RoboCart: Toward Robot-Assisted Navigation of Grocery Stores by the Visually Impaired*

Vladimir Kulyukin   Chaitanya Gharpure   John Nicholson
Computer Science Assistive Technology Laboratory
Department of Computer Science
Utah State University
Logan, Utah, U.S.A
vladimir.kulyukin@usu.edu,{cpg, jnicholson}@cc.usu.edu

***Abstract*— This paper presents RoboCart, a proof-of-concept prototype of a robotic shopping assistant for the visually impaired. The purpose of RoboCart is to help visually impaired customers navigate a typical grocery store and carry purchased items. The current hardware and software components of the system are presented. For localization, RoboCart relies on RFID tags deployed at various locations in the store. For navigation, RoboCart relies on laser range finding. Experiences with deploying RoboCart in a real grocery store are described. The current status of the system and its limitations are outlined.**

***Index Terms*— service robotics**

I. Introduction

Grocery shopping is a routine activity that people all over the world perform on a regular basis. However, grocery stores and supermarkets remain largely inaccessible to people with visual impairments. The main barrier is the inadequacy of the principal navigation aids, such as guide dogs and white canes, for negotiating the seemingly simple topological structure of a typical grocery store. While guide dogs are helpful in micro-navigation, e.g., local obstacle avoidance and homing in on simple targets such as exit doors, they offer little assistance with macro-navigation, which requires topological knowledge of the environment. A guide dog may memorize routes to a small set of grocery items through repeated exposure to those routes. However, the guide dog cannot help its handler in a routine situation when the store changes the location of several items between visits or stops carrying a product. Nor can the dog assist its handler with pushing a shopping cart. Needless to say, the white cane does not fare any better under the same circumstances.

In August 2004, the Computer Science Assistive Technology Laboratory (CSATL) of the Department of Computer Science (CS) of Utah State University (USU) and the USU Center for Persons with Disabilities (CPD) started a collaborative project whose objective is to build a robotic shopping assistant for the visually impaired. The purpose of the shopping assistant is not to do the grocery shopping for the visually impaired. Rather, the purpose is to assist visually impaired customers in navigating the store and carrying the purchased items around the store and to the check-out registers.

Fig. 1.   RoboCart in a grocery store.

This paper describes RoboCart, a proof-of-concept prototype of a robotic grocery shopping assistant for the visually impaired. RoboCart is shown in Fig. 1. The paper is organized as follows. First, related work is reviewed. Second, RoboCart's current software and hardware components are presented. Third, the scope and feasibility issues are discussed. Finally, the status of the system is described through experiences with deploying RoboCart in a supermarket.

II. Related Work

Several researchers have developed robotic guides for indoor environments. While these systems are not explicitly designed for the visually impaired, they offer valuable fea-

*This work is supported, in part, by NSF Grant IIS-0346880 and, in part, by two Community University Research Initiative (CURI) grants (CURI 2003 and CURI 2004) from the State of Utah.

sibility studies in robotic navigation of indoor environments. Horswill [6] built Polly, a small robotic guide for the MIT AI Lab. Polly used lightweight vision routines that depended on textures specific to the lab. Burgard et al. [2] developed RHINO, an interactive tour guide that was deployed in a museum in Germany. Thrun et al. [16] used the approaches developed in RHINO to build MINERVA, an autonomous tour-guide robot that was deployed in the National Museum of American History in Washington, D.C.

Borenstein and Ulrich introduced GuideCane [7], a mobile obstacle avoidance device for the visually impaired. GuideCane consists of a long handle and a sensor unit mounted on a steerable two-wheel axle. The sensor unit consists of ultrasonic sensors that detect obstacles and help the user to steer the device around them. A steering servo motor, operating under the control of the onboard computer, steers the wheels relative to the handle.

The Haptica Corporation developed Guido, a robotic walking frame for people with impaired vision and reduced mobility [17]. Guido uses the onboard sonars to scan the immediate environment for obstacles and communicates detected obstacles to the user via speech synthesis. Mori and Kotani [12] developed HARINOBU-6, a robotic travel aid to guide the visually impaired on the Yamanashi University campus. HARINOBU-6 is a motor wheel chair equipped with a vision system, sonars, a differential GPS, and a portable GIS.

