Workshop

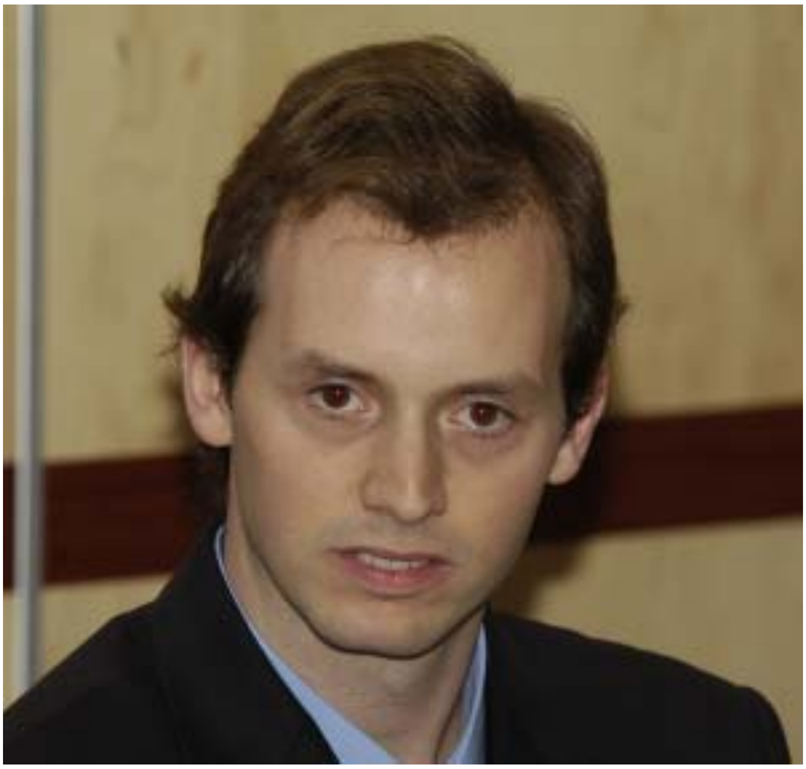
"ROBOTICS AND REMOTE-ASSISTANCE"

Guido, the Robotic SmartWalker for the frail visually impaired

Diego Rodríguez-Losada, Fernando Matía, Agustín Jiménez, Ramón Galán and Gerard Lacey

[1]Universidad Politécnica de Madrid. ETSII. DISAM y

[2]Haptica Ltd. Trinity College Enterprise Center



Guido, the Robotic SmartWalker for the frail visually impaired

Diego Rodríguez-Losada[1], Fernando Matía[1], Agustín Jiménez[1], Ramón Galán[1] and Gerard Lacey[2]

[1]Universidad Politécnica de Madrid. ETSII. DISAM y

[2]Haptica Ltd. Trinity College Enterprise Center

Abstract

Guido is a healthcare robot that serves as support and navigation aid for the frail and visually impaired. Many robotic technologies are implemented in Guido to let it achieve its task: Simultaneous Localization and Map Building (SLAM), pose tracking, path planning, collision avoidance and human robot interaction. All these components of Guido's navigation system are described in this work, as well as recent experiments both in the lab and in a hospital with real end users, showing the ease of use and convenience of the system, and the usefulness of the software tools that enable the system to be installed in a new environment in just minutes.

Index Terms

SLAM, healthcare, service robot, assistive aid.

1. Introduction

The aim of Guido™ the smart walker of Haptica Ltd. [1] is to develop a mobility aid for people who are both frail and visually impaired. 75% of the visually impaired are elderly and they often have related mobility problems that limit severely their independence. Guido aims to provide a mobility aid with makes taking independent exercise a safer and more enjoyable experience.

