# A Blackboard Based Implementation of a Mobile Robot Architecture

**Mauricio Matamoros, Jesús Savage**
Bio-Robotics Laboratory
Universidad Nacional Autónoma de México

### Abstract

This paper describes the software that controls the autonomous service robot called Justina, built in the laboratory of Bio-robotics at the National University of Mexico. The robot is based on the ViRbot architecture for autonomous mobile robots operation. ViRbot defines a robotics system that consist on several modules that, in overall, operates a mobile robot. those systems are a human-robot interaction interface based on Natural Language Processing, for understanding voice and gesture commands, Conceptual Dependence, Action Planning, based on the sensor's information and the state of the robot, and a set of Subtasks, performed by a Simple Task Planner which coordinates several modules that process information from the sensors and controls the hardware. For simplifying task planning along with real time awareness, all modules communicate with each other through a central module called Blackboard, which supports shared variables, with publisher/subscriber pattern, and message passing.

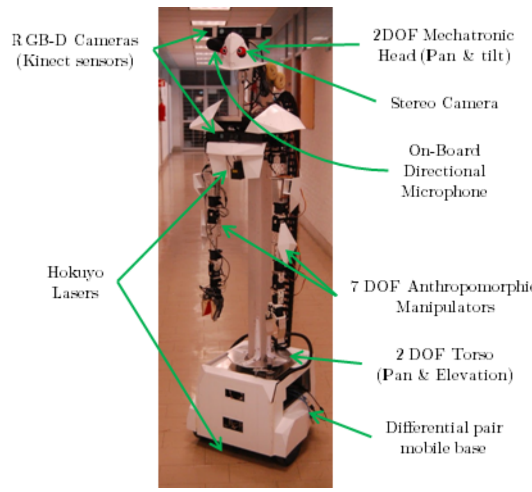



Figure 1: Robot Justina

## 1 Introduction

Service robots are hardware and software systems that can assist humans to perform daily tasks in complex environments. To achieve this, a service robot has to be capable of understanding commands from humans, avoiding static and dynamic obstacles while navigating in known and unknown environments, recognizing and manipulating objects and performing other several tasks that the human beings ask for.

The main objective of the ViRbot architecture (Savage et al. 2008), is to operate autonomous robots that can carry out daily service jobs in houses, offices and factories. This system has been tested in the last six years in the RoboCup competition at the categories @Home and will be used again in the competition in Eindhoven, The Netherlands in 2013, with robot Justina (see Figure 1).

Robot Justina integrates the work of several research areas, such as expert systems, computer vision, action, path and motion planning, robot localization, arm control and place conceptualization and natural language understanding.



Section 2 is an overview of the ViRbot Architecture; this system provides a platform for the design and development of robot Justina's software. Section 3 is about the implementation of ViRbot Architecture as a set of software modules that perform the control of the hardware and a Blackboard for the common data and communication management. In Section 4 we present the conclusions and the future work.

## 2 ViRbot: A System for the Operation of Mobile Robots

In this system, the operation of a mobile robot is divided in several subsystems, as shown in figure 2. Each subsystem has a specific function that contributes to the final operation of the robot. Some of this layers will be described in this section.

**Simulator.** The ViRbot system contains a simulator that provides simulated values of the internal and external sensors implemented in the robot. Also the simulation of new sensors can be incorporated easily. This simulator allows to design and test new algorithms and, using optimization techniques, improve their performance.

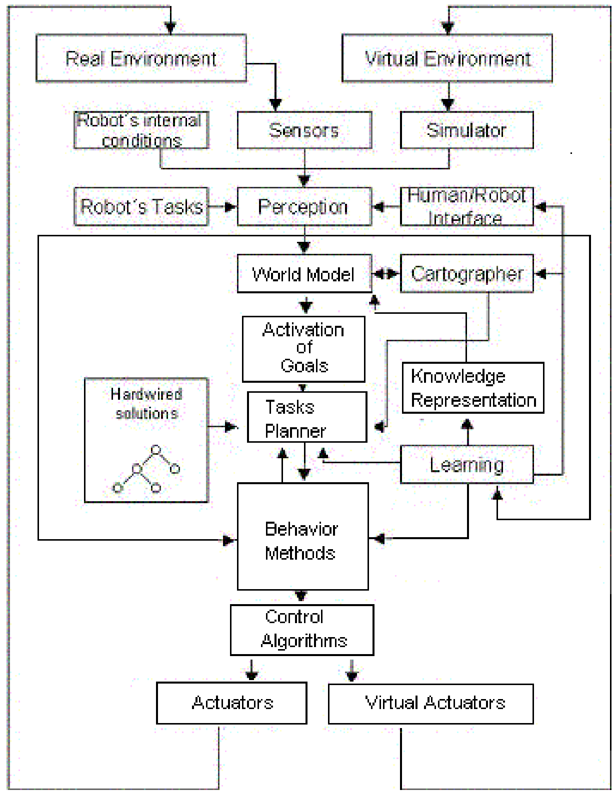



Figure 2: ViRbot System Architecture

**Robot Internal Conditions.** This module provides information about the robot internal state through temperature, encoders, battery charge, inclinometer sensors, etc.