RoboCart itself grew out of the Robot-Assisted Navigation project currently under way at the USU CSATL [9]. The project has developed a robotic guide for the visually impaired for indoor office environments. The guide, whose name is RG, which abbreviates *robotic guide*, was successfully deployed and tested with visually impaired participants in two dynamic and complex indoor environments [9], [10].

Insomuch as RoboCart's localization is based on Radio Frequency Identification (RFID), RoboCart is related to several robotics projects that used RFID for indoor navigation. Kantor and Singh used RFID tags for robot localization and mapping [7]. Once the positions of the RFID tags are known, their system uses time-of-arrival type of information to estimate the distance from detected tags. Tsukiyama [15] developed a navigation system for mobile robots using RFID tags. The system assumes perfect signal reception and measurement and does not deal with uncertainty. Hahnel et al. [4] developed a robotic mapping and localization system to analyze whether RFID can be used to improve the localization of mobile robots in a simple office environment. They proposed a probabilistic measurement model for RFID readers that accurately localizes RFID tags in a simple office environment. The SpotOn system developed at the University of Washington [5] is an RFID-based localization system for indoor environments. The system relies on the strength of the signal from RF tags and uses triangulation to estimate their positions. Another RFID-based navigation system for indoor environments was developed at the Atlanta VA Rehabilitation Research and Development Center [13], [14]. In this system, the blind users' canes are equipped with RFID receivers, while RFID transmitters are placed at hallway intersections. As the users pass through transmitters, they hear over their headsets commands like *turn left*, *turn right*, and *go straight*.

Fig. 2. RFID tag in a grocery store.

Since RoboCart is designed to exploit structural regularities present in grocery stores, the overall design philosophy is similar in spirit to the research conducted by Fu et al. [3] on action and perception in man-made environments. Fu et al. [3] developed SHOPPER, a simulated robot that uses the structural and functional regularities of grocery stores to optimize and simplify its sensing routines to shop efficiently.

III. Hardware and Software

Like its predecessor, RG [9], RoboCart is built on top of the Pioneer 2DX robotic platform from the ActivMedia Corporation (See Fig. 1). RoboCart's navigation system, called a Wayfinding Toolkit (WT), resides in a polyvinyl chloride (PVC) pipe structure mounted on top of the platform. The WT includes a Dell^{TM} Ultralight X300 laptop connected to the platform's microcontroller, a laser range finder from SICK, Inc., and to a radio-frequency identification (RFID) reader from Texas Instruments, Inc. The TI Series 2000 RFID reader is connected to a square $200mm \times 200mm$ antenna. Fig. 2 shows a TI RFID Slim Disk tag attached to a grocery store shelf. These tags can be attached to any objects in the environment or worn on clothing. They do not require any external power source or direct line of sight to be detected by the RFID reader. The tags are activated by the spherical electromagnetic field generated by the RFID antenna with a radius of approximately 1.5 meters. Each tag is programmatically assigned a unique ID. When detected, RFID tags enable and disable local navigation behaviors, e.g., follow-aisle, turn-left, turn-right, avoid-obstacle, make-u-turn, etc.

The basic philosophy behind this approach is to alleviate localization and navigation problems of completely autonomous approaches by instrumenting environments with inexpensive and reliable sensors that can be placed in and out of environments without disrupting any indigenous activities.

RoboCart's software architecture currently includes three components: a user interface (UI), a path planner, and a behavior manager. The UI's input is entered by the user from a hand-held keypad. The original UI also had speech input. However, speech was abandoned after unsuccessful trials with visually impaired participants [8]. The UI's output mode uses speech synthesis. Destinations entered by the user from the keypad are turned into goals for the path planner. The user can learn the available commands from a Braille directory, a roll of paper with Braille signs, that attaches to the handle at the back of the robot. The directory contains a mapping of key sequences to destinations, e.g., produce, deli, coffee shop, etc.

The path planner and behavior manager are inspired by and partially realize Kupiers' Spatial Semantic Hierarchy (SSH) [11]. The SSH is a framework for representing spatial knowledge. It divides spatial knowledge of autonomous agents, e.g., humans, animals, and robots, into four levels: the control level, causal level, topological level, and metric level. The control level consists of low level mobility laws, e.g., trajectory following and aligning with a surface. The causal level represents the world in terms of views and actions. A view is a collection of data items that an agent gathers from its sensors. Actions move agents from view to view. The topological level represents the world's connectivity, i.e., how different locations are connected. The metric level adds distances between locations.

