Markovito's Team Description RoboCup@Home 2016

L. Enrique Sucar, Eduardo Morales, José Martínez-Carranza, Luis V. Coronado, David Carrillo, Francisco Marquez, and Juan Carlos González

National Institute for Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro 1, 72840 Tonantzintla, Mexico
http://ccc.inaoep.mx/∼markovito

Abstract. In this paper we present Sabina, a service robot developed by the Markovito team at INAOE. Sabina is based on a PatrolBot robot platform and incorporates a set of general purpose modules for service robots that achieve basic robot skills, such as map building; localization and navigation; object and people recognition and tracking; and human interaction using facial animation, speech and gestures and manipulation. All these modules are integrated in a layered behavior-based architecture implemented on the Robot Operating System (ROS). In addition to these Sabina's capabilities, we also describe a novel approach to autonomous 3D object reconstruction. The Markovito team has participated in the Robocup@Home category in previous Robocup competitions; in Turkey 2011 our team qualified for the second stage of the competition. In 2015 and 2016, the Markovito team won a 1st place in the mexican RoboCup@home competition.

1 Sabina's Hardware Platform.

Sabina is a service robot developed by the Markovito team (see Figure 1), is based on a PatrolBot robot platform [1] with the following components:

- 2 wheels to be controlled.
- 2 motors with encoders.
- 1 laser SICK LMS200.
- 1 sonar ring.
- 1 video camera Canon VCC5.
- 1 set of speakers.
- 1 directional microphone.
- 1 integrated computer.
- 2 latops.
- 2 web cams.
- 1 Katana 6M arm with 5 DOF
- 2 Kinect device.

(a) PatrolBot Platform (b) Katana 6M arm with 5 DOF

Fig. 1. Sabina's hardware platform.

2 Sabina's Software Architecture

Sabina's software architecture has been designed and developed as a layered behavior-based [2] architecture that use shared memory for communication. As you can see in Figure 2 all general porpuse modules developed are integrated in this architecture. The architecture has three different levels:

1. Functional Level: Here the modules interact with the robot sensors and actuators, relaying commands to the motor or retrieving information from the sensors.
2. Execution level: Modules in this level interact with the functional level through shared memory. This level includes the modules to perform basic tasks such as navigation, loclization, visual perception, human-robot interaction, etc.
3. Decision level: This is the highest level in the architecture, Markov Desicion Processes (MDPs) are used as a global planner to coordinate the execution level modules.

The layered structure and a transparent communication mechanism allow different configurations to be defined without modifying the modules and without affecting the rest of the system. In this architecture, a robot behavior is an independent software module that solves a particular problem, such as navigation or gesture recognition. The complete system was developed using mainly C/C++ language and run on Linux.

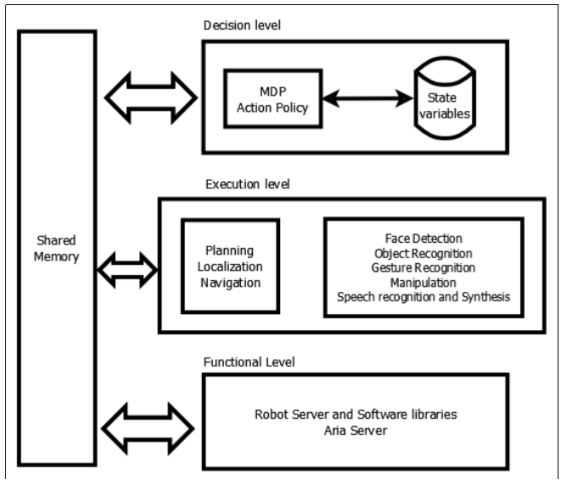


Fig. 2. The software architecture of Sabina: a layered behavior-based architecture that use shared memory for communication.

2.1 The Execution Level

We are implementing different general-porpuse modules that are common to several services robot's applications. All of these modules are in the execution level on the Sabina's architecture. The libraries used in these modules are listed in the Table 1.

Table 1. Modules and libraries used in Markovito.