Guido (Figure 1) is a 4 wheel robust walking frame or rollator provided with two motors to steer the front wheels, but is has to be pushed by the user. It has a SICK laser to sense the environment, and a force sensor in the handle to sense the steer command. The on board processor is a PC104 300Mhz Geode system with 32Mb of flash memory (hard disk) and 32Mb of RAM, running

Haptica-TinyDCLinux. One of its serials port is used for the laser, and the other one connect to the Haptica H8 board, equipped with a Hitachi motor controller, motor power driver, digital inputs-outputs and sonar boards. Voice recorded messages are played by the Geode DSP device. An Ethernet port is available for communicating with the Geode system. Four batteries provide 24V power supply to the robot.

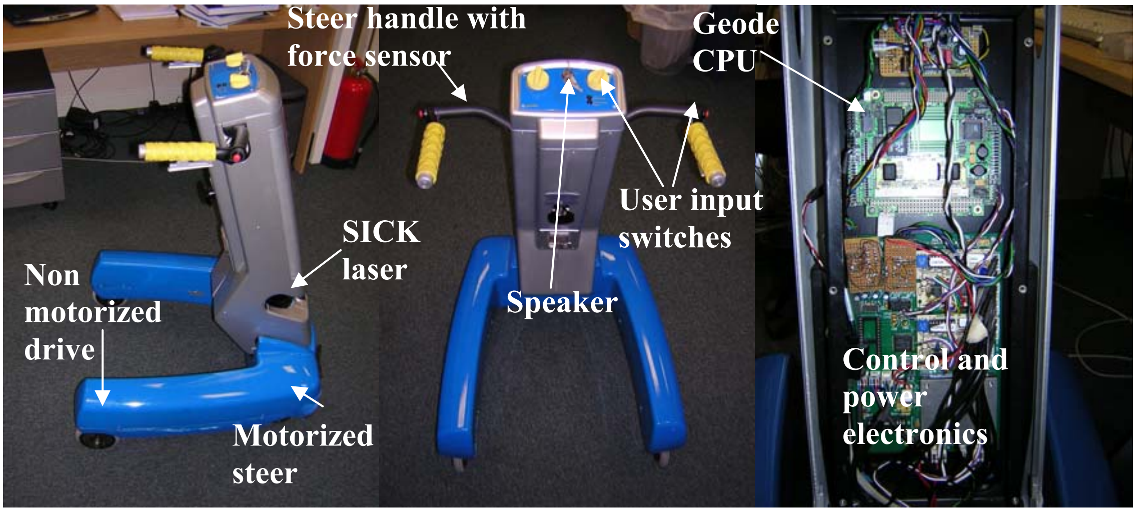


Figure 1. Guido, the SmartWalker

A former version of the control software was reactive behavior based, using the CleanSweep [2] control, part of Haptica's Interaction Engine™. This control was able to safely avoid obstacles, but the user had the control with the steering handle. Environment features are detected and reported to the user as “Opening left”, “T-junction”, etc. for assistance.

This control software has been recently replaced by a map based navigations system [3], developed by the authors for other indoor service robots [4],[5], as Blacky and Urbano (Fig 2), the interactive tour guide mobile robots developed in DISAM.

Figure 2. Urbano (left) and Blacky (right) tour guide robots.

A robot with map based navigation control uses an internal representation of the environment (map) to achieve its tasks. Although supplying an a priori man-made map is sometimes possible, being able to automatically build a map of the environment with a mobile robot possesses unquestionable advantages for many applications. This problem is known as Simultaneous Localization and Mapping (SLAM): the robot must incrementally build a map of the environment while localizing itself in that map. This chicken and egg problem focuses major attention in the mobile robotics community (see [6] for a survey). Our solutions to this problem are explained in Section 2.

Once the map is known, the robot only has to maintain (track) its pose (position and orientation) in that map. This is known as the continuous localization or poses tracking, which is much simpler than the SLAM problem. To reach a certain place in the environment, the robot has to execute a path planning algorithm, and then to execute the resulting trajectory. But it also has to account for dynamic obstacles (people) and other objects not represented in the map. The collision avoidance module implementes this functionality. All these subjects are covered in Section 3.

Section 4 briefly describes the software implementation details. Section 5 explains the experiments and field trials and presents the achieved results. Our conclusions are summarized in Section 6.

