AUT @Home 2016 Team Description Paper

Esmaeil Mehrabi, Edwin Babaians, Alireza Ahmadi, Ameneh Sheikhjafari, Shaghayegh Gharghabi, Navid Khazaee Korghond, and Saeed Shiry Ghidary

Robotics Research Center, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
http://sar.aut.ac.ir/en/?p=96

Abstract. This paper describes the AUT@Home robot (*Sepanta*) of Amirkabir University of Technology, Tehran, Iran, for the participation at the RoboCup@Home 2016 in Leipzig, Germany. We describe the hardware characteristics and capabilities of Sepanta to perform tasks which are related to human daily life. In addition it covers our approaches for Natural Intelligence (NI) tasks such as speech and object recognition which has made our robot reliable for a wide range of tasks in home environments.

1 Introduction

M.Sc. and B.Sc. students from various departments with different backgrounds participate in AUT@Home team. Our team members are previously involved in a variety of projects which cover various aspects of *"At home"* robots and *"Social Robots"*, in general. These projects are advised by the faculty members of Amirkabir University of Technology. *Sepanta III* is our robot and you can find evolutionary stages of its history in the following section. It has many improvements such as new arms, better object recognition method, SMACH system for complex behaviors[1], better navigation system and Kinect for Windows v2 sensor[2].

One of the most important points of our work is that AUT@Home robot, Sepanta, is equipped with a RGB-D Camera and laser range scanner for 3D perception. Sepanta uses the data from these sensors for navigation, real-time environment perception, object recognition, and manipulation. Besides, actuators of Sepanta are highly reliable which allow us to have a perfect control over the robot's manipulation and motion mechanism. Since Sepanta has two arms, it can carry out a wide range of manipulation tasks in the real world. Also, Sepanta has a holonomic navigation system which helps it to move better in complex situations such as passing through the door and manipulating objects.

The rest of paper is organized as follows; First, we have mentioned our backgrounds. Second, the electrical and mechanical design of the robot is introduced.

[1] SMACH is a task-level architecture for rapidly creating complex robot behavior, available at http://wiki.ros.org/smach

[2] Kinect for Windows v2 sensor from Microsoft Company, available at http://www.microsoft.com

Third, the robot software platform is described. Finally, our plans for future modifications on the robot are mentioned.

2 Background

Our team has been founded in December 2007, which was known as *"Sourena Team"*. The first participation in the international competitions was at RoboCup 2008. In that competition, we had a good demonstration and achieved the 5th place. Also, we participated in RoboCup competitions three years in a row. Through these years, we have collected invaluable experiences which made our team stronger for future participation. At 2013, we built a new robot. This robot, called Sepanta, participated in AUTCup 2013 and achieved the 1st place. Then, we redesigned it and called this new version Sepanta II. It achieved the 8th place in Robocup 2014, Brazil, and 2nd place in IranOpen 2014 competitions. Now, Sepanta III is the latest version of our robot that will participate in the next competition.

3 Robot Platform

3.1 Mechanical Design

Since @Home robots should be able to perform in a dynamic environment, we have designed and implemented a *"Holonomic"* motion mechanism for Sepanta. As a holonomic mechanism in a discoid workspace has three degrees of freedom (DOF), it can move Omni-directionally in the environment. Our holonomic mechanism consists of four Mecanum wheels.

In addition, based on the fact that the robot is assumed to accomplish human-like duties, we have developed two arms for manipulation tasks. Each Arm has 4 DOF and are coherently connected to the shoulder. Also, their control procedures are handled on a certain and common processor called Device Communication Manager (DCM) which has access to the robot's sensors and actuators. Moreover, we have tried to design a bio-inspired mechanism for Sepanta. In this regard, we have used a prismatic joint in Z axis to let the robot reach objects in all heights and situations.

3.2 Electrical Design

In this section we describe sensors and circuits which are used for data acquisition and filtering. Furthermore, the robot controller architecture is presented and discussed in the following text. In this robot, we used Robotis Open CM9 embedded board[1] as a low level controller and the DCM. Different instruments such as actuators and sensors communicate with the main processor by this

[1] CM-900 Series with STM32F103C8 based on 32Bit ARM Cortext-M3 and its circuit diagram and the entire sources are open to public, available at www.robotis.com

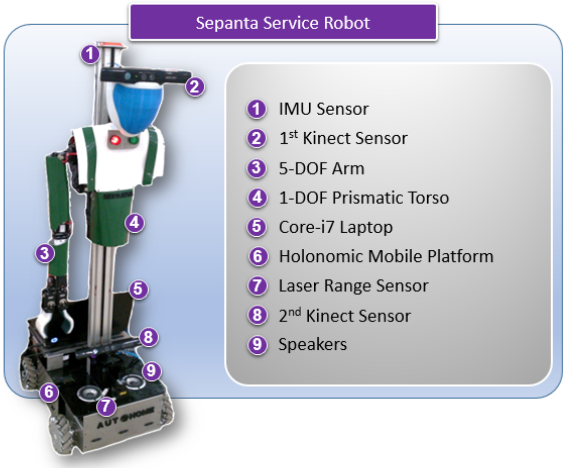

Fig. 1. Sepanta II equipment's description.