Module	Source code/Libraries
Navigation	Aria, ARNL, MRPT
Vision	OpenCV, SIFT algorithm, ORB descriptors, OpenKinect
Interaction	Festival, Microsof Speech Recognition, OpenGL Custom Render
Coodination	SPUDD (MDP)

Map building and Planning The first task that a service robot has to do is to know the environment where it will be. For this porpuse a robot service requires a model or map of this environment. Sabina combines information from a laser scanner and odometer to construct an occupancy map. We use the ICP algorithm from Mobile Robot Programming Toolkit (MRPT) [3] to build this model. The ICP algorithm matches a point cloud and some reference. In this case, the point cloud are the current laser readings, the reference are the previous one (see Figure 3).
For planning purposes, a probabilistic roadmap (PRM) [4, 5] is built using a random generation of points in the configuration space. These points are joined

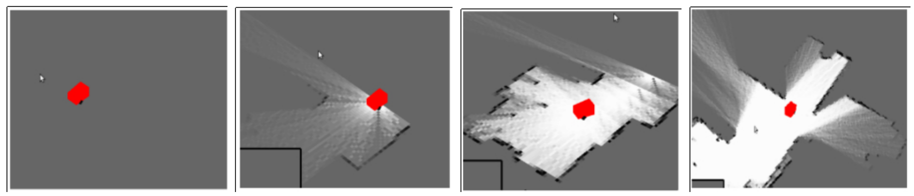


Fig. 3. Sabina building a map of the environment using its SLAM algorithm based on MRPT ICP.

if there is a free path between them and information is stored in a graph G. Given an initial configuration s and a goal configuration g, the problem consists of connecting s and g in G. This process is illustrated in Figure 4.

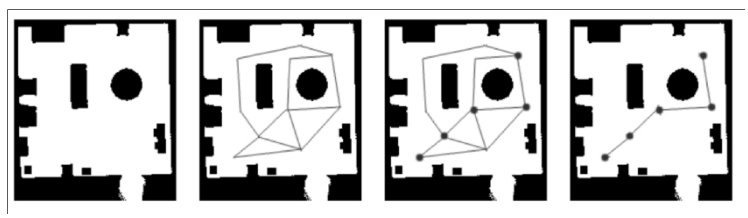
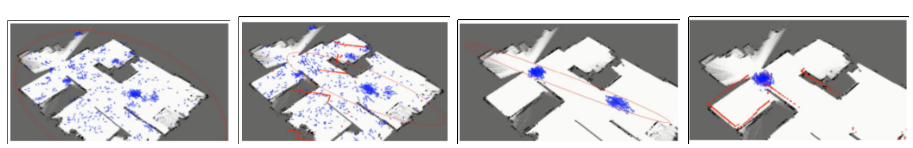


Fig. 4. Example of a PRM constructed. The PRM is build using a random generation points. The points are joined if there is a free path between them. Intermediate points in the path are given as a goals to the navigation module.

Localization and Navigation Due to odometric error, the ability for mobile robots to locate themselves in an environment is not only a fundamental problem in robotics but also a pre-requisite for safe navigation tasks. In order to locate itself, Sabina uses laser information. Given a set of laser readings, a MRPT particle filter process is performed to estimate the real robot position. Figure 5 shows the localization process.



(a) (b) (c) (d)

Fig. 5. Sabina running the MRPT particle filter localization. a),b) c) Shows the algorithm initialization. d) Shows the algorithm convergence, the mean of the particles is the robot position estimation.

In order to control the robot's movement to follow a specific path, we have implemented a simple but effective navigation module based on Aria actions [1]. Also we use the integrated ARNL library [1] for navigating in more complex environments.

Face Detection and Recognition We have developed a face recognition system allowing a mobile robot to learn new faces and recognize them in indoor environments (see Figure 6). First, the image is enhanced by equalizing its histogram and performing a local illumination compensation [6]. Next, we use an object detection scheme based on a boosted cascade of simple feature classifiers [7] to detect eyes, mouth, and nose. For each region, SIFT features [8] are extracted. The features in a sequence of images are compared to the models in a database using a Bayesian scheme.