As explained above, the navigation system of Guido is composed by several subsystems. The detailed mathematical formulation involved in the development of each one is out of the scope of this work, so this paper will present an overview of the system, the concepts and tools involved, and its functionality, while avoiding formulae. The interested reader will be referred to the bibliography for further information.

2. Map building

The probabilistic approach has dominated the solution to the SLAM problem, which is considered the key for developing a truly autonomous mobile robot. Many different stochastic mapping approaches have been developed in the literature [6], from occupancy grid mapping to the recent MonteCarlo particle filter based FastSLAM algorithm. Expectancy Maximization, hybrid approaches and even topological SLAM are also other successful solutions to the problem.

2.1. Map building with EKF

Since the seminal paper [7], SLAM with an Extended Kalman Filter (EKF) [8], has probably been the most extended approach to stochastic mapping. Our first solution to SLAM uses an EKF for estimating a feature based map, composed by the walls of the environment. A formulation based on the SPMap [9] has been used for dealing with the problem of partiality in uncertainty presented by the symmetries of the geometric entities used in the map representation: bounded and unbounded segments.

It has been observed that the edges of the walls are very easy to detect and an extremely useful navigation reference, for example in large corridors. The naïve approach represents these limits as independent (stochastic) points, causing information redundancy with an increase in the computational cost. Integrating the edge information into the segment representation, as our formulation proposes, is a more efficient approach.

It is widely assumed that the EKF linear assumption is correct, as in the consistency and convergence properties proofs of the SLAM-EKF algorithm of [10] which are known to be the strongest proofs of convergence in the SLAM domain. Despite these interesting theoretical properties there has been an important lack of successful implementations for large indoor environments. It is a known problem that linearizations of the EKF can lead to divergence of the filter [11], but only [12], [13] have pointed out this effect on the SLAM domain. Furthermore, the experiments of [12] have been thoroughly extended [14] to analyze the applicability of SLAM-EKF for indoor environments concluding that the high accuracy of the SICK sensor and the very low measurement noise produces a very high inconsistency as previously stated for the generic EKF [11]. To deal with this problem we propose the use of perfectly known shape constraints between map segments. Parallelism, orthogonality, and colinearity are applied while the robot is exploring unknown parts of the environment. This algorithm has been implemented (Section 2.3), and tested in several environments. The full description of the algorithm can be found in [14].

One drawback of this approach is its computational complexity $O(n^2)$. Nevertheless, with an external laptop connected to Guido by the Ethernet Port, small to medium size environments can be mapped in real time, which is perfect for this application. The map of Haptica's office (Figure 3) was built in real time while Guido was manually controlled around the environment, with an exploration of 146 meters in 6 minutes. The map is composed by 87 features (segments). Offline map building requires only 37 seconds.

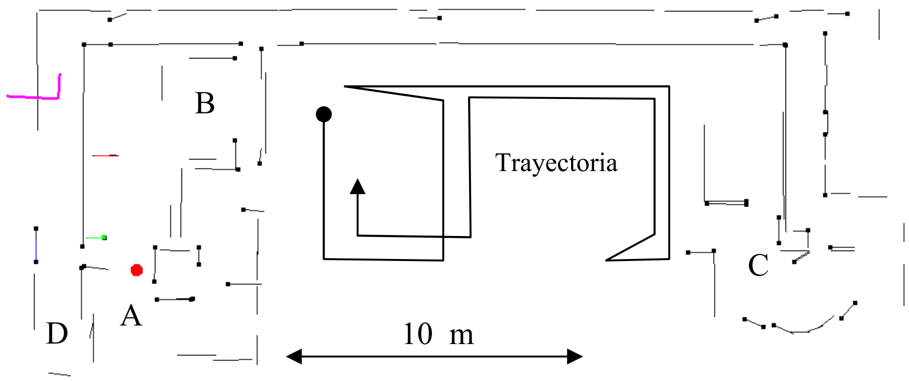


Figure 3. Haptica feature based map.