board. In low-level computations on this board, we drive the sensors such as infrared GP2D120 distance sensors, IMU data fusion and filtering, high level interfacing and data transfer, MD49[1] motor driver and wireless Robotis Controller pad interfacing. Also, we used two Intel core-i7 mini PCs instead of using laptops on the robot. Intel NUC and Gigabyte Brix mini PCs are preferred for their flexibility and reliability.

4 Sepanta's Software and Artificial intelligence

Processes which run on the main controller are known as our software parts and are discussed in this section. The robot's operating system architecture is based on ROS (Robot Operating System)[2] which runs on Ubuntu Linux. ROS is a semi-operating system, with a core which manages software resources and inter-process communications among ROS Nodes. These nodes are like processes in a real operating system and they can be easily connected to each other using ROS concepts. We are using Kinect for windows v2 partly for skeleton, speech and face recognition purposes. So, the skeleton, speech and face recognition modules are partly developed under Windows operating system. For transferring commands and data we used UDP protocol and LAN network infrastructure to create a bridge between our ROS nodes and Windows Processes.

We use the SMACH state machine system for our main stage challenges. Each process is managed from a main node which handles robot's status. SMACH is

[1] The EMG49 Motors and MD49 Drivers from Devantech Company, available at www.robot-electronics.co.uk

[2] The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications, available at www.ros.org

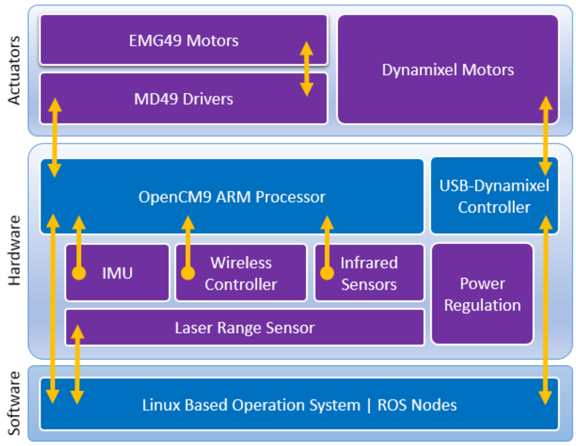

Fig. 2. High-level and low-level data fusion and interfacing.

a task-level architecture for rapidly creating complex robot behaviors. At its core, SMACH is a ROS-independent Python library to build hierarchical state machines.

4.1 Human Robot Interaction (HRI)

Human detection and following are the basic concepts of HRI which require real-time computation. Since the robot is not allowed to damage its surrounding environments, HRI should be safe for both humans and their environments. @Home robots are designed to work in human daily environments which are highly dynamic. These robots should be able to follow humans without any collision. In the following text we will describe human following ability of Sepanta.

As Sepanta has a Kinect sensor, depth information of a scene is available and the background of each image can be detected and removed. We have implemented this part of our algorithm using Microsoft Kinect SDK V2. By using these libraries, background is removed and humans are detected properly. In the next step, for human following, OpenTLD [1] is used. This library is an implementation of TLD algorithm [2] which has the ability of learning and tracking a certain part of an image. It is worthy to mention that OpenTLD fails to track human in cluttered backgrounds, so we introduced a novel algorithm for human following which combines the information of both skeleton and OpenTLD which enables the robot to follow human in crowded environments [3]. For now, human face identification and recognition are included and face expression identification will be added for better interaction with human in the future.

4.2 Person Re-identification

Since a service robot has to be able to interact with humans, person recognition during the interaction is a crucial task and should be done in real-time. It is clear that human face is one of the most significant features by which we can detect humans and distinguish them from each other. In this case we used sparse method for face recognition which is invariant to illumination changes and occlusion. For face detection we used Viola-Joines method [4] then we applied sparse method [5] to obtain area for face recognition. Our algorithm for person recognition consists of the following steps;

1. Learning: in this phase we teach the robot via some training data.
2. Face detection using Viola-Joines method.
3. Applying Sparse algorithm to the detected faces.
4. Similarity calculation between detected faces and trained faces.

The input image is coded as a sparse linear combination of the training face images. The output value of algorithm will be used as reconstruction error for each training class. Based on this value the testing input images are classified according to minimum reconstruction error. In this classification, we also have employed body features including height, shoulder width, etc. using skeleton information provided by Kinect sensor [6].