**Sensors.** This module provides information from the external world where the robot interacts. It has laser, sonar and infrared sensors as well as video and RGB-D cameras.

**Robot's Tasks.** The set of tasks and subtasks that the robot can accomplish.

**Human/Robot Interface.** The purpose of this subsystem is to recognize and process the voice and gesture commands. It is divided in three modules:

- **Speech Recognition.** Using digital processing algorithms, it analyses voice commands given to the robot.
- **Natural Language Understanding.** This module is used to find a symbolic representation of spoken commands given to the robot using the recognized sentences coupled with Conceptual Dependency techniques.
- **Conceptual Dependency.** The Conceptual Dependency (CD) theory was developed by Roger Schank (Schank 1975) for representing meaning. This technique finds the structure and meaning of a sentence in a single step. One of the main advantages of CDs is that they allow rule based systems to make inferences from a natural language system in the same way humans do. CDs facilitate the use of inference rules because many inferences are already contained in the representation itself. The CD representation uses conceptual primitives, not the actual words contained in the sentence. These primitives represent thoughts, actions, and the relationships between them. CD structures facilitate the inference process by reducing a large number of possible inputs into a small number of actions. This makes CD suitable for the representation of commands or simple questions without a strict sentence grammar, but they are not suitable for the representation of complex sentences.

**Perception.** With the information provided by the Robot's internal conditions, sensors, Robot's tasks and Human/Robot Interface modules the Perception module generates beliefs of the state of the environment of the robot.

**World Model and Activation of Goals** The belief generated by the perception module are validated by this module, it uses the knowledge representation module, thus a situation recognition is created. Given a situation recognition, a set of goals are activated in order to solve it.

**Cartographer** This module has different types of maps for the representation of the environment like:

- **Raw maps.** These are obtained by detecting the position of the obstacles using the robot's laser sensors.
- **Symbolic maps.** These contain all the known obstacles as polygons defined by their vertexes.
- **Topological and Probabilistic /Hidden Markov Model) maps.** which contains useful information about the environment (Thrun and Bücken 1996).

**Knowledge Representation** A rule based system is used to represent the robot's knowledge, in which each rule contains the encoded knowledge of an expert.

**Planner** The objective of the action planning is to find a sequence of physical operations to achieve the desired goal generated by the the World Model module. The Planner module has two subsystems: the Action Planner and the Movements Planner which are explained in the following sections.

**Hardwired Solutions** A set of hardwired procedures represented by state machines are used to partially solve specific problems, including movement, object manipulation, etc.

**Navigation** This module controls the robot's movement to follow a path through the environment. This specified path is given by the Motion Planner module(Llarena et al. 2012).

**Behavior Methods** A set of reactive algorithms is used to solve problems not foreseen by the planner, like avoiding unknown obstacles.

**Control Algorithms and Real and Virtual Actuators** Control algorithms, like PID, are used to control the operation of the virtual and real actuators. The virtual and the real robot receive the commands and execute them by interacting with the virtual or real environment and with the user.

