The Impact of Technology on Orientation Aid for the Visually Impaired

Ines Bula*, Edin Bula, Muzafer Shala

Faculty of Mechatronics, University for Business and Technology, Lagjia Kalabria p. n., 10000 Prishtinë, Kosovo;
**bulaines@hotmail.com*

Simulation Notes Europe SNE 24(1), 2014, 33 - 38
DOI: 10.11128/sne.24.tn.10229
Received: Nov. 20, 2013 (Selected ICBTI 2013 Postconf. Publ.);
Accepted: January 15, 2014;

Abstract. Statistic states that 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. About 90% of the world's visually impaired people live in developing countries.

Taking in consideration that Mechatronics is a methodology used for the optimal design of electromechanical products, and by combining technologies that are available to us we can develop a very useful tool that blind people and people with sight problems can change their lives.

Combining smart phones and digital camera there are possibilities to build smart glasses which will give information to blind people.

In this paper definitely a new ap-proach for making peoples life easy is proposed. Initially the re-sults are reached from simulation using Matlab/SIMULINK package which will lead this research to real time experimental results.

Introduction

By taking into consideration that Mechatronics is a branch of engineering that combines different disciplines of engineering from Computer Science, Mechanical engineering, Electrical engineering, Electronics, to natural sciences such as Physics and Applied Mathematics, in order to solve a particular problem at hand. This proposed product is no exception for we took pre existing technologies and modified it to our benefit. Worldwide statistic states that 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. About 90% of the worlds visually impaired live in developing countries.

We saw a problem and searched for possible solutions and we came up that by combining smart phones, digital cameras, GPS, 3G or 4G telephone network (that supports internet) and of course some cutting edge programming the goal of developing a tool that is able to make life's of millions a little bit better is very much achievable.

1 Reserch with Autonomus Robots (Robo Earth)

If we take in consideration that The majority of the world's 8 million service robots are toys or drive in preprogrammed patterns to clean floors or mow lawns, while most of the 1 million industrial robots repetitively perform preprogrammed behaviors to weld cars, spray paint parts, and pack cartons [2]. To date, the vast majority of academic and industrial efforts have tackled these challenges by focusing on increasing the performance and functionality of isolated robot systems. However, in a trend mirroring the developments of the personal computing (PC) industry [3], recent years have seen first successful examples of augmenting the computational power of individual robot systems with the shared memory of multiple robots. In an industrial context, Kiva Systems successfully uses systematic knowledge sharing among 1,000 individual robots to create a shared world model that allows autonomous navigation and rapid deployment in semi structured environments with high reliability despite economic constraints [4], [5]. Other examples for shared world models include research on multi agent systems, such as RoboCup [6], where sharing sensor information has been shown to increase the success rate of tracking dynamic objects [7], collective mapping of autonomous vehicles [8], [9], or distributed sensing using heterogeneous robots [10].

However, in most cases, robots rely on data collected once in a first, separate step. Such pooled data have allowed the development of efficient algorithms for robots, which can then be used offline without access to the original data. Today's most advanced personal assistant robots rely on such algorithms for object recognition and pose estimation [11], [12]. Similarly, large training data sets for images and object models have been crucial for algorithmic advances in object recognition [13]–[14].

The architecture and implementation of RoboEarth is guided by a number of design principles, centered on the idea of allowing robots to reuse and expand each other's knowledge. To facilitate reuse of data, RoboEarth supports and leverages existing standards. The database is made available via standard Internet protocols and is based on open source cloud architecture to allow others to set up their own instance of RoboEarth, resulting in a truly distributed network. The code generated by the RoboEarth Consortium will be released under an open-source license, and will provide well documented, standardized interfaces. Finally, RoboEarth stores semantic information encoded in the World Wide Web Consortium (W3C) - standardized Web Ontology Language (OWL [17]) using typed links and uniform resource identifiers (URIs) based on the principles of linked data [15].

2 Architecture of Robo Earth

RoboEarth is implemented based on a three-layered architecture (Figure 1). The core of this architecture is a server layer that holds the RoboEarth database [Figure 1(a), the "Architecture: Database" section]. It stores a global world model, including reusable information on objects (e.g., images, point clouds, and models), environments (e.g., maps and object locations), and actions (e.g., action recipes and skills) linked to semantic information (e.g., properties and classes), and provides basic reasoning Web services. The database and database services are accessible via common Web interfaces.

As part of its proof of concept, the RoboEarth Consortium [16] is also implementing a generic, hardware-independent middle layer [Figure 1(b)] that provides various functionalities and communicates with robot-specific skills [Figure 1(c)]. The second layer implements generic components. These components are part of a robot's local control software. Their main purpose is to allow a robot to interpret RoboEarth's action recipes. Additional components enhance and extend the robot's sensing, reasoning, modeling and learning capabilities and contribute to a full proof of concept that closes the loop from robot to the World Wide Web database to robot.

