.6cm, so we decide to pave tags every 15cm to ensure that there will be at least one tag in the sensing range if we walk along the tactile paving.

***iCane* Sensing Rate.** The main factor that will affect the reliability of our system is the sensing rate of the tag. In order to assure data correctness, the location information is expected to be updated as often as possible, i.e., the distance between each tag being sensed should be as short as possible.

On testing the sensing rate, two issues should be taken into consideration. First, the way users wave their canes. Second, the speed at which the user walks. In the first case, the visually impaired needs be taught how to wave a cane in three gestures. In our tests, we elect just one of the waving gestures: knocking at the ground in the walking path of the user and waving high by two sides to detect the obstacles. And the walking speed is fixed at 3 km/hr – about the walking speed of an adult.

We have done some experiments to measure the sensing interval and analyze the results. Our experiment environment was set as a 10 meter-long tactile paving assembling by 30 cm x 30 cm movable paving. The experiment procedure is, first, the user setting up with *iCane* system walked through the tactile paving as stable as possible and we recorded the interval between the detected paving. Then, we repeated this experiment for 40 times and established a figure showing probability distribution of the sensing interval. The result shows that the average sensing interval is about 1.39 meters. In most of the case (80%), users can sense the next tag in a reasonable interval - less than 2 meters. Therefore, this result confirms the feasibility of *iCane* system.

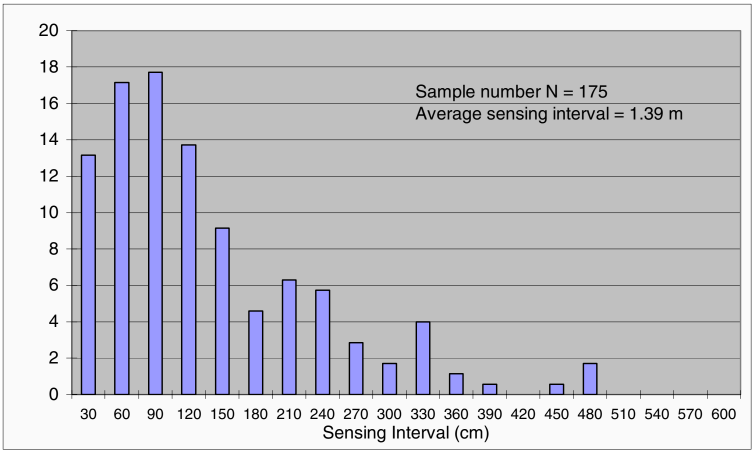


Fig. 6. Distribution of the Sensing Interval

Software prediction is used to improve the reliability of the *iCane* System. The software prediction method approximates the user's walking speed using previously sensed tag locations in the tag history. If the user is closing to the Blister paving, he will be notified to slow down his walking speed.

6 Conclusion

Using *iCane*, visual information about the public space now becomes available for the visually impaired. With the help of RFID technology, we are able to retrieve richer information and present it in more accessible forms, e.g. voice, to the sightless user. In addition, location-based services such as path finding are provided to help the visually impaired in unfamiliar environments.

The *iCane* system will become more feasible with the maturity of the RFID technology. It is a first step toward an obstacle-free space for everyone to explore and broaden his/her experience and enjoy life in a brand new way. The visually impaired is no longer confined within the limits of their homes and familiar environments. The rich information and interaction offered by *iCane* open up many new possibilities and allows the visually impaired to go beyond the boundaries of their disability.

Acknowledgements

This research is supported in part by a grant (NSC-93-2218-E-002-148) from the National Science Council in Taiwan and a grant from Intel.

We would like to thank Jane Hsu, our advisor, John Wang, who made revision in our first draft paper and Cultural & Educational Foundation for Blind, who gave us comments during our research.

References

1. “RFID Journal.” [Online] Available http://www.rfidjournal.com/, 2005.
2. Klaus Finkenzeller. RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition. John Wiley & Sons, Inc., 2003.
3. Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach, Second Edition. Pearson Hall, pp. 97-110, 2003.
4. “Speech Recognition SDK for Windows” [Online] Available http://www.neuvoice.com/products/audientppc.php, 2005.
5. Iwan Ulrich, Johann Borenstein. “The GuideCane-Applying Mobile Robot Technologies to Assist Visually Impaired.”, IEEE Transactions on Systems, Man, and Cybernetics, —Part A: Systems and Humans, Vol. 31, No. 2, pp. 131-136, March 2001.
6. Vladimir Kulyukin, Chaitanya Gharpure, Nathan De Graw, John Nicholson, Sachin Pavithran. “A Robotic Guide for the Visually Impaired in Indoor Environments.” Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-04), San Jose, CA, 2004.
7. Tomohiro Amemiya, Jun Yamashita, Koichi Hirota, Michitaka Hirose, "Development of Virtual Leading Blocks for the Blind", In Proc. of VRSJ 8th Annual Conf., pp. 359-362, VRSJ 8th Annual Conf., Gifu, 2003.9.(In Japanese)
8. “PocketPC (WinCE) GPS and Navigation.” [Online] Available http://www.gpsinformation.org/dale/PocketPC/wince.htm, 2005.
9. “MapServer Homepage.” [Online] Available http://mapserver.gis.umn.edu/index.html, 2005.