A General Framework for Developing Service Robots

L. Enrique Sucar and Eduardo F. Morales
Department of Computer Science, INAOE
Luis Enrique Erro # 1
Tonantzintla, Puebla, 72840 México
{esucar,emorales}@inaoep.mx

Abstract

We present a general framework for developing service robots that can help people in daily activities. This framework is based on a general, distributed architecture which integrates several components; (i) coordinator, (ii) navigator, (iii) perception, and (iv) human-robot interface. The coordinator uses a decision-theoretic approach to select the appropriate action according to the current state. The navigator allows the robot to localize itself and navigate in a dynamic environment, using natural landmarks. The perception module combines vision, sonar and lasers so that the robot can detect the relevant objects in the environment, including people. The human-robot interface provides a natural communication with people, using voice, gestures and portable devices. This provides a general and flexible framework for developing house hold robots, so the same infrastructure can be applied to different tasks by just changing the coordinator, reducing costs in the development of different applications.

Keywords: service robots, architectures, navigation, perception, human-robot interaction.

1 Introduction

We are concerned with the problem of building service mobile robots. For example, robots that can help elderly people in their home, robots that can serve as hosts and guides in museums or shopping malls, robots that can serve as aides in hospitals, etc. Such robots need navigation, mapping, localization and obstacle avoidance capabilities to move around in an uncertain and changing environment. They also need to model the dynamics of people in an environment, including their locations in space and their behavioral patterns. Finally, human users require such robots to present clear, simple and natural interactive interfaces, which enable easy exchanges of information between robots and humans. Several applications of service robots require similar capabilities. For instance, in most applications the robot has to navigate from one place to another, has to localize itself, has to identify and interact with people, among other common tasks. Thus, if we can have a set of software modules to perform these tasks and a general architecture to combine them, we can easily adapt a robot for different service applications. In this work we propose a general architecture for service robots, including several general purpose components for (i) navigation and localization, (ii) person detection, (iii) human-robot communication, and (iv) task coordination. We illustrate this framework in the design of two different service robots, a messenger and a host.

2 Related work

Building service robots to help people has been the subject of recent research. The challenge is to achieve reliable systems that operate in highly dynamic environments and have easy to use interfaces. This involves solving both the more traditional robot problems of navigation and localization and the more recent problems in human-robot interaction.

RHINO [4], was one of the most successful service robots ever built. It was designed as a museum tour guide. RHINO successfully navigated a very dynamic environment using laser sensors and interacted with people using prerecorded information; a person could select a specific tour of the museum by pressing one of many buttons on the robot. RHINO's task planning was specified using an extension to the GOLOG language called GOLEX.

MINERVA [14], was the successor of RHINO. It differed from RHINO in that it could generate tours of exhibits in real-time as opposed to choosing one of several predetermined tours. MINERVA also improved on the interaction by incorporating a steerable head capable of displaying different emotional states. The GOLOG language was combined with decision theoretic planners in DTGOLOG, used in the implementation of a service delivery robot [3].

More recently, the robot PEARL escorted elderly people around an assisted living facility [12]. Its navigation and localization used probabilistic techniques with laser sensors. PEARL is more focused on the interaction side with an expressive face and a speech recognition engine. One of PEARL's contributions is the use of a hierarchical partially observable Markov decision process (HPOMDP), which is an extension of hierarchical MDPs (HMDPs) [5] to model uncertain observations. HMDPs use an specified hierarchical decomposition of the domain, and introduce *abstract actions* in higher level MDPs which invoke the policies of lower-level MDPs.

3 Software Architecture

Crucial to the design of a human-interactive mobile robot is the ability to rapidly and easily modify the robot's behavior. These design constraints call for a mobile robot to have a modular software system, with a planning module that coordinates the different software modules to achieve the goal.

Our software architecture is based on a Behavior-based architecture [1]. A behavior is an independent software module that solves a particular problem, such as navigation or face detection. We refer to behaviors interchangeably as modules. Behaviors exist at 3 different levels:

Functional level: The lowest level behaviors interface with the robot's sensors and actuators, relaying commands to the motors or retrieving information from the sensors.

Execution level: Middle level modules perform the main functions, such as navigation, localization, speech recognition, etc. These interface with the lowest level through a shared memory mechanism. Each middle level module computes some aspect of the state of the environment. The outputs of these modules are typically reported to the highest level modules.

Decision level: The highest level coordinates the middle level modules using a global planner based on Markov decision processes.