The map of Loyola House in St. Mary´s Hospital for the Blind (Figure 4) was also built in real time when the robot was installed for field trials purposes (see Section 4), with an exploration of 170 meters in 3 minutes and 30 seconds. The map has 150 features and a loop of approximately 100 meters was successfully closed. Offline map building can be done in 62 seconds.

2.2. Map building with particle filters

Although the performance of that software is fine for small-medium size environments, its complexity is still quadratic in the number of features. Furthermore, it relies on strong assumptions on the orthogonal shape of the environment to apply shape constraints that minimize the linearization errors that produce inconsistency in the estimation. While this approach works fine in such orthogonal environments it will most likely fail in other less structured environments.

Recently [15] introduced Rao-Blackwellized particle filters as an efficient solution to the SLAM problem. This approach has been followed by several authors, using a feature based approach [16] with the name of FastSLAM (Factored Solution to SLAM) and also with grid mapping representations [17].

The former allows a very compact representation of the map, but relies on extractors of predefined features. The latter can easily represent any object of the environment but requires large amounts of memory. The main drawback of the particle filter approach is its computational complexity measured in the number of required particles to correctly build a map. The approach [17] tries to reduce that number with more accurate sampling distributions.

The final goal of our research is to have a robust inexpensive SLAM algorithm that can run onboard in real time. We have developed a novel factorization of the SLAM problem [18] that has interesting potential advantages. This factorization has been implemented for application to indoor environments, letting us to, up to our knowledge, successfully build for the first time indoor feature based maps with particle filters.

With this novel formulation, the loop in St. Mary´s map is successfully closed 100% of times (25 runs) using just 30 particles, despite the large odometric error represented in (Fig 4). Both the topological correctness and the orthogonal shape of the environment can be appreciated, despite any shape constraints has been used.

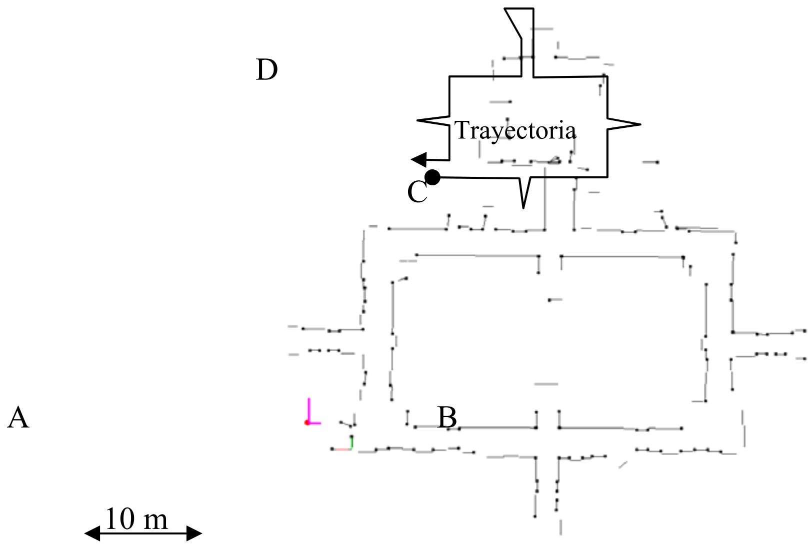


Figure 4. St. Mary´s Hospital Map built with EKF (left) and with a particle filter (right).

2.3. Map building and editing tool

The map building algorithms have been fully implemented in an independent portable C++ kernel. A GUI for Windows has also (Figure 5) been developed, using OpenGL for 3D rendering of the multiple views.

Once the geometric map is built, the installer of the system has to define the places of the environment that can be selected when using the robot with the command switches. Also, the graph of the environment that connects the goals has to be defined. This graph is composed by nodes connected by straight segments. Each node can have a different size. Both tasks can be easily done using the graphical editor integrated in the map building application with a few mouse clicks. A toolbar allows multiple operations that can be executed clicking on the 3D views: deleting, adding and correcting walls, adding, removing and changing nodes of the graph, adding edges, etc.

