Markovito's Team Description RoboCup@Home 2012

L. Enrique Sucar[1], Eduardo Morales[1], Patrick Heyer[1], Irving Vasquez[1], Miguel A. Palacios-Alonso[1], Hugo Jair Escalante[1], J. Manuel Oropeza[1], Shender Ávila[1], Aarón Rocha[1], Juan Herrera[1], Angel Ocaña[1], Alejandro Reyes[1], Harold Vázquez[1], Paulette Zorilla[1], Jean Hayet[2], Mariano Rivera[2], Judith Esquivel[2], Domingo Rodriguez[2], and Claudia Estevez[3]

[1] National Institute for Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro 1, 72840 Tonantzintla, México
{esucar,emorales}@inaoep.mx,
http://ccc.inaoep.mx/~markovito

[2] Research Center in Mathematics
Computer Science Department,
Jalisco S/N, Col. Valenciana, 36240 Guanajuato, México
{mrivera,jbhayet}@cimat.mx

[3] Guanajuato University
Lascuráin de Retana 5 Centro, 36000 Guanajuato, México

Abstract. With the incorporation of service robots to daily activities, it is expected that they will require to perform different complex tasks. Althought there is much work in developing different abilities for this kind of robots, little attention has been paid for the integration of these behaviors into a complete functional system. In this paper we present *Sabina*, a service robot that incorporates a set of general modules 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. *Sabina*'s arquitecture considers a general framework to easily develop different applications. Markovito's team have participated in the Robocup@Home category in the previous Robocup competitions; in Turkey 2011 our team qualified for the second stage of the competition.

1 Hardware and Software Platforms.

Sabina is a service robot based on a PatrolBot robot platform [1]. It has a sonar ring, two wheels, two motors with encoders, a Laser SICK LMS200, one video camera Canon VCC5, a directional microphone SHURE SM81, speakers, an integrated PC, a standard Laptop, a Katana 6M arm, and a Kinect device (see Figure 1).

We have developed a set of general purpose modules for service robots, integrated in a layered behavior-based architecture and using share memory for communication. In our architecture exists three different levels: i) functional level, ii) execution level and iii) decision level. This allows that a module can be changed without affecting the rest of the system. The software libraries and source code considered to develop each module are summarized in Table 1.

Fig. 1. *Sabina*, INAOE's service robot based on a PatrolBot platform

1.1 Software Architecture.

Sabina's software architecture is based on a Behavior-based architecture [2]. In this archictecture, a behavior is an independent software module that solves a particular problem, such as navigation or object recognition.

In this paper, behaviors are also referred as modules. Behaviors exist at three different levels (see Figure 2):

- Functional Level: In this level, the modules interact with the robot's sensors and actuators, relaying commands to the motors or retrieving information from the sensors.
- Execution level: The modules in this level interact with the functional level through a TCP connection to Player server. This level includes the modules

Table 1. Modules and libraries used in Sabina.

Module	Source code/Libraries
Navigation	Player/Stage server, Pmap utility, Sw-Prolog
Vision	Player/Stage server, OpenCV, SIFT algorithm,OpenKinect
Interaction	Sphinx II, Audacity, Listener, Festival, Player/Stage server
	OpenGL Custom Render
Coordinator	SPUDD (MDP)

to perform basic tasks such as navigation, localization, visual perception, human-robot interaction, etc.

– Decision Level: This is the highest level in the architecture, coordinates the execution level modules using a global planner based on Markov Decision Processes (MDPs).