The third layer implements skills and provides a generic Interface to a robot's specific, hardware-dependent functionalities via a skill abstraction layer.

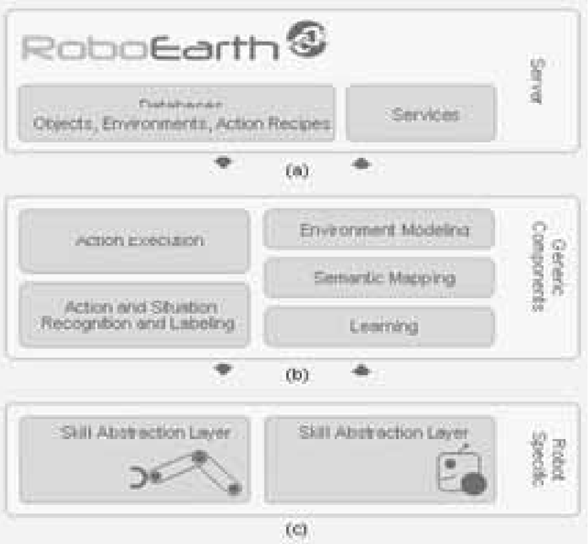

Figure1: Robo Earth's three layered architecture. [1]

3 Data Base of Robo Earth

RoboEarth stores CAD models, point clouds, and image data for objects. Maps are saved as compressed archives, containing map images and additional context information such as coordinate systems. Robot task descriptions are stored as human readable action recipes using a high level language to allow sharing and reuse across different hardware platforms. Such action recipes are composed of semantic representations of skills that describe the specific functionalities needed to execute them. For a particular robot to be able to use an action recipe, the contained skills need to have a hardware-specific implementation on the robot. To reduce redundancy, action recipes are arranged in a hierarchy, so that a task described by one recipe can be part of another more complex recipe. In addition, database services provide basic learning and reasoning capabilities, such as helping robots to map the high-level descriptions of action recipes to their skills or determine what data can be safely reused on what type of robot.

The RoboEarth database has three main components (Figure 2).

First, a distributed database contains all data organized in hierarchical tables [Figure 2(a)]. Complex semantic relations between data are stored in a separate graph database [Figure 2(b)]. Incoming syntactic queries are directly passed to the distributed database for processing. Semantic queries are first processed by a reasoning server. Data are stored in a distributed database based on Apache Hadoop [20], which organizes data in hierarchical tables and allows efficient, scalable, and reliable handling of large amounts of data.

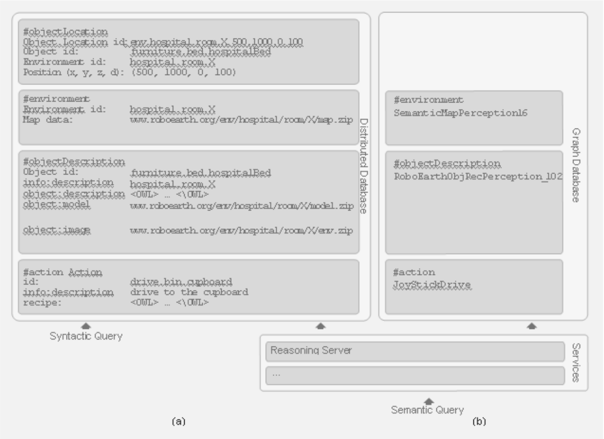

Figure2: The three main components of the RoboEarth database. [1]

Second, a centralized graph database holds semantic information encoded in the W3C-standardized OWL [17]. It stores the following data and their relations.

3.1 Objects

The database stores information on object types, dimensions, states, and other properties as well as locations of specific objects a robot has detected and object models that can be used for recognition (Figure 3). Figure 3(a) describes a recognition model for a certain kind of object (defined by the property providesModelFor), giving additional information about the kind of model and the algorithm used. The actual model is linked as a binary file in the format preferred by the respective algorithm (defined by the property `linkToRecognitionModel`). Figure 3(b) describes the recognition of a specific object. An instance of a `RoboEarthObjRec`- Perception is created, which describes that the object `Bottle2342` (linked through the property `objectActedOn`) was detected at a certain position (linked through the property `eventOccursAt`) at a given point in time using that recognition model (defined by the property `recognizedUsingModel`).

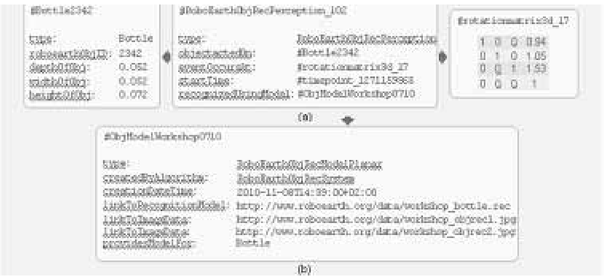

Figure3: The object description, recognition model. [1]

3.2 Environments

The database stores maps for self-localization as well as poses of objects such as pieces of furniture (Figure 4).