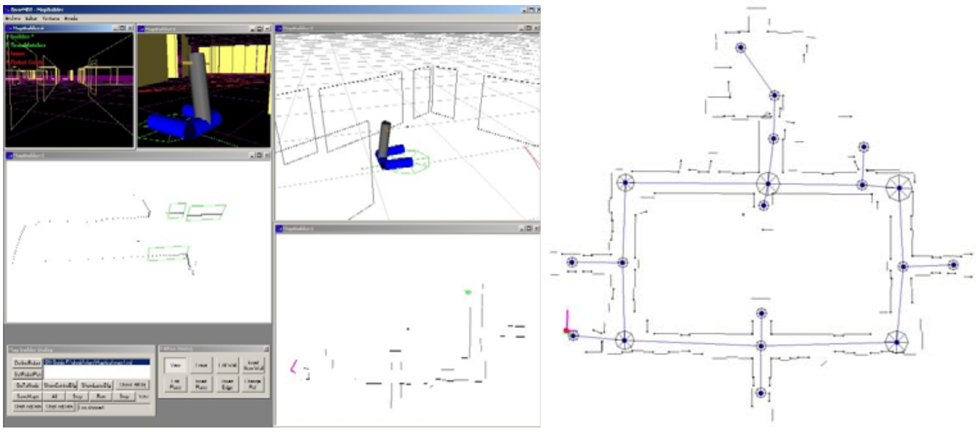


Figure 5. Map building and editing tool (left). St. Mary´s environment graph (right)

3. Navigation

Once the map is built, it is saved in Guido onboard CPU, and it is automatically loaded when Guido starts its normal operation, and used to guide the user to the desired goal. Guido has 3 modes of operation, selected by a control switch: Manual, Program and Automatic mode. In Manual mode, the user is able to steer the robot, using only the localization module. This mode is intended for assistants only. The Program mode allows selecting the desired destination (navigating through the known places with Guido switches), while the robot brakes are applied for safety. When the goal is selected and control switch is changed to Automatic mode, the path to the goal is computed, the brakes are released, and Guido starts to steer to drive the user to the goal in full Automatic mode.

3.1. Localization

Once the map is downloaded to the robot, it is able to perform a robust and accurate continuous localization or pose tracking, using a simplification of the map building algorithm. The new state for the EKF is the robot pose, and all the features of the map are supposed to be known without uncertainty.

The edge information is used in the localization, as well as the euclidean prefilter, multiple matches are allowed and the JCBB test is performed.

The robot cannot be "kidnapped" (by lifting and moving it) in this version, as global localization is not performed.

3.2. Path planning and execution

The first step in the path planning is applying an A star algorithm in the predefined graph of the environment. Thanks to this graph, this can be done in a very short time (<0.5 sec) despite the few computational resources. A full path-maneuver planner using the grid map could take several seconds in Guido processor, which is highly undesirable for the users. When the first step is done, the set of straight segments of the resulting trajectory is interpolated by circle arcs in their intersections, whose radius are defined by the size of the node common to each pair of segments. The remaining large straight segments are splitted in short ones to apply the corrections required to avoid dynamic and non mapped objects.

Once the trajectory is computed and before executing that trajectory, a collision free trajectory is computed. A local point map is maintained at the laser frequency, which represents dynamic and not represented objects of the segment feature based map.

The distance of each local map point to every path point is computed. If a minimum distance criterion is not met, then the path point is accordingly displaced along the path tangent direction.

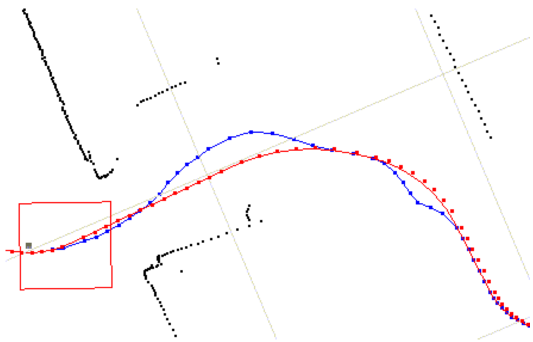


Figure 6. Local point map and trajectory correction for collision avoidance.