This architecture can be implemented in a distributed platform, such that each level and each module within a level could be on a different processor. A transparent communication mechanism permits different configurations without need to modify the modules. Thus, some processing could be done on board the robot (lower level modules) and other off board (high level modules).

4 Coordinator

4.1 Markov Decision Processes

Markov decision processes (MDPs) have become the semantic model of choice for decision theoretic planning (DTP) in the AI community [13]. They are simple for domain experts to specify, or can be learned from data. They have many well studied properties including approximate solution and learning techniques. An MDP is a tuple $\{\mathcal{S}, \mathcal{A}, \Pr, R\}$, where $\mathcal{S}$ is a finite set of states and $\mathcal{A}$ is a finite set of actions. Actions induce stochastic state transitions, with $\Pr(s, a, t)$ denoting the probability with which state t is reached when action a is executed at state s. $R(s, a)$ is a real-valued reward function, associating with each state s and action a. Solving an MDP is finding a mapping from states to actions. Solutions are evaluated based on an optimality criterion such as the expected total reward. An optimal solution is one that achieves the maximum over the optimality measure, while an approximate solution comes to within some bound of the maximum.

The space and time complexity of MDPs increases with the number of states. This problem is reduced by using factored representations [10], in which the state is decomposed in a set of variables or factors, and the transition functions are represented using dynamic Bayesian nets. However, if we require simultaneous actions using a single MDP, we need to consider all the possible action combinations, which will imply an additional increase in the size of the state-action space. So we propose a framework for task coordination based on multiple MDPs, that we call *Multiply Sectioned Markov Decision Processes* (*MS-MDPs*) [6].

4.2 Multiply Sectioned MDPs

A MS-MDP is a set of N MDPs, all of which share the same goal and state space, but have different action sets. We assume that the actions of each MDP do not *conflict* with the other processes, so that each action set can be executed concurrently with the others. We do not find optimal solutions for the global MDP, but simply simultaneously execute the optimal solutions from each sub-MDP. Intuitively, we can think that each MDP is solving one aspect of the global task, coordinated by a common state vector, and in this way accomplish the common goal.

Given that we have a factored representation of the state space, each MDP only needs to consider the state variables that directly influence its actions and reward. This implies that each MDP, P_i, will in general have a subset of the state variables spanning its local state space, S_i. Further, we do not consider the effects of combined actions. These two aspects can make a considerable reduction in the action-state of the problem.

Although the assumption of no conflicts could seem too restrictive, it is not so for many applications. If there are potential conflicts there are two basic alternatives. One is to duplicate certain resources, for example, have one camera for navigation and another for human interaction. The other is to impose certain restrictions, such as do not look for persons while navigating.

5 Map Building and Navigator

A mobile robot requires a model or map of its environment to perform many tasks. Our map building module uses information from a laser scan, its odometer and a sonar ring to construct an occupancy grid map using particle filters. Due to space constraints, we will not describe this module in detail (see [9] for more information).

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 problems: local and global. Local localization techniques aim to compensate for odometric errors during navigation and require information about the initial position of the robot. Global localization aims to locate the robot's position without prior knowledge of its current location. These problems are particularly hard in dynamic environments where new obstacles can corrupt the robot's sensor measurements [4].

In order to locate itself either during navigation or globally, this module uses natural landmarks that have the advantage that the environment does not need to be transformed. In this work, we use discontinuities, that can be easily extracted from laser scanner data with high accuracy, to solve the local and global localization problem.

A discontinuity is defined as an abrupt variation in the measured distance of two consecutive readings of the laser, as shown in Figure 1. Given a set of landmarks (discontinuities), a triangulation process is performed between all the visible landmarks to estimate the robot's position. The information from all the visible landmarks is combined considering the angle between landmarks, the distance between the robot and its farthest landmark, and if there are land-

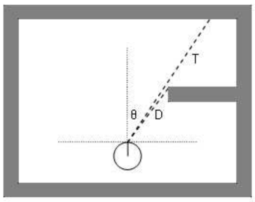

Figure 1: An example of a discontinuity extracted from laser scanner data.