4.3 Voice interaction

One of the most important communication tools is voice interaction, which makes robot's speech skills vital. There are variety of tools and packages that we can use such as Pocket Sphinx[1], Microsoft Speech SDK[2], and Android's speech API[3]. In this robot, we have used Android speech API for speech recognition. To avoid noisy data from the micro-phone, Sepanta is equipped with "Sennheiser MKE400" microphone which has noise canceler. Also, we simultaneously work on Kinect for windows v2 because of using Kinect make it easy to determine location of the speaker. Moreover, for robust sentence recognition, a grammar file is used. Sepanta uses Festival [7] to synthesize speech while talking with people. In our future plan, we are going to implement dialogue generation.

4.4 Simultaneous-Localization and Mapping

In this research, almost like our previous work [8], we have fused data from both a laser scanner and a Kinect sensor for localization and motion estimation. Although during certain tasks visual odometry using Kinect might be disabled to reduce processing. Localization using laser scanner is done using standard AMCL [9]. Visual odometry is more complex and is done using Fovis [10] algorithm, but

[1] Available at http://wiki.ros.org/pocketsphinx
[2] Available at http://www.microsoft.com/
[3] Available at http://developer.android.com/

it will fall back to GICP [11] if Fovis failed to provide a motion estimate. We used a key frame approach for SLAM and place recognition. Key frames are defined when the robot moves a certain distance or rotates for a certain angle. If a key frame is detected, the current image captured by the Kinect sensor will be sent to place recognition module which uses RTAB-Map [12] algorithm. If the image was previously seen, the place recognition algorithm will produce a loop-closure message for the SLAM system. Our SLAM system uses g2o [13] algorithm to sparsely represent the robot's movements.

Although we mostly rely on laser scanner data for navigation using GMapping and ROS navigation package, for more complex environments we can also use OctoMap [14] to produce a 3D map using Kinect's point clouds.

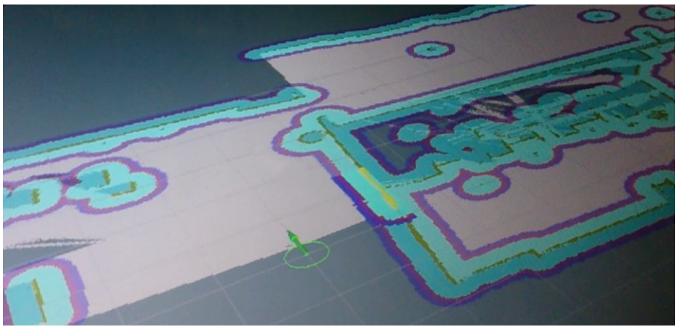

Fig. 3. 2D Cost-map generated with ROS navigation and Hector SLAM.

4.5 Object Recognition

In this section, object recognition algorithm is described. For reduction of computational complexity, a downsampled point cloud dataset using a voxelized grid approach is used. Then, searching for planes starts by using Random Sample Consensus (RANSAC) [15]. After that, the detected planes will be eliminated from the point cloud and the remaining points will be clustered based on Euclidean clustering. For each cluster some key points are extracted based on Intrinsic Shape Signatures (ISS) [16] method. Afterwards, key points are associated to a local description, SHOTCOLOR descriptor [17]. The similarity between this set and previously known sets in the object database will be calculated and the one with largest similarity (smallest distance) is considered as the match for the input cloud. It is clear that if the detected object was not as much as similar to any known objects, it will be classified as unknown.

4.6 Object Manipulation

After object recognition and making the decision to achieve that object, the object manipulation module starts its work. The input of this module is the coordinates of tables and objects in the observed scene and its output is the trajectory which the robot should follow to reach the object.

The output trajectory of this module is influenced by two items; Obstacles which prevents the robot arm to achieve the goal position, and the inverse kinematics of the manipulator. Based on these items, during trajectory generation and execution, the position of the manipulator end-effector and joints are adjusted.

In addition to improve this block we started working on MoveIt[1]. It provides an easy-to-use platform for developing advanced robotics applications and capabilities.

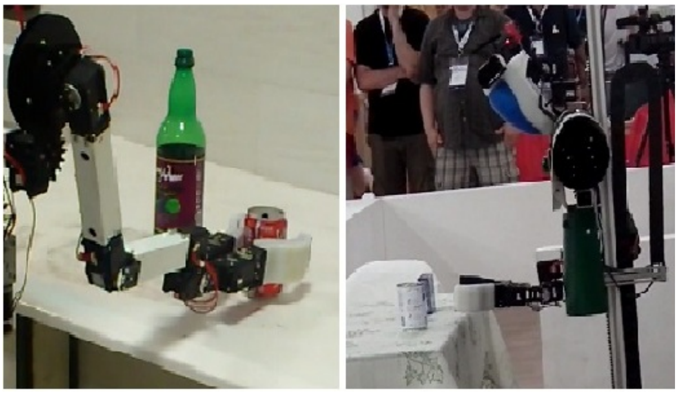

Fig. 4. Object recognition and grasping during the ARC lab tests and Robocup 2014 in Brazil.