Guido is a non holonomic platform, and the control is designed for forward direction. A lateral displacement proportional controller has been implemented to follow the collision free path. The shortest distance from the robot rear midpoint and back midpoint to the dynamic trajectory is computed. The reference steer angle is the tangent of the trajectory in the closest point to the robot rear midpoint. Although this is only a rough approximation to the Ackerman geometry, it has proven to work for the trajectory following if the turn radius is large enough. If the turn radius is small then the back of the robot deviates from the trajectory and the likely of a collision increases. The final position tolerance is defined by the end node size.

3.3. Voice messages.

Prerecorded voice messages are used to assist the user in Guido control. Under Program mode, the messages inform the user which is the selected goal. In Automatic mode it tells the user about different navigation events: path blocked, path clear, turn on the spot, walk straight ahead, arrived to “place name”, passing by “place name”, etc. The messages that identify the names of the relevant places have to be recorded when the system is installed in a new environment, in the map editing step.

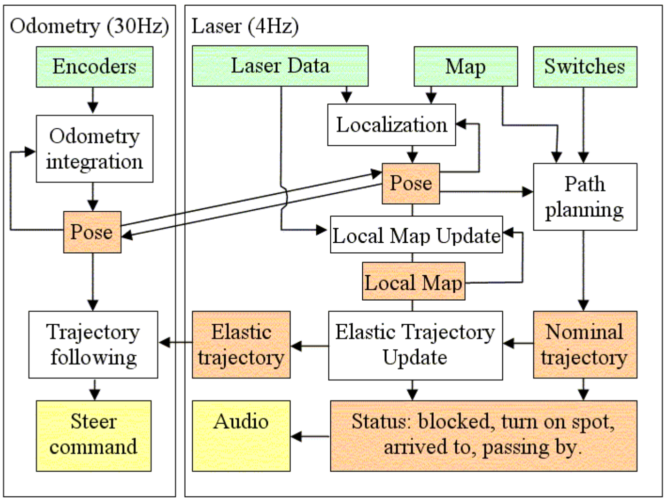


Figure 7. Software implementation

3.4. Implementation

In the map based guidance mode, all the software has to run in the onboard processor for full autonomy (Figure 7). The bottleneck of the processing was the laser data acquisition via the serial port, which is as low as 4Hz. This low rate and the poor quality of steer motor controllers produce a poor trajectory control that can be avoided by increasing the control rate. This is done adding a new thread for integrating the odometry and performing the trajectory control. The remaining computations are done in the laser thread at a lower frequency. Careful information sharing (especially the robot pose) had to be implemented for this purpose.

4. Experiments

The system has been installed and successfully tested in two different environments: Haptica's lab and St Mary's Home for visually impaired women (both located in Dublin, Ireland). While the former served for testing, debugging, and performing extensive experimentation, the latter served to prove the adequateness of the installation tools as well as to get the feedback of real end users (frail and visually impaired).

4.1. Haptica Lab

In this environment a comparison experiment between the assistive control (previous) and the map based control (new) was done. Seven (artificially blind) users had to accomplish the task of going from place A (battery charger) to place B (meeting room), then to place C (management office) and finishing in place D (building exit) (See Figure 3).

It is shown how the required time to complete the task can be reduced by 50% (Figure 8). The office was cluttered (Figure 9), so maneuvering Guido with the assistive control was quite complicated, while it was absolutely straightforward with the map based control. The number of collisions (frontal bumps and lateral scratches) is drastically reduced, avoiding all frontal bumps (that happens with the assistive control) that require posterior maneuvering. The collisions when using the map based control are reduced to a very few and slight scratches, that occurred when crossing a narrow doorway at high speed because the path regulator was tuned for lower speed. It can be seen that the force applied to the handle is also reduced more than 50%. This means that the necessary energy in the task is reduced to 25%, taking into account the reduction in time. All the users definitely preferred the map based navigation.