The path planner realizes the causal and topological levels of the SSH. It contains the declarative knowledge of the environment and uses that knowledge to generate paths from point to point. The behavior manager realizes the control and causal levels of the SSH. The control level is implemented with the following low-level behaviors all of which run on the WT laptop: follow-aisle, turn-left, turn-right, avoid-obstacles, and make-u-turn. These behaviors are written in the behavior programming language of the ActivMedia Robotics Interface for Applications (ARIA) system from ActivMedia Robotics, Inc.

The behavior manager also keeps track of the robot's global state. The global state is shared by all the modules. It holds the latest sensor values, which include the laser range finder readings, the lastest detected RFID tag, current velocity, current behavior state, and battery voltage. Other state parameters include: the destination, the command queue, the plan to reach the destination, and internal timers. The other modules use the current state in two ways: 1) to access and

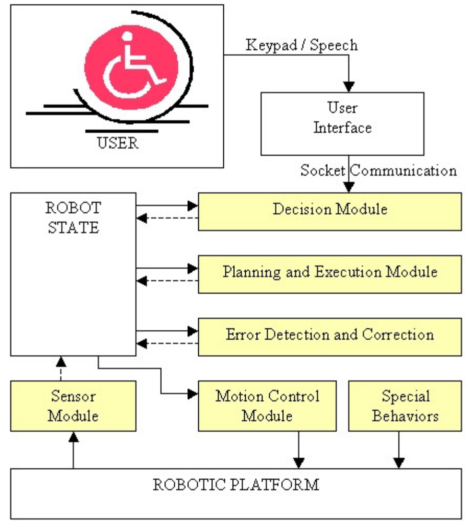

Fig. 3. Software architecture.

update the latest sensor readings and 2) to post messages for other modules.

The behavior manager, in turn, consists of six modules: sensor module (SMod), motion control module (MCMod), special behaviors module (SBMod), error detection and correction module (EMod), planning and execution module (PEMod), and decision module (DMod).

The SMod is the interface to the robot's sensory system. The SMod uses the ARIA's API to obtain the latest sensor readings. The MCMod partially corresponds to the control level of the SSH. This module deals with the robot's navigation behaviors, such as following aisles and making left, right and u-turns. The behaviors are enabled and disabled by RFID tags and the laser range finder's readings. For navigation, the MCMod uses a potential fields approach in conjunction with a greedy selection of empty spaces described in detail elsewhere [9].

The SBMod falls into the control level of the SSH. Behaviors like entering and exiting elevators are treated as special behaviors since they require a different navigation sequence and more frequent interaction with the user. Currently, the SBMod implements the negotiate-elevator behavior.

The EMod partially realizes the control and causal levels of SSH. The EMod checks for errors in terms of lost paths. The EMod relies on the RFID tags seen along the path and the behavior sequence stored in the robot's state. If the EMod detects a lost path, it sends an *icmd_lost* command to the DMod that causes the robot to replan and resume on user permission. It should be noted in passing that all commands starting with *icmd* are internal commands, i.e., generated by the robot. The PEMod is responsible for the stepwise execution of the plan received from the planner. It keeps checking if the tags detected are part of the current plan. As soon as a valid tag is detected, the corresponding behavior is triggered. It generates the commands, such as *icmd_left*, *icmd_right*, *icmd_uturn*, based on what tags are detected.

The DMod partially realizes the causal level of the SSH. It polls the command queue for new commands and triggers the appropriate behaviors when a new command is received. It deals with two types of commands: navigation commands, such as "go to location X," "pause," "resume," "cancel" (cancelling a run), and information commands like "where am I?". The DMod is also responsible for the clean-up work for the *icmd_stop* and *icmd_lost* commands. The *icmd_stop* command is generated when the destination tag is encountered. The *icmd_lost* command is generated when the EMod detects that the robot is lost.

The UI translates all commands into an internal command representation for the DMod. For example, if location X is mapped to tag 17, the command "go to location X," is received by the DMod as "tag:17." As soon as this command is received, the DMod obtains a plan from the planner and sends an *icmd_move* command to the MCMod. On receiving the commands "pause," "resume," "cancel," the DMod sends the *icmd_pause*, *icmd_resume*, and *icmd_stop* commands, respectively, to the MCMod. These commands make the robot change its internal behavior states.