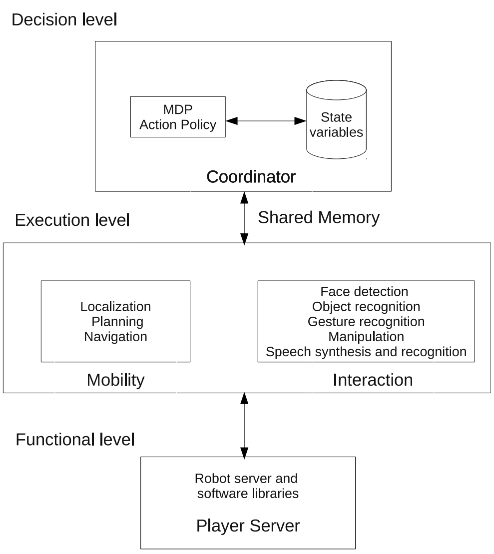


Fig. 2. Sabina's software architecture

This architecture can be implemented in a distributed platform, such that each level module within a level can run on a different processor. A transparent communication mechanism allows different configurations to be defined without modifying the modules. Also, a module can be changed without affecting the rest of the system. The complete system was developed using mainly C/C++ language.

2 The Modules.

Our research focuses on the development of independent general software modules. We are implementing different general-purpose modules that are common to several services robot's applications. The following describes the behavior modules we are currently working on.

2.1 Map Building and Localization.

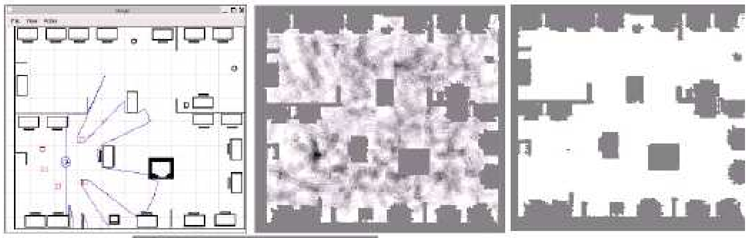
A mobile robot requires a model or map of its environment to perform tasks. *Sabina* combines information from a laser scanner and odometer to construct an occupancy map based on particle filters [5]. Each particle represents a trajectory followed by the robot and a map associated with that path.

The ability for mobile robots to locate themselves in an environment is not only a fundamental problem in robotics but also a pre-requisite to many tasks such as navigation. There are two types of localization problem: local and global. In order to locate itself, *Sabina* uses natural landmarks such as discontinuities, corners and walls as described in [3]. Given a set of landmarks, a triangulation process is performed between all the visible landmarks to estimate the robot's position. Figure 3a shows the global localization process.

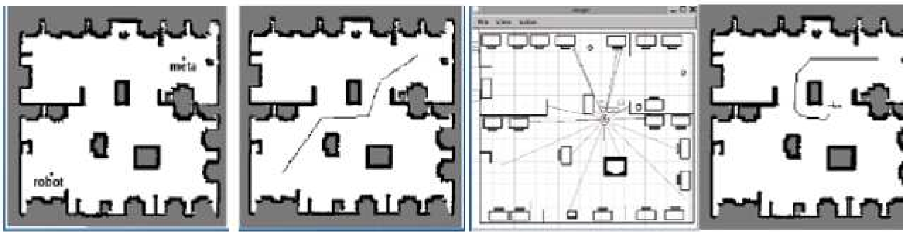
2.2 Navigation.

We have implemented a navigation module that uses a dynamic programming algorithm to assign costs to each cell of the map; in this approach an exponential cost function is used to value every cell near an obstacle below a distance threshold. Following this criterion, the least expensive path is obtained by selecting cells with a lower cost. In order to avoid new obstacles, the robot is sensing its environment while moving.

We are currently working on a novel navigation strategy that uses machine learning techniques. In case a new obstacle is placed in front of the robot the module finds an alternative path (see Figure 3b). The robot has to learn how to perform simple skills, like obstacle avoidance and orientation towards a goal. First-order logic relations are learned to build the navigation module. This module consists of a set of reactive rules known as TRP's (TeleReactive Programs) [4].



(a) Global localization. Showing the robot and some discontinuities. The robot is localized after the second stage



(b) Re-planning with obstructed paths.