4.2. St. Mary´s Home for visually impaired

Four places were defined in this environment: place A (the sun room), place B (the smoking room), place C (the dining room) and place D (the main entrance). Two women, the first one aged 83, partially sighted (light and dark shapes only) and the second one aged 86, totally blind, tested Guido. They were very delighted that Guido could drive them to their goals, instead of having to steer it. Several tasks were successfully accomplished, moving between the four predefined places. They were extremely interested in having Guido drive them to their rooms, which couldn't be done only because these goals were not defined in the map. Nevertheless, this showed that the map based navigation is an added value for these real end users.

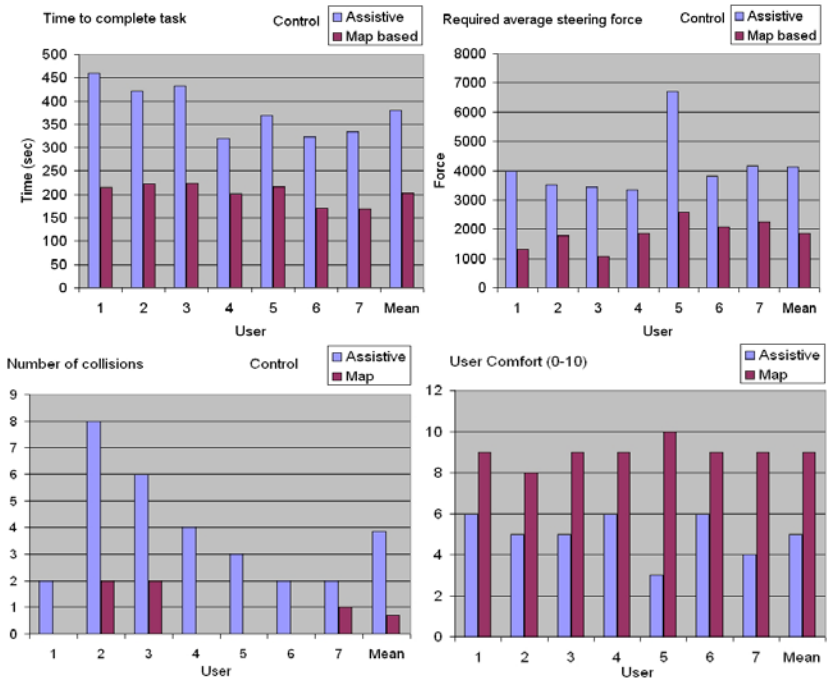


Figure 8. Results: Comparision charts

Figure 9. Field trials. St. Mary's Home (left), Haptica (right)

5. Conclusions and further work

This work has presented the recently developed map based navigation system for Guido, an assitance walking aid for the frail visually impaired. A description of all components of this system has been done, as well as the experiments that showed a great improvement in Guido performance and added value compared to the previous navigation software.

We are currently trying to improve the system by adding more automatic functionality to Guido:

- One of our goals is to develop a robust and computationally inexpensive algorithm that could be integrated onboard Guido, to avoid the use of any external equipment in the installation phase. Our novel particle filter formulation could be a promising one, and we will continue working in SLAM algorithms. Altough running on an external laptop, the developed software tools have been proved to be extremely useful, letting us to install Guido in new environments in just few minutes.
- The addition of a predefined graph of the environment could be avoided if a grid map is used for that purpose, combined with an automatically precomputed voronoi graph that allows a fast path planning. The navigation software has to be more extensively tested and debugged for robustness. More field trials by end users has to be done.
- Voice messages have to be prerecorded. This task could be automated by off the shelf text to speech software. The system could also be highly improved with some voice recognition capabilities. In any case, a new hardware version of Guido robot with a reconfiguration of the user input for the new functioning mode should be done.

Acknowledgment

Diego Rodriguez-Losada thanks Haptica Ltd. for hosting his research visit and providing all the necessary material and human means for this work.