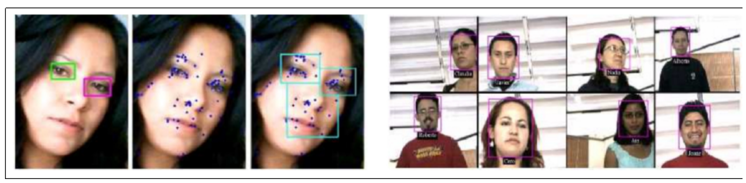


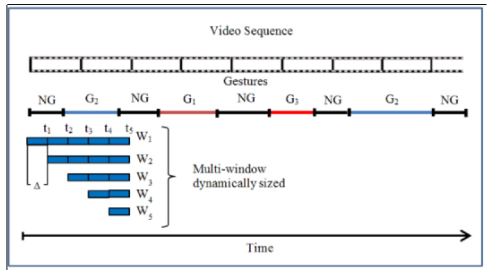
Fig. 6. Face detection and recognition.

Gesture Recognition To recognize gestures we are incorporating the Kinect device to take advantage of its depth information. Our approach allows the robot to simultaneously segment and recognize gestures from continuous video [11]. Figure 7a shows Sabina recognizing the attention gesture, Figure 7b shows a graphical description of the approach. We use dynamic windows to segment the video secuence. Each time step, the features belonging to every window is evaluated to get a gesture prediction. Windows that better fit the segment of the gesture, will produce a higher probability and thus their votes must have higher weight.

Human Tracking In this module we extract torso boundaries using a histogram and the back projection image [9] coupled with Haar functions [10] with a monocular camera. Our torso detection and tracking system is divided in two stages. The first stage is the torso localization process, that uses a face detection algorithm based on color histograms in RGB. Once the face is detected, the torso position is estimated based on human biometry. The color histogram of the torso is registered by this module. The second stage consists of tracking the torso using the color histogram obtained at the first stage, coupled with detectors based on motion and appearance information. Finally, a distance transform is applied, considering a pinhole camera model.

Object Recognition Currently we are exploring the use of ORB algorithm which is basically a fusion of FAST keypoint detector and BRIEF descriptor.

(a) Sabina recognizing a user gesture (taken from [11])



(b) Graphical description of simultaneously segmentation and recognition (taken from [11])

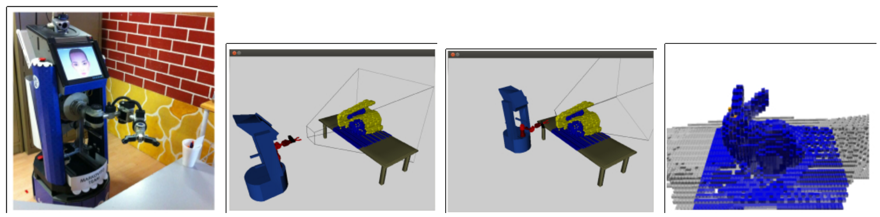
Fig. 7. Sabina's recognition module for continous video. a) Sabina capture user feature trough the Kinect Device. b)Diagram illustrating the simultaneous segmentation and recognition. The system has a video secuence as input. Then 5 windows are generated every time step. W1 starts at t0 , W2 starts at t1 , and so on. Each window is evaluated an voted, all continue simultaneously until certain stop criteria is met.

It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch. This process is done on pre-processed images where a background removal is done using depth information from the Microsoft kinect mounted on top of our robot.

Manipulation and 3D Reconstruction For manipulation purposes we have added a Katana arm (see Figure 1b) to the Patrolbot platform. This arm is able of grasping objects which are inside its reachable space.

Once that object and robot positions have been obtained inside the environment (point cloud), we plan the controls to reach the configuration which grasps the object by using a Rapidly-exploring Random Tree Technique (RRT) [12]. Here our goal state is the grasping configuration (see Figure 8.a). The environment where the RRT checks for collision is given by an octree updated with the point cloud of the environment. Sabina also has the ability of building 3D models from real objects in its environment. We have developed an autonomous 3D object reconstruction algorithm. To generate the model, several scans from different configurations are taken by the robot. For each configuration, the robot takes a scan, updates the incremental representation of the object and plans the next configuration. To plan the next configuration, we uniformly sample the robot configuration space and test each sample in order to evaluate their utility. The configuration with the highest utility is selected. In addition, the trajectory to reach the planned configuration is calculated with a Rapidly-exploring Random Tree (RRT). Figure 8 shows some simulated experiments of this approach.