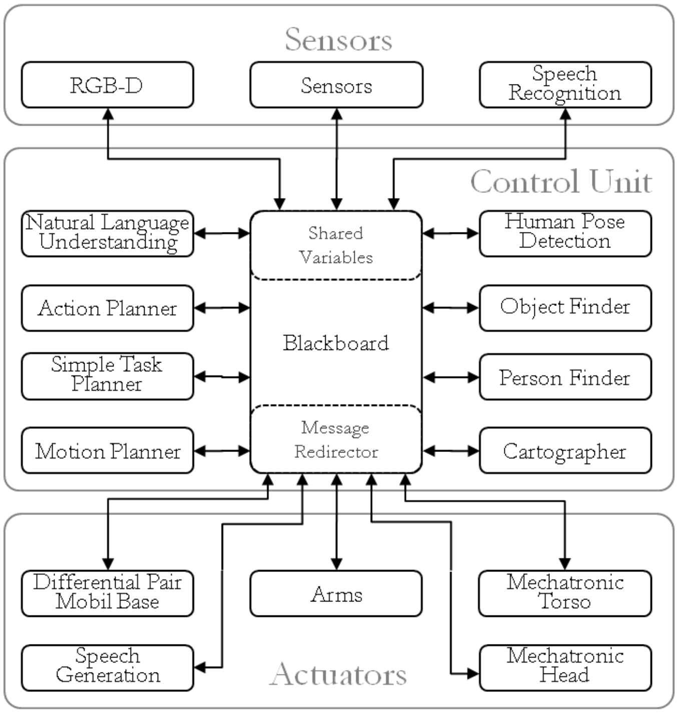



Figure 3: Blackboard structure

**Learning** The following learning algorithms are used for the robot:

1. *Map building.-* Cluster sets are used to locate and represent obstacles and free space in the environment.
2. *Self Localization.-* Hidden Markov Models using Vector Quantization & Self Associative Neural Networks are used to estimate the robot's position and orientation.
3. *Behaviours.-* Genetic Algorithms are used to learn new behaviours.

## 3 ViRbot implementation on Robot Justina

ViRbot is implemented in robot Justina by means of a Blackboard architecture for the communication between modules. This allows to use more than one platform and run software modules, developed in different programming languages, in different computers. Robot Justina uses two computers running both Linux and Microsoft Windows, and its modules are programmed in C++, C# and the rule based language developed by NASA, CLIPS (Riley 1994).

### 3.1 The Blackboard module

A general purpose service robot requires several highly specialized systems which must communicate with each other, therefore, a blackboard-like central module is included: a flexible system to control the transmission of messages between modules, monitor their status and store the common data used by all the modules that integrate the system. All common data is stored in the Blackboard as shared variables to grant access at any time to other modules. It also uses a producer/subscriber pattern to enhance the Real Time Awareness of the system and reduce communication overloads. Also the Blackboard offers a flexible platform to implement ViRbot as shown in figure 3.

While ViRbot determines the operation of the system in a high level of abstraction, it is irrelevant how their modules are distributed in programs and how they communicate with each other. In practice, several modules were merged or split in order to reduce the development time and enhance performance, taking the Blackboard module as a central message redistribution system and a repository to store common information.

The Blackboard module not only centralizes the communications between modules and stores information. It was also designed to be easily configurable via a XML configuration file which defines: 1) the list of the shared variables of the system (with option to define default value and write permissions) 3) the list of modules of the system, 4) the supported commands of each module, and 5) several sets of actions to take when some events occur. However, opposite to a conventional blackboard, the Blackboard module does not integrate a Control Component to determine access privileges and other behaviours (although one may be added as a plug-in). Also, the data is stored as is, with no congruency nor data-type validation which may be a setback but also offers flexibility, allowing any kind of data to be allowed.

Unless access rights are defined in the configuration file, all modules have both read and write access to all the shared variables. Shared variables may also be created at runtime, in which case the creator module defines the access to that shared variable. For message passing things change a little. The blackboard organizes it in command/response pairs (not al commands need to generate a response but all responses are due to a command). When a command is received, the Blackboard redirects it to the target module, marks the target module as *busy* and denies other command requests for it until the response arrives, there is a time-out for the command execution, or the module disconnects from blackboard. Also there are high-priority commands which are always redirected. The reception of a response for those commands which require one is granted, even if the target module halts or is unavailable, the Blackboard module sends a failure response.

Finally, the Blackboard has helped to reduce the development and debugging time, making easier to localize faults. Also it helped to loose-couple the modules, and making the test over distributed environments almost trivial.

### 3.2 Planning modules

The planning of the actions which must perform a general purpose mobile service robot in order to accomplish a task can be, in most cases, split into various levels of abstraction in order to simplify planning and keep the software maintainability. Therefore, three modules are used for planning: The Action planner for high-level task planning, the Simple Task Planner for low level task planning and the Motion Planner which focuses on the spatial-planning of the movements to be performed by the robot.

**Action Planner** For the Action Planner, the rule-based expert system CLIPS, is used. With CLIPS it is easy to implement tasks of several subsystems of ViRbot Architecture such as the Robot's Tasks, Perception and Knowledge Representation subsystems. The Action Planner works in the highest level of abstraction, coordinating the tasks that the robot must perform and choosing the adequate behaviour in each situation.

The Action Planner has several rule definitions setting the Robot's Tasks, a list of goals that can achieve the robot.

When a command is received, the Natural Language module generates a conceptual dependency primitive that in turn a set of goals are activated.

With this command generated, it performs the Activation of Goals to achieve of the given command. CLIPS has an inference engine and this uses forward state-space search, that finds a sequence of steps that leads to a solution given a particular problem, making a plan to solve job. Also CLIPS is an expert system that can handle the Knowledge Representation in the form of facts inferred from the definition of the rules and operations executed.