A. Knowledge representation

Knowledge engineering for the path planner is done in OpenCyc [18], a free knowledge engineering tool from the Cyc Corporation that allows one to represent common sense knowledge using first order logic. The knowledge base represents an aerial view of the store in which RoboCart operates. Currently, the knowledge base consists of tag connectivity graphs, tag to destination mappings, and low-level behavior scripts associated with specific tags. The environment is represented as a graph where nodes represent the RFID tags and the edges represent the behaviors required to travel from one tag to another. Views consist of laser range finder readings and the IDs of the RFID tags currently detectable. For example, the following assertion in the OpenCyc's knowledge representation language CYCL represents a graph node:

```
(#$rfidTag 10
  (#$TheList
    (#$TheList 11
      (#$sMakeAisleUturn)
      (#$sFollowAisle 1))
    (#$TheList 12
      (#$sFollowAisle 1)))
  (#$TheList 4))
```

This assertion states that this node is represented by the RFID tag whose ID is 10. The second argument to the predicate `#$rfidTag` is a list of nodes that can be reached from node 10. In this example, from node 10 one can reach nodes 11 and 12. Each reachable node has a sequence of behaviors associated with it. A single behavior is represented by a predicate, such as `#$sFollowAisle`. The only argument to the `$sFollowAisle` predicate is 0 or 1 with 0 standing for the left side and 1 standing for the right side. In the above example, tag 11 can be reached from tag 10 by first making a u-turn and then following the right side of the aisle until tag 11 is detected. Similarly, tag 12 can be reached from tag 11 by following the right side of the aisle in the same direction.

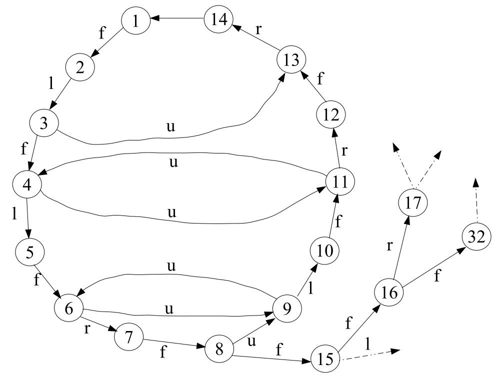

Fig. 4. A tag connectivity graph.

While OpenCyc provides a great deal of versatility and power in declarative knowledge engineering, experiments with RoboCart demonstrated that the OpenCyc server cannot meet the real time commitments necessary for a deployed robotic system. Consequently, for efficiency reasons, all OpenCyc representations, after they are finalized, were converted into a set of semantically equivalent text files. Specifically, these text files contain a set of Lisp-like s-expressions that contain the same information on the topology of a given environment. For example, *(path 12 13 f)* asserts that tags 12 and 13 are connected and that tag 13 can be reached from tag 12 by following the hallway. Thus, a grocery store is represented as a connected graph similar to the one shown in Fig. 4.

B. Plan execution

Given the connectivity graph, the planner uses a standard breadth first search (BFS) to find the shortest path from the start tag to the destination tag. The path is a *tag-behavior-tag* sequence. For example, if the start tag is 4 and the destination tag is 17, *(4 l 5 f 6 r 7 f 8 f 15 f 16 r 17)* is a path. The PEMod uses this plan to guide RoboCart along the path.

The behavior manager treats a plan as a sequence of records with three fields: *prev-tag*, *behavior*, and *next-tag*. In other words, a plan is a sequence of triples $< T_i, B_{ij}, T_j >$, where T_i and T_j are the tags connected by the behavior B_{ij}. If T_i and the currently detected tag are identical, the current tag is valid and triggers the behavior B_{ij}, which, if successfully executed, allows the robot to detect T_j, at which point B_{ij} is continued, disabled, or replaced with a new behavior.

As an example, consider the plan *(1 f 2 r 3)*. As soon as the valid tag 1 is detected, *follow-aisle* is triggered. This is the guide's default behavior when an *icmd_move* command is given. RoboCart follows the aisle until tag 2 is detected. The detection of tag 2 is followed by the generation of the *icmd_right* command, which triggers *turn-right*. The turning behavior is complete when shelves are detected on both sides by the laser range finder within a specified range.