5 Conclusions

In this research, we redesigned and improved some abilities of Sepanta and developed new features for it. This feature made Sepanta more maneuverable in dynamic environments. Based on the abilities, this robot can interact with people via speech and other equipment which helps it to complete assigned tasks in Robocup2016. There are some novelties in both robot's construction and algorithms which are mentioned widely in this paper. These novelties are included in object recognition algorithm, human following, mechanical design and other parts. More information about our team and Sepanta is available on our website.

References

1. G. Nebehay, *Robust object tracking based on tracking-learning-detection.* na, 2012.

[1] Available at http://wiki.ros.org/moveit

2. Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.
3. E. Babaians, N. K. Korghond, A. Ahmadi, M. Karimi, and S. S. Ghidary, "Skeleton and visual tracking fusion for human following task of service robots," in *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on*, pp. 761–766, IEEE, 2015.
4. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511, IEEE, 2001.
5. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.
6. S. Gharghabi, F. Shamshirdar, T. A. Shangari, and F. Maroofkhani, "People reidentification using 3d descriptor with skeleton information," in *Informatics, Electronics & Vision (ICIEV), 2015 International Conference on*, pp. 1–5, IEEE, 2015.
7. A. W. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," *University of Edinburgh*, vol. 1, 2002.
8. A. Ahmadi, E. Babaians, A. Badamchi, M. Falahi, S. Gharghabi, R. Gomari, S. Jahangiri, M. Kariminoori, E. Khani, I. Mehrabi, *et al.*, "Aut@ home 2014 team description paper,"
9. F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 1322–1328, IEEE, 1999.
10. A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *International Symposium on Robotics Research (ISRR)*, vol. 2, 2011.
11. A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *Robotics: Science and Systems*, vol. 2, 2009.
12. M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *Robotics, IEEE Transactions on*, vol. 29, no. 3, pp. 734–745, 2013.
13. R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3607–3613, IEEE, 2011.
14. K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.
15. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
16. Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 689–696, IEEE, 2009.
17. F. Tombari, S. Salti, and L. D. Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 809–812, IEEE, 2011.

Team Name: Sepanta (Means "Virtuous", it comes from the Avestan language.)
Contact Information (E-mail): ismaiel.mehrabi@gmail.com
website: http://sar.aut.ac.ir/en/?p=96
Team Members:

Team Leader: Esmaeil Mehrabi
Navigation and DCM: Edwin Babaians, Yaser Sarfi
Mechanic and Electronic: Alireza Ahmadi, Amirreza Kebritchi
Speech: Amin Abedi
Object Recognition: Esmaeil Mehrabi, Mohammad Hosein Heydari
Face Recognition and Person Detection: Faraz Shamshirdar
Arm Navigation: Alireza Zamanian, Esmaeil Mehrabi
Financial Manager: Hamid Ziari, Yasin Noorifard

Hardware:

Base: Fully Holonomic Mecanum wheel platform (self design).
Torso: 1-DOF Prismatic.
Manipulators: Two 6-DOF SCARA based design arms using 12 Dynamixel actuators (4 MX-106)(6 MX-64)(2 MX-28).
Head: 2-DOF (Pant and Tilt) using 2 Dynamixel actuators (MX-64).
Dimensions: Base ($0.4m \times 0.45m$), Height (1.50m).
Sensors: 1 Hokuyo UTM-30LX laser range finder, Kinect for Xbox One RGBD Sensor and 5 IR distance sensors.
Microphone: Kinect for Xbox One microphone array.
Batteries: 3×24 volt custom lithium-ion batteries with up to 3 hours constant operation power.
Computers: 1 Gigabyte mini-pc Brix Core-i7 with 8 GB RAM and 60 GB SSD HDD, 1 Lenovo ThinkPad Laptop Core-i7 with 8 GB RAM and 120 GB SSD HDD.

Software:

Operation System: Ubuntu 14.04 LTS with ROS Indigo on laptop, Windows 10 on mini-pc.
Low-level: RTOS on OpenCM9 Arm based Processor.
Localization: AMCL , Hector SLAM odometers and wheel odometers.
Navigation: Global A* Planner with custom move base engine using cost map for dynamic obstacles. Also 3D perception using Octomap.
Arm Navigation: ROS Move it.
Arm Navigation: SHOT Descriptor using PCL.
People Detection: OpenTLD Tracker , Kinect v2 SDK.
Speech Recognition: Google Speech API.
Speech Recognition: SMACH.