Fig. 3. Simulated examples of the navigation process

2.3 Planning.

The previous TRP's are used with a probabilistic roadmap module that returns collision free paths. A probabilistic roadmap (PRM) [6, 7] is build using a random generation of points in the configuration space. These points are joined 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 to nodes s' and g' in G. The points in the path are given as intermediate goals to the navigation algorithm. This process is illustrated in figure 4.

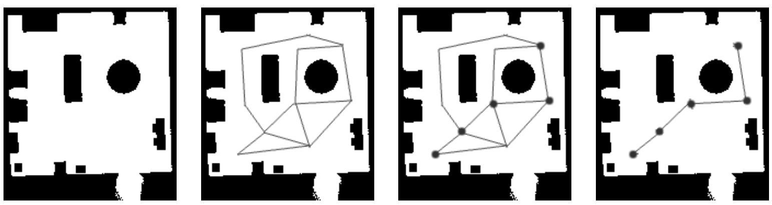


Fig. 4. PRM constructed for the navigation process. The PRM is build using a random generation points. The points are joined if there is a free path between them. The points in the path are given as a intermediate goals to the navigation algorithm.

2.4 Visual Perception

Service robots must integrate sophisticated perception to operate in complex and dynamic enviroments. *Sabina* combines several types of sensors, such as sonar, laser, sound and vision. In order to interact effectively with people and its environment a service robot must integrate abilities such as people detection, recognition of their activities and object recognition.

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. First, the image is enhanced by equalizing its histogram and performing a local illumination compensation [8]. Next, we use an object detection scheme based on a boosted cascade of simple feature classifiers [9] to detect eyes, mouth, and nose. For each region, SIFT features [10] are extracted. The features in a sequence of images are compared to the models in a database using a Bayesian scheme.

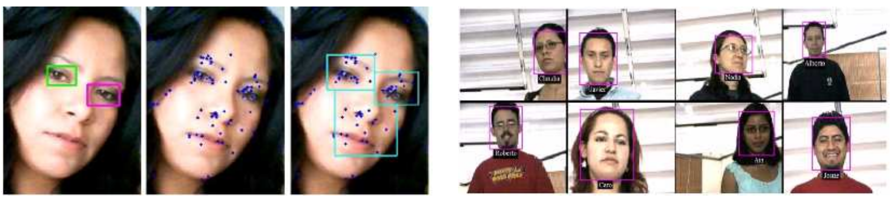


Fig. 5. Face detection and recognition

Human Tracking. In this module we extract torso boundaries using a histogram and the back projection image [11] coupled with Haar functions [12] 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.

Gesture Recognition. To recognize gestures we propose an alternative model for hidden Markov models (HMMs), that we call dynamic naive bayesian classifiers (DNBCs) [13]; which combines motion and posture features extracted from the images for gesture classification.

At the present time, we are incorporating the Kinect device to take advantage of its depth information.

Object Recognition. We are working on a new approach that uses an ensamble of classifiers based on color and local features. This approach use several measures of similarity between images (see Figure 6), such measures are assigned weights to obtain a normalized measure that represents the similarity between two images. The similarity measures used are: HSV normalized cross-correlation, absolute differences in HSV, absolute differences in histograms discretized in HSV, mutual information of the HSV histograms, differences in absolute value in SCILAB, and SURF features.

Fig. 6. Object recognition using measures of similarity. The object in the red rectangle is recognized in the image.

Manipulation. The Katana arm (see Figure 7) will be capable of grasping objects which are inside its reachable space. This task assumes that: i) the object has been previously recognized and its position (in R^3) with respect to the robot is given, ii) a point cloud of the environment (Robot's surrounding space) is provided, iii) the object has been segmented inside this point cloud, iv) a 3D approximated model of the object is given, and v) there is a path to grasp the object.

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 [14].

Rapidly-exploring Random Tree (RRT) is a technique which provides the controls to take the robot from an initial state to a goal state. Here our goal state is the grasping configuration. The environment where the RRT checks for collision is given by an octree updated with the point cloud of the environment.