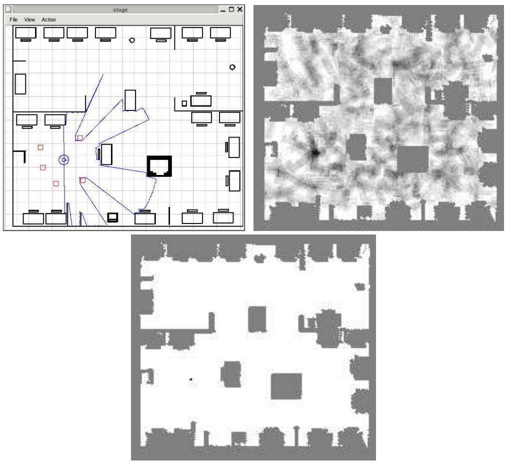

Figure 2: Global localization: after the discrete relaxation algorithm a few cells are the most probable locations.

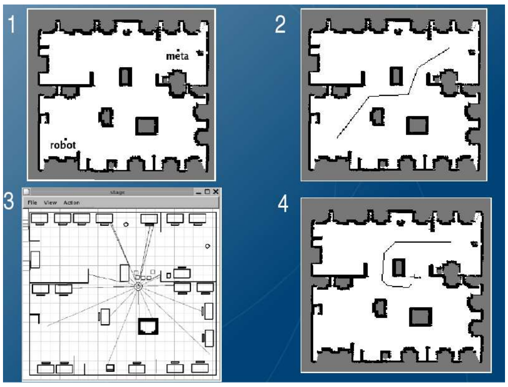

Figure 3: Re-planning with obstructed paths.

marks at both sides of the robot or only one, to give more accurate estimates.

For the global localization problem, a ray tracing approach is used to simulate laser readings on the map. Each cell is associated with all its visible landmarks and their values. This process is performed off-line once a map is constructed. To match a cell with the current readings of the robot, an initial stage filters out a large number of candidate positions with a fast algorithm. It counts the number of discontinuities obtained from the laser data that matches the distance, depth, and orientation of the previously stored discontinuities associated to each cell. A modified discrete relaxation algorithm is used in a second stage to determine the similarity of each cell with the observations of the robot, considering the distances between the discontinuities in this stage. Our global localization algorithm is able to locate the robot even with new obstacles as can be seen in Figure 2, where 5 new obstacles are added to the environment (see [8] for more details).

The navigation module uses a dynamic programming algorithm, with exponential costs near obstacles, to find the least expensive path. In order to avoid new obstacles, the robot is sensing its environment while moving. In case a new obstacle is placed in front of the robot the module finds an alternative path, as shown in Figure 3 (see [7] for more details).

6 Perception

Service robots require a sophisticated perception to operate in complex and dynamic environments. We use a multi–modal approach, combining several types of sensors, such as vision, sonar, laser, and sound. Some perception capabilities are integrated within specific modules, such as the navigator and gesture recognizer. Other are developed as specific modules such that they could be used by different subsystems. One example is a person detector.

The person detector combines sound and vision to detect persons in the environment. The sound subsystem detects the proximity of people by differentiating speech from non-speech sounds. Based on frequency attributes from the sound signal, a Bayesian classifier was trained with samples of speech and other common sounds. On-line, the sound classifier can detect a person when she speaks in a range of about 4 meters. The vision subsystem is based on color information to detect people, basically faces. We combined several color models and used a feature selection approach to determine the *best* color attributes (R,G,Y) for skin detection [11]. These components are integrated using a Bayesian classifier in a robust and very efficient skin detector. Both subsystems are combined using a probabilistic OR, assuming that they are independent. That is, a person is detected if $P(S\ OR\ V) = 1[(1 - P(S))(1 - P(V))] > T$, where $P(S)$ is the probability of speech given by the sound subsystem, $P(V)$ is the probability of skin given by the vision subsystem, and T is a threshold. Figure 4 shows an example of face and hand detection.

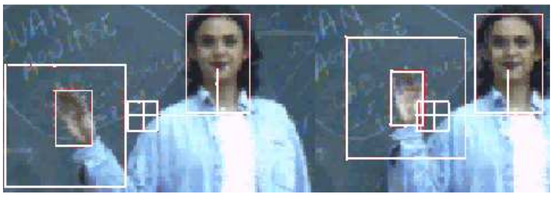

Figure 4: Face detection and tracking of the user's right hand.

7 Human-Robot Interface

Service robots require a natural and effective communication with humans. We consider a multi–modal framework for human–robot interaction with 3 main modes: (i) speech, (ii) gestures, and (iii) portable devices. For speech communication we use a standard, off-the-shelf speech recognition and synthesis system. Depending on the domain, a specific vocabulary and syntax is defined which is used by the recognition system. For many applications, a reduced vocabulary with a simple syntax is appropriate.