IV. Scope and Feasibility

Several important issues will be beyond the project's scope. First, RoboCart guides its users to locations, but not inside location. For example, RoboCart will guide its user to a restroom and wait for her near the entrance, but will not guide the user inside the restroom.

Second, RoboCart is not meant for individual ownership. The intention is not to replace the guide dog or white cane. The intention is to help the dog's handler or the white cane user to overcome the limitations of their navigational aids in grocery stores. Dynamic and complex indoor environments, such as grocery stores and airports, are both feasible and socially valid for robot-assisted navigation. Guide dogs, white canes, and other navigation devices are of limited use in such environments, because they cannot help their users localize and find paths to useful destinations. Furthermore, such environments can be instrumented with small sensors that make robot-assisted navigation feasible.

Third, it is assumed that institutions, such as grocery stores and airports, will operate such guides on the premises in the future. Estimates from the manufactures of the hardware components that comprise the Wayfinding Toolkit indicate that, when produced in large quantities, the cost of the toolkit may be 3,500-4,000 USD. This is an affordable price for grocery stores and airports. For example, each WalMart store already has 2-3 assisted mobility vehicles called MartCarts(TM) to which wayfinding toolkits can be attached in the future. The same modifications can be done to SmartCarts(TM) available at all major airports.

Fourth, there is no plan to extend the domain of RoboCart to outdoor environments. The state of the art in outdoor robot navigation technology does not allow one to reliably navigate outdoor environments. Additionally, the expense of deploying and maintaining such systems is prohibitive to many organizations. Given the state of the art in outdoor localization and computer vision, outdoor environments are not currently within reach of robots unless the robots are teleoperated, at least part of the time.

Fifth, RoboCart, at this point, cannot assist its customers with picking specific items from shelves. Finding a specific item on a shelf is a wearable computing problem that has little to do with navigation. A solution to that problem is likely to use an entirely different set of sensing mechanisms, e.g., a glove-embedded barcode scanner. In sum, RoboCart will lead its user to the shelf that contains the needed item but will not assist the user in finding that item on the shelf.

Fig. 5. RoboCart during a trial run.

Fig. 6. RoboCart's handle.

V. Status

RoboCart has been repeatedly deployed in Lee's MarketPlace, a grocery store in Logan, Utah, over a period of five months, from August 2004 to December 2005. Fig. 5 shows RoboCart during a trial run in the grocery store. Continuous deployment in the store was not logistically possible. Therefore, the research team would bring the robot to the store for 2-3 hours at a time, deploy the robot and the RFID tags, and run the robot on various routes. The area in which the robot was tested covered a total of five aisles. The tags were placed on both sides of the aisle to mark different items as well as the beginning and end of each aisle.

The original design included the possibility for the user to follow RoboCart holding a dog leash attached to the robot's battery bay. However, several visually impaired people indicated that they would prefer something more static so that they could get more haptic feedback on the direction and motion of the robot. The dog leash was subsequently replaced with a static PVC handle shown in Fig. 6. Several visually impaired people indicated that the handle is better than the dog leash.

Instrumentating environments with RFID sensors is fast and requires only commercial off-the-shelf (COTS) hardware and software components. RFID tags are inexpensive (15 USD a piece), reliable, and easy to maintain, because they do not require external power supplies. RFID provides accurate localization. Although RFID tag reading failures occur, they are rare and can be recovered from as long as other RFID tags are placed in the environment. The placement of RFID tags in the environment does not seem to disrupt any indigenous activities. People who work in the grocery store did not seem to mind the tags due to their small sizes.

During the trial runs it became clear that it was difficult to do u-turns in store aisles. The reason for this is that it is hard for the user to hold on to the handle when the robot executes a u-turn. An additional complication is other shoppers in the same aisle that may be in the way of the handle. Lack of knowledge of when executing a u-turn is appropriate is a limitation of the system.

Another limitation is that the spacial sensing of the system is confined to a horizontal plane 180 degrees in front of the robot and approximately 50 centimeters from the floor, i.e., the signal plane of the robot's laser range finder. This allows the robot to avoid shoppers and boxes in the aisles. However, it makes it impossible for the robot to detect advertisement boards placed above 50 centimeters and attached to the shelves. Thus, it is possible for the basket mounted on the robot to collide with a solid object placed above the horizontal plane of the robot's laser range finder. Additional range sensors are needed that will scan the immediate space upward. It might also be possible to extract more information from laser range signatures.