Also has the task of the Perception sub-module, which constantly monitors the the state of all other modules and sensors and analyzes the symbolic representation of the data to achieve real time environment awareness.

**Simple Task Planner** In the ViRbot system, the Hardwired Solutions involve simple and repetitive tasks which requires little or no planning but several transformations and can be easily achieved by state machines (like look for a person/object or grasp an object). Those tasks are performed by the Simple Task Planner module. Also this module incorporates reactive behaviours which allows the robot to operate with very limited awareness if the Action Planner is not available.

**Motion Planner** The motion planner is responsible for finding the best sequence of movements to reach the final destination given by the Action Planner or the Simple Task Planner combining classic and modern techniques. It uses Cartographer's maps to calculate the path planning. In parallel to the geometrical representation, it generates a topological representation of the environment and, by Dijkstra algorithm (LaValle 2006), finds the optimal path between the current and goal positions. Obstacle avoidance is achieved using Potential Fields and Finite State Machine based behaviours.

## 3.3 Vision Subsystem modules

This subsystem consist of five modules: the RGB-D module, Person Finder module, the Object Finder module and the Human Pose Detection module.

**RGB-D** This module uses the Kinect Microsoft Xbox 360$^{TM}$(Microsoft 2010), which features an RGB camera and a depth sensor. This sensor enables the robot to see objects in three dimensions, which is essential for object manipulation and environment navigation. The data acquired from the Kinect sensor is stored in the Blackboard as a RGB bitmap, depth bitmap and human skeleton array to be used by the other vision modules such as the Object Finder, the Person Finder and the Human Pose Detector.


(a) Visual identification of objects

(b) Objects are segmentated by depth to reduce compute time and enhance accuracy

Figure 5: Object recognition and location using depth sensors

**Human Pose Detection** The Human Pose Detection Module uses data from the Kinect module to retrieve the detected human skeletons which are converted to Orthogonal Direction Change Chain Codes (Figueroa et al. 2012) which are used as input for a Hidden Markov Model (Rabiner 1989), with a state grammar to detect, recognize and label human movements like stand-up, sit-down, walk-run, etc. (see Figure 4).

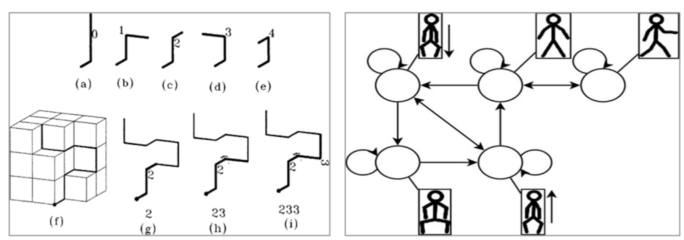
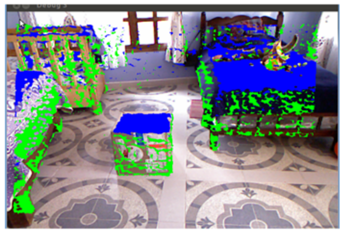



Figure 4: Human skeletons are processed in a state grammar.

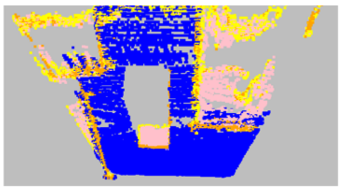
**Object Finder** The Object Finder Module consumes data from the RGB-D module and features a robust implementation of an object tracker where objects are segmented using receptive fields (Ekvall and Kragic 2005; Chen, Chen, and Chien 2009; Chang and Krumm 1999), and represented by feature points which are described in a multi-resolution framework, that gives a representation of the points in different scales (see Figure 5).

Detection and description of interest points are based on the SIFT (Scale-Invariant Feature Transform) algorithm

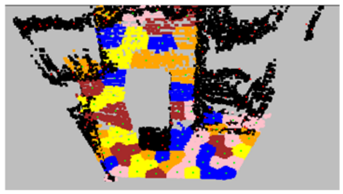

(a) Envoronment



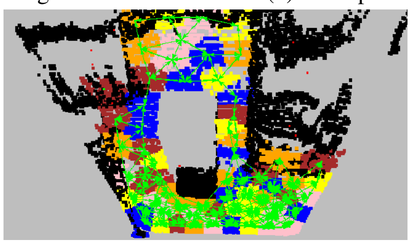
(b) Plane detection



(c) Navigatable area



(d) Free space clusterization



(e) Topological map

Figure 6: Creation of the topological map of the environment using 3D sensor.