Beside speech, gestures are an important element in human–human communication, and are also useful for human–robot interaction. Some commands are easier to communicate with gestures, such as go there, bring me that object, etc. Also, gestures could help as a complement to speech. We have designed a set of natural gestures to command mobile robots, and a gesture recognition system based on dynamic Bayesian networks (DBN) [2]. Once a person is detected, the gesture recognition system localizes the hand and tracks it in a video sequence (see figure 4). Motion and posture attributes are obtained from the sequence [2], and are the inputs the DBN models. The probability of each model given the observation sequence is computed, and the model (gesture) with highest probability is selected. The current system can recognize 9 types of gestures with a precision of over 90%.

Portable devices, such as PDAs and cell phones, are another alternative to communicate with service robots. These are particularly useful when the robot is out of sight. Based on a PDA, we developed a simple and effective interface to command mobile robots. In the screen a reduced map of the environment or the robots view is displayed. The user can command the robot to go to certain location by: (i) pointing to a location in the map, or (ii) by using the control keys (forward, backward, left, right) based on the view of the environment from the robot camera.

8 Applications

We have applied our proposed architecture to two service robot applications: a delivery robot and a host robot.

The delivery robot's goal is to accept a message for a particular person, search for this person and deliver the message. The robot uses the perception module to locate person. It then uses its natural language module to receive a message in natural language and the name of the person to whom this message should be delivered. The robot moves through its environment using its navigation capabilities and its human localization module to find persons. It then uses again its natural language interface module and asks the human for his/her name. In case it matches the target's name it delivers the message, otherwise it continues searching for persons in the environment.

The host robot goal is to provide information about a particular place to visitors, in our case, information is provided about a University and its research group. This robot also requires the map building, localization and navigation modules, the human recognition module, the natural language interface, and uses the gesture recognition module. Once a person is recognized, the robot starts a dialog with the human to give information about a particular place. The robot is able to move to designated places of interest using its navigation module even in the presence of new obstacles. It can then provide information about that particular place and can also project previously stored videos. The user is also able to direct the robot to particular places using arm movements and the gesture recognition module.

The messenger robot has been tested successfully in a real environment. We have tested independently the different modules for the host robot, which are currently being integrated. New modules can be added or interchanged to create new host robots with different capabilities. For instance, a new application can be created just by changing the dialogs of the natural language module.

9 Summary and Future Work

A general framework for creating service robots has been described. The architecture uses an MDP framework and a set of independent modules that can be incorporated and interchange for different applications.

There are several research directions that we are planning to follow, creating other modules and improve existing ones. We are planning to incorporate a module that can be used to teach the robot by example. We also plan to combine the gesture recognition and natural language modules to enhance our human-robot interface. Finally, we want to improve our coordinator so that it can deal with conflicting situations that can arise when two modules require the same resources at the same time.

References

[1] R.C. Arkin. *Behavior-based Robotics.* MIT Press, 1998.

[2] H. Aviles and L.E. Sucar. Recognizing similar gestures. In *Proc. IEEE Inter. Conf. on Pattern Recognition (ICPR)*, China, August 2006.

[3] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI '00)*, 2000.

[4] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI '98)*, Madison, Wisconsin, July 1998.

[5] T.G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[6] P. Elinas, L.E. Sucar, J. Hoey, and A. Reyes. A decision-theoretic approach for task coordination in mobile robots. In *Proc. IEEE Robot Human Interaction (Ro-Man)*, Japan, September 2004.

[7] S. Hernández-Alamilla. Navegación de un robot móvil en ambientes interiores usando marcas naturales del ambiente. Master's thesis, ITESM - Campus Cuernavaca, Mexico, 2005.

[8] S. Hernández-Alamilla and E.F. Morales. Global localization of mobile robots for indoor environments using natural landmarks. In *Proceedings IEEE Conference on Robotics, Automation and Mechatronics (RAM-2006)*, 2006.

[9] V.M. Jacquez-Leal. Construcción de mapas y localización simultánea con robots móviles. Master's thesis, ITESM - Campus Cuernavaca, Mexico, 2005.

[10] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2), 1998.

[11] M. Martínez and L.E. Sucar. Learning an optimal naive bayes classifier. In *Proc. IEEE Inter. Conf. on Pattern Recognition (ICPR)*, China, August 2006.

[12] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI '02)*, Edmonton, Canada, 2002.

[13] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, NY., 1994.

[14] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second-generation museum tour-guide robot. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, May 1999.