Finally, no systematic evaluations of the system with visually impaired participants have so far been conducted. Such evaluations are planned in the future. These evaluations will undoubtedly lead to modifications of the current hardware and software solutions.

Acknowledgments

The authors would like to thank Mr. Lee Badger, the owner of Lee's MarketPlace, a supermarket in Logan, Utah, for allowing them to use his store for testing RoboCart. The authors would like to thank Dr. Sarah Rule, Director of the USU Center for Persons with Disabilities, and Mr. Martin Blair, Director of the Utah Assistive Technology Program, for administrative support and assistance. The authors would like to thank Mr. Sachin Pavithran, a visually impaired assistive technology specialist at the USU Center for Persons with Disabilities, for generously volunteering his time for testing RoboCart. Finally, the authors would like thank Bharath Ramaswamy, a graduate EE student at USU, for many helpful hardware suggestions.

References

[1] J. Borenstein and I. Ulrich, "The GuideCane - a computerized travel guide for the active guidance of blind pedestrians," IEEE International Conference on Robotics and Automation, 1994, San Diego, CA: IEEE Press.

[2] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an Interactive Museum Tour-Guide Robot," Artificial Intelligence, 114:3-55, 1999.

[3] D. Fu, K. Hammond, and M. Swain, "Action and perception in man-made environments," 14th International Joint Conference on Artificial Intelligence (IJCAI-95), August 1995, Montreal, Quebec, Canada: AAAI/MIT Press.

[4] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and Localization with RFID Technology," Technical Report IRS-TR-03-014, December 2003, Intel Research Institute, Seattle, WA.

[5] J. Highthower, R. Want, and G. Borriello "SpotOn: an indoor 3D location sensing technology based on RF signal Strength," Technical Report CSE 2000-02-02, 2000, University of Washington, Seattle, WA.

[6] I. Horswill, "Polly: a vision-based artificial agent," 11th Conference of the American Association for Artificial Intelligence (AAAI-93), Washington, D.C.: AAAI/MIT Press.

[7] G. Kantor and S. Singh, "Preliminary results in range-Only localization and mapping," IEEE Conference on Robotics and Automation, May 2002, Washington D.C.: IEEE Press.

[8] V. Kulyukin, C. Gharpure, P. Sute, N. De Graw, J. Nicholson, and S. Pavithran, "A Robotic wayfinding system for the visually impaired," Innovative Applications of Artificial Intelligence (IAAI-04) Conference, July 2004, San Jose, CA: AAAI/MIT Press.

[9] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran, "RFID in robot-assisted indoor navigation for the visually impaired," IEEE/RSJ Intelligent Robots and Systems (IROS 2004) Conference, September - October 2004, Sendai, Japan: Sendai Kyodo Printing Co.

[10] V. Kulyukin, C. Gharpure, N. De Graw, J. Nicholson, S. Pavithran, "A Robotic guide for the visually impaired in indoor environments," Rehabilitation Engineering and Assistive Technology Society of North America (RESNA 2004) Conference, June 2004, Orland, FL: Avail. on CD-ROM.

[11] B. Kupiers, "The Spatial Semantic Hierarchy," Artificial Intelligence , 119:191-233, 2000.

[12] H. Mori and S. Kotani, "Robotic Travel Aid for the Blind: HARUNOBU-6," Second European Conference on Disability, Virtual Reality, and Assistive Technology, 1998, Sövde, Sweden.

[13] D.A. Ross, "Implementing assistive technology on wearable computers," IEEE Intelligent Systems, May-June:2-8, 2001.

[14] D.A. Ross and B.B. Blasch. "Development of a wearable computer orientation system," IEEE Personal and Ubiquitous Computing, 6:49-63, 2002.

[15] T. Tsukiyama, "Navigation System for the Mobile Robots using RFID Tags," IEEE Conference on Advanced Robotics, June-July 2003, Coimbra, Portugal.

[16] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, C. N. Roby, J. Schutle, and D. Schultz, "Minerva: A Second Generation Mobile Tour-Guide Robot," IEEE International Conference on Robotics and Automation (ICRA-99), June 1999, Antwerp, Belgium.

[17] `http://www.haptica.com/walker.html`, Guido, Haptica Corporation.

[18] `http://www.opencyc.org`, The OpenCyc Project: Formalized Common Knowledge, Cycorp, Inc.