The semantic map combines a binary map that is linked using the linkToMapFile property with an object that was recognized in the respective environment. The representation of the object is identical to the one in Figure 3. This example shows that both binary (e.g., occupancy grids) and semantic maps consisting of a set of objects can be exchanged and even combined. The given perception instance not only defines the pose of the object but also gives a time stamp when the object was seen last. This can serve as a base for calculating the position uncertainty, which increases over time.

Figure 4: The environment map used in the second demonstrator. [1]

3.3 Action recipes

The stored information includes the list of subaction recipes, skills, and their ordering constraints required for executing an action recipe as well as action parameters, such as objects, locations, and grasp types (Figure 5). Action classes are visualized as blocks, proper ties of these classes are listed inside of the block, and ordering constraints are depicted by arrows between the blocks. The recipe is modeled as a sequence of actions, which can be action recipes by themselves, e.g., the GraspBottle recipe. Each recipe is a parameterized type specific subclass of an action such as Translation. Atomic actions, i.e., actions that are not composed from sub actions, represent skills that translate these commands into motions.

A first type of service is illustrated by RoboEarth's reasoning server. It is based on KnowRob [18] and uses semantic information stored in the database to perform logical inference. Services may also solely operate on the database.

RoboEarth's learning and reasoning service uses reasoning techniques [19], [18] to analyze the knowledge saved in the RoboEarth database and automatically generates new action recipes and updates prior information. For example, given multiple task executions, the database can compute probabilities for finding a bottle on top of the cupboard or on the patient's nightstand. Using the additional information that cups are likely to be found next to bottles, the service can automatically create a hypothesis for the probability of finding cups on top of the cupboard.

Figure 5: The action recipe used for the second demonstrator. [1]

Such cross correlations between objects can provide powerful priors for object recognition and help to guide a robot's actions. Additionally, if there are two action recipes that reach the same goal in different ways, the learning and reasoning service can detect this, fuse the recipes, and explicitly represent both alternatives.

For example, if robot A was equipped with a dexterous manipulator but robot B only with a tray, the component could create a single action recipe 'serve drink to patient' with two branches depending on the robot's abilities, which would have different requirements: the first branch would require a graspable bottle, whereas the second branch would require the availability of human or robotic help to place the bottle on the tray.

4 Proposal and Conclusion

By utilizing the avaible technologies and modifying them to our needs is mechtronics in action. It may interes you why we focust so much on robo earth.

The answer is that this kind of technology is at the very core of our product for it is easy to connect a digital camera to a smart phone and to use GPS and the internet. But it is al in vain if you don't have an image processor to analyse it and a data base to store and catologe it. All of that robo earth offers us.

The ability of the smart robots to recognize opsticles an independently avoid them in adition to their ability to move without outside aid is something we need for the realization of this project.

From the three layers only two may be needed because hardwere control layer may not be nessesary.

So if we take a smart phone that in it self has a GPS tracking system, significant processing power, storage and internet support. So if we connect a miniature digital camera with the smart phone. And that camera sends us life video from the environment to the phone. Because smart phones are like a small scale PC pre processing and cataloging can be done and then send via internet to the created cloud where the data is analysed, categorized and made available to all the users in this cloud.

So for example a visually impared person walks down the streat and and an unknown object is detected.

By means of a digital camera which records objects that are before us at a certain distance. The picture is then send sent to an application installed on the smart phone, this new application which is connected to a central data- base which one containing the list of possible objects that could be faced along the way, but with enlargement opportunities, that means if the user saw an unknown object that is not registered in the database, the object will be recorded saved, analyzed and registered in the database, and when we encounter it on the road again, it will be registered in the database and it will be available to users.

All objects that are registered in the database must be encoded in advance and each of them is given a code to identify them, therefore all new objects which we encounter and are not encoded, are recorded by the digital camera which then sends the picture to the application which then notifies the database for an object unknown, the user of the tool is then automatically notified even though we don’t know what kind of object are we talking about of an potential risk in certain distance and size of that object.

At first only object of a certain size, velocity and distance will be reported to the user in order to not confuse the user. But possible extentions of servise like face regognition, regognition of everyday items, reading text and so on.

This idea of combining technologies is perfect example of mechatronics in use. Even though the idea is very ambitious for a university level it is very much feasible but huge components especially the software and database is in copyright of few companies and the only alternative is a from the scratch approach who is quite difficult if we take into consideration the resources needed to build it. But of coarse if we use different simulation software’s a simulation of the product is very much possible. Initially the results are reached from simulation using Matlab/SIMULINK package which will lead this research to real time experimental results.