References

[1] MacNamara, S.; Lacey, G.; “A smart walker for the frail visually impaired”, ICRA'00. IEEE International Conference on Robotics and Automation, Vol: 2 , 2000. Pages:1354 – 1359

[2] Cassidy. D. “Improved obstacle avoidance for a human-machine system”, Master of Science Thesis, Trinity College Dublin. September, 2002

[3] Rodriguez-Losada, D.; Matia, F. Jimenez, A. R.Galan, G.Lacey. “Implementing Map Based Navigation in Guido, the Robotic SmartWalker” IEEE International Conference on Robotics and

Automation, 2004. ICRA '05, Barcelona, Spain. Proceedings. Page(s): 3401-3406.

[4] Rodriguez-Losada, D.; Matia, F.; Galan, R.; Jimenez, A. “Blacky, an interactive mobile robot at a trade fair” IEEE International Conference on Robotics and Automation, 2002. ICRA '02, Washington DC., USA. Proceedings. Volume: 4 , 2002 . Page(s): 3930 -3935 vol.4

[5] Matia, F.; Rodriguez-Losada, D.; Galan, R.; Jimenez, A. “Experiments at trade Fairs with Blacky the Robot”. IEEE International Conference on Intelligent Robots and Systems, 2002. IROS '02. Laussane, Switzerland. Workshop WS9 "Robots in Exhibitions" Proceedings. Page(s):61-66

[6] S. Thrun. “Robotic mapping: A survey” In G. Lakemeyer and B. Nebel, editors, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002

[7] R.Smith, M. Self, P. Cheeseman “Estimating uncertain spatial relationships in robotics” in Uncertainty in Artificial Intelligence 2, J.F. Lemmer and L. N. Kanal, Eds. New York: Elsevier, 88

[8] Kalman R.E., 1960. “A new approach to linear filtering and prediction problems”. Trans. ASME, Journal of Basic Engineering. pp. 34-45.

[9] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardos “The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building” in IEEE Trans. on Robotics and Automation, vol. 15, n5 Oct 99.

[10] Dissanayake G., Newman P., Clark S., Durrant-Whyte H., Csorba M., 2001. “A solution to the simultaneous localisation and map building (SLAM) problem”. IEEE Transactions on Robotics and Automation. Vol 17. n. 3 pp. 229-241

[11] Bar-Shalom Y., Li X.R. Kirubarajan T., 2001. “Estimation with Applications to Trackking and Navigation”. Wiley InterScience.

[12] Julier S., Uhlmann J.K., 2001. “A counter example to the theory of simultaneous localization and map building”. IEEE International Conference on Robotics and Automation. Vol. 4. Seul, Corea. pp. 4238-4243.

[13] Castellanos J.A., Neira J., Tardos J.D., 2004. "Limits to the consistency of EKF-based SLAM". Fifth IFAC Symposium on Intelligent Autonomous Vehicles IAV'04. Lisbon, Portugal.

[14] Rodriguez-Losada, D. "SLAM geometrico en tiempo real para robots mobiles en interiores basado en EKF (Geometric Real Time EKF based SLAM for indoor mobile robots)". PhD Thesis. Polytechnical University of Madrid. Spain. May 2004.

[15] A. Doucet, J.F.G. de Freitas, K. Murphy, S. Russel. Raoblackwellized particle filtering for dynamic bayesian networks. Conf. on Uncertainty in Artifcial Intelligence (UAI), 2000.

[16] Montemerlo M., Thrun S., Koller D., Wegbreit B., 2002. "FastSLAM: A factored solution to the simultaneous localization and mapping problem". AAAI Nat.Conf. on Artif. Intelligence. Edmonton, Canada.

[17] G. Grisetti, C. Stachniss, and W. Burgard Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In Proc. of the IEEE ICRA'05 pp 2443-2448, Barcelona, Spain, 2005

[18] Rodriguez-Losada, D.; Matia, F. Jimenez, A. R.Galan, G.Lacey. "Dual of the Factored Solution to the Simultaneous Localization and Mapping Problem" Submitted to IEEE International Conference on Robotics and Automation 2006.