Speech Recognition and Synthesis. This functionality is performed using a standard laptop computer. We use Festival [15] and Sphinx II libraries [16].
Different dictionaries or sets of recognizable phrases are defined depending of the task to be performed by the robot. The system can identify only the phrases

Fig. 7. Sabina's Katana arm

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.

Facial Animation. Friendlines and user acceptance are improved by providing the robot with a face to which the user can talk. We are incorporating expressiveness capabilities to convey a basic set of emotions such as happiness, anger and surprise (see Figure 8). The animation is done through interpolation of key-frames . Key postures and timing information are defined *a priori*. The 3D rendering is done with OpenGL.

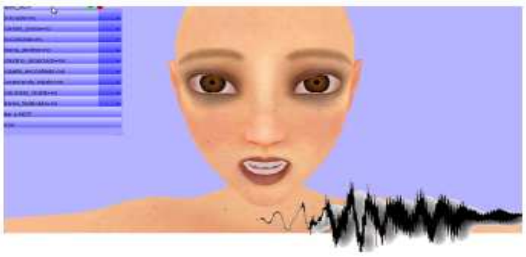


Fig. 8. Animated interface of *Sabina*

3 Coordinator

The behavior modules are coordinated by a decision-theoretic controller based on MDPs [17]. 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 [18] to solve them.

The model is specified manually by the programmer according to the task. We use an interactive approach to define the model. An initial model is defined and solved with SPUDD. Then we used a simulator to verify the obtained policy, and if there are inconsistent or strange actions, the model is modified and the process is repeated.

We are currently working on a framework for concurrent execution of actions coming from different modules. Subsequently, through a learning process, the robot learns to coordinate the actions of each module.

4 Conclusions

We have developed a set of general purpose modules for service robots, integrated in a layered behavior-based architecture. These modules are utilized for performing the different RoboCup@Home tasks coordinated by a decision-theoretic controller. Based on this framework and a PeopleBot platform we have participated in the Mexican Robotic tournament with our robot *Markovito* since 2007; and in the international Robocup@Home competition in 2009 and 2011, in the last one we classified for the second stage. We are currently developing a new robot, *Sabina*, based on a PatrolBot and the same software framework to compete in RoboCup 2012.

References

1. MobileRobots Inc. The PatrolBot platform. *http://www.mobilerobots.com/ResearchRobots/ResearchPatrolBot.aspx* (2012).
2. Arkin, R. C.: Behavior-based Robotics. MIT Press (1998).
3. Henández S. and Morales E.: Global localization of mobile robots for indoor environments using natural landmarks. In Proceedings IEEE Conference on Robotics, Automation and Mechatronics, pages 1-6 (2006).
4. Benson Scot and Nilsson Nills J.: Reacting, planning and learning in an autonomous agent Oxford University Press, Inc, pages 29-62 (1996).
5. Jáquez V.: Construcción de mapas de localización simultánea con robots móviles. Masther thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey- Cuernavaca (2005).
6. 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).
7. Latombe J. C.: Robot Motion Planning Kluwer Academic Publishers (1991).
8. Ramírez-García G.: Detección de rostros con aprendizaje automático. Masther thesis, INAOE (2006).

9. 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).
10. Lowe D.: Distint image features from scale-invariant features. In Proceedings of the Seventh International Conference on Computer Vision, pages 1150-1157 (1999).
11. 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).
12. 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).
13. Avilés-Arriaga H., Sucar L. E., and Mendoza E.: Visual Recognition of similar gestures. In 18th International conference on Pattern Recognition of similar gestures, pages 1100-1103 (2006).
14. Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), pages 378-400, (2001).
15. The Festival Speech Synthesis System. *http://www.cstr.ed.ac.uk/projects/festival/* (2012).
16. Open Source Speech Software from Carnegie Mellon University. *http://www.speech.cs.cmu.edu/* (2012).
17. Puterman M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc, 1st edition (1994).
18. 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).