Next steps might be the development of similar laboratory equipments and development of software and database some of which may be accessible via an open source which can be modified to our needs and the rest of the software will be necessary to be programmed by ourselves.

Acknowledgement

This contribution is a post-conference publication from ICBTS 2013 Conference (Durres, Albania 1-2 November 2013), co-organized by KSIM, the Kosovo Simulation society (EUROSIM Member). The contribution is a modified publication from the paper published in the Proceedings of ICBTS 2013, UBT Publications, E. Hajrizi (ed.), ISBN: 978-9951-437-24-0, at website http://conferences.ubt-uni.net/2013/digital-proceedings/icbti2013-cse-mek-proceedings.pdf.

References

[1] http://www.roboearth.org/wp-content/uploads/2011/06/Waibel2011RoboEarth.pdf

[2] “World Robotics 2010,” Int. Federation Robot. Statist. Dept., 2010

[3] Gates B. A robot in every home. *Scientific Amer. Mag.* 2007; 296(1): 58–65

[4] Wurman P, D’Andrea R, Mountz M. Coordinating hundredsof cooperative, autonomous vehicles in warehouses. *AI Mag*. 2008; 29(1)

[5] Guizzo E. Three engineers, hundreds of robots, one warehouse. *IEEE Spectr*. 2008; 45(7): 26–34

[6] Buchheim T, Kindermann G, Lafrenz R. A dynamic environmentmodelling framework for selective attention. In: Proc. IJCAI Workshop: Issues in Designing Physical Agents for Dynamic Real-Time Environments: World Modeling, Planning, Learning, and Communicating. 2003; P.91–98.

[7] Roth M, Vail D, Veloso M. A real-time world model for multirobot teams with high-latency communication. In: Proc. IEEE/RSJ Int.Conf. Intelligent Robots and Systems 2003(IROS’03), vol. 3; P.2494–2499.

[8] Fenwick J, Newman P, Leonard J. Cooperative concurrent mapping and localization. In: *ICRA’02*. Proc. IEEE Int. Conf. Robotics and Automation 2002 . vol. 2; P. 1810–1817

[9] Chang H, Lee C, Hu Y, Lu Y, Multi-robot SLAM with topological/metric maps. In: *IROS’07*. Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems 2007. P. 1467–1472

[10] Parker L, Fregene K, Guo Y, Madhavan R, Distributed heterogeneous sensing for outdoor multi-robot localization, mapping, and path planning. In: Proc. 2002 NRL Workshop Multi-Robot Systems: From Swarms to Intelligent Automata, Washington, DC (Multi-Robot Systems: From Swarms to Intelligent Automata, A. C. Schultz and L. E. Parker, Eds.) Norwell, MA: Kluwer, Mar. 2002, P. 21.

[11] Grundmann T,Eidenberger R, Schneider M, Fiegert M, v. Wichert G. Robust high precision 6D pose determination in complex environments for robotic manipulation. In: Proc. Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation at the Int. Conf. Robotics and Automation, 2010, P. 1–6.

[12] Rusu RB, Bradski G, Thibaux R, Hsu J. Fast 3D recognition and pose using the viewpoint feature histogram. In: Proc. Int. Conf. Intelligent Robots and Systems, 2010, P. 2155–2162

[13] Everingham M, Zisserman A, Williams CKI, Van Gool L, et al. The 2005 PASCAL visual object classes challenge.In: Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment (LNAI 3944), J. QuinoneroCandela, I. Dagan, B. Magnini, and F. d’Alche-Buc, eds. Berlin: SpringerVerlag, 2006, pp. 117–176.

[14] Merler M, Galleguillos C, Belongie S, Recognizing groceries in situ using in vitro training data. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[15] Bizer C, Heath T, Berners-Lee T. Linked data—the story so far. *Int. J. Semantic Web Inform. Syst.* 2009; 5(3): 1–22

[16] v. d. Molengraft R, et al. (2010). *The RoboEarth Consortium*[Online]. Available: http://www.roboearth.org

[17] W3C OWL Working Group. (2009 Oct.). *OWL 2 web ontology Language document overview, W3C Recommendation* [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

[18] Tenorth M, Beetz M. KnowRob—Knowledge processing for autonomous personal robots. In: *IROS 2009*. Proc. IEEE/RSJ Int. Conf. IntelligentRobots and Systems 2009: P. 4261–4266

[19] Geusebroek J, Burghouts G, Smeulders A. The Amsterdam libraryof object images. *Int. J. Comput. Vis.*. 2005; 61(1): 103–112

[20] Murthy AC, et al. Apache Hadoop, a framework for reliable, scalable and distributed computing [Online]. Available: http:// hadoop.apache.org

[21] Daniels P. Analog and Digital Writing. In: *The World's Writing Systems*. 1996. 886p.