Speech Recognition and Synthesis We use Microsoft Speech Recognition[14] engine and Kinect SDK libraries for speech recognition. Different dictionaries or



(a) (b) (c) (d)

Fig. 8. Autonomous reconstruction 3D experiment.

sets of recognizable phrases are defined depending of the task to be performed by the robot. The system can identify only the phrases or words defined in its dictionary. The coordinator (MDP) sets the right set of phrases to be used by the speech recognition module on each task. For synthesis we use Festival [13].

Facial Animation We have incorporated a set of emotions such as happiness, anger and surprise to Sabina by providing it with a friendly animated face (see Figure 9). The animation is done with key-frames interpolation. OpenGL is used to render the 3D model and key postures and timing information are defined a priori.

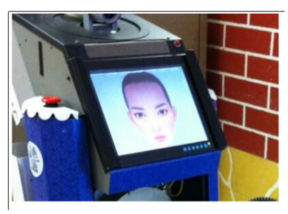


Fig. 9. Sabina's face.

2.2 The Decision level

The behavior modules are coordinated by a decision-theoretic controller based on MDPs [15]. An MDP is specified for each task and solved to obtain an optimal policy. In our current implementation we use a factored representation to specify the MDPs and SPUDD [16] to solve them. The model is specified manually by the programmer according to the task. We use an interactive approach to define the model.

3 Conclusions

In this document, Sabina's hardware and software architecture have been described. The robot is based on a PatrolBot platform where an arm, a kinect device and a set of web cams have been added in order to get a better perception performance and effective interaction with people. For such tasks, we have also developed a set of general purpose modules, integrated in a layered behavior-based architecture based on ROS. These features enable Sabina to perform different RoboCup@Home tasks. Based on this framework and a PeopleBot platform, we have participated in the Mexican Robotic Tournament since 2007 achieving top positions each year, for instance, in 2015 and 2016 we won a 1st place in the Mexican RoboCup@home competition. We have also participated in the international Robocup@Home competition in 2009, 2011 and 2012.

References

[1] MobileRobots Inc. The PatrolBot platform. http://www.mobilerobots.com/ResearchRobots/ResearchPatrolBot.aspx (2014).
[2] Arkin, R. C.: Behavior-based Robotics. MIT Press (1998).
[3] Mobile Robot Programming Toolkit. http://www.mrpt.org (2014).
[4] Choset Howie, Lynch Kevin M., Hutchinson Seth, Kantor George, Burgard Wolfram, Kavraki Lydia E., and Thrun Sebastian: Principles of robot motion: Thoery, Algorithms, and implementations. MIT Press, Cambridge,MA (2005).
[5] Latombe J. C.: Robot Motion Planning Kluwer Academic Publishers (1991).
[6] Ramírez-García G.: Detección de rostros con aprendizaje automático. Masther thesis, INAOE (2006).
[7] Viola P. and Jones M.: Rapid Object detection using a boosted cascade of simple features. In Proceedings IEEE Conference on Computer Vison and Pattern Recognition (2001).
[8] Lowe D.: Distint image features from scale-invariant features. In Proceedings of the Seventh International Conference on Computer Vision, pages 1150-1157 (1999).
[9] Bradski G.R.: Real time face and object tracking as a component of a perceptual user interface. In IEEE Computer Society, editor, Proceedings of the 4th IEEE Workshop on Applications and Computer Vision, pages 214-219 (1998).
[10] Snow D., Jones M. J., and Viola P.: Detecting pedestrians using patterns of motion and appearence. In Ninnth IEEE International Conference on Computer Vision, pages 734-741 (2003).
[11] Towards Human-Robot-Interaction in Continuous video with Gestures for Sabina. Harold Vasquez Chavarria, Hugo Jair Escalante, L. Enrique Sucar, Miguel Palacios, Patrick Heyer. Proceedings of the Workshop on Robocup@home league. ICAR 2013 - 16th International Conference on Advanced Robotics. (2013)
[12] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), pages 378-400, (2001).
[13] Festival Speech Synthesis.http://www.cstr.ed.ac.uk/projects/festival/ (2014).
[14] Developer Network Microsoft. http://msdn.microsoft.com/en-us/library/System.Speech.Recognition(2014)
[15] Puterman M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley and Sons, Inc, 1st edition (1994).
[16] Hoey J., St-Aubin R., Hu A.,and Boutilier C.: SPUDD:Stochastic planning using decision diagrams. In Proceedings of International Conference on Uncertainty in Artificial Intelligence, Stockholm (1999).