(Lowe 2003) after a first segmentation by depth, see figure 5a. After an object recognition, the geometry of the object is computed from the depth map to fetch the object's centroid and orientation(Figueroa et al. 2011).

**Person Finder** The Person Finder Module uses a proprietary software: the VeriLook SDK, for multiple face detection and recognition. During the training phase, the pattern of the detected face is associated with a name an is stored on a local database within the module. In the recognition phase, the detected face pattern is checked against all the known patterns; if the confidence is high enough is considered as a match for the known pattern. The name associated to the detected faces, if known, and it's confidence is stored in the Blackboard.

**Cartographer** The Cartographer module generates the several maps used by the motion planer and is responsible of storing the map information into the blackboard (such as the name of the locations and places, the location of the detected objects, etc.) so it can be used to task planning. It also generates topological maps with the 3D points obtained from the depth map of the Kinect module as is shown in Figure 6.

**Speech Recognition** This module is part of the Human/Robot Interface ViRbot subsystem and uses the MS SAPI 5.3 with a set of robust grammars for the recognition of the expected voice commands on each task. An array of hypothesized strings with its confidences is stored in the Blackboard to let the Action Planner to choose the most adequate candidate for the robot's current context.

**Natural Language Understanding** Recognized sentences are represented using conceptual dependency primitives. For instance, if the recognized sentence was either "Robot, give the newspaper to Jhon" or "Robot, bring me the newspaper" (the command was given by Jhon) the same Conceptual dependency primitive is generated even when both sentences have only two words in common. This primitive is:

```
( ATRANS (ACTOR Robot) (object newspaper)
 (from newspaper_place) (to Jhon))
```

With the Conceptual dependency primitive representation the inference engine finds the set of actions to accomplish the command given to the robot.

**Speech Generation** This module is part of the ViRbot's Human/Robot Interface and allows the robot to interact with humans by synthesizing voice. It uses the Microsoft SAPI 5.3 along with the Loquendo Susan voice (Microsoft 2011).

### 3.4 Hardware interface and control modules

**Differential Pair Mobile Base** This module controls and interfaces with the mobile base of the robot. The robot uses a differential pair mobile base to navigate with a micro-controller-based board which sends and receives commands for motor control and encoder reading, using a serial communication interface.

**Arms** This module controls and interfaces with the two arms of the robot. Each arm is a 7 DOF manipulator with anthropomorphic configuration built with Dynamixel servomotors and a micro-controller based embedded system for control and trajectory planning. The arm has an anthropomorphic design in order to perform a better obstacle avoidance and natural human-like movements. For path tracking, a PID plus gravity compensator control is used and a vision based control is implemented for handling objects.

**Torso** This module controls and interfaces with the Mechatronic Torso of the robot. The torso has 2 DOF for control the elevation and pan of the arms and head. The torso increases the grasping range of the manipulators and allow the robot to look straight to both, small and tall humans.

**Sensors** The Sensors module controls the data acquisition from several sensors like the lasers. The acquired data is stored in the Blackboard to be used by other modules. The robot has several sensors for getting information on the surrounding environment: laser range finder for motion planning and obstacle avoidance, a Kinect system for seeking humans and objects, a stereo VGA camera for pattern recognition and a directional microphone for natural-speech command interpretation. Digital signal processing techniques are applied to the obtained signals to interact with the dynamic environment.

**Mechatronic Head** This module controls and interfaces with the Mechatronic Head of the robot. The Head design is based on the corresponding movements of the human head with 2 degrees of freedom: pan and tilt. This freedom of movement allows to point the sensors to obtain accurate readings of the environment and perform a systematic search in a particular zone of interest. It carries three different sensing devices on it: a Kinect system, a directional microphone and a stereo camera. The sensors are integrated with a friendly plastic face providing confidence to humans that interact with the robot.

## 4 Conclusions

The ViRbot system has been tested on several robots since 2006 and it allowed the Team Pumas to achieve the third place in the Robocup@Home category in Atlanta in 2007. In these years, the full system has been improved having reliable performance and showing promising results. New algorithms for the recognition of human faces and objects, along with localization, 3D mapping and more complex behaviours arbitrated by the Action Planner's expert system are constantly tested, improved and redesied. As the software which controls the robot has grown bigger an several modules has been splited due to specialization, the blackboard module has proved to be a powerful ally by reducing the development time while keeping the system maintainability and robustness.

To improve navigation, SLAM techniques are being developed using the visual relationship between two different views of the same scene. Also, to provide compatibility with the most used framework for robots, a bridge between the Blackboard and ROS is being developed.

Some videos showing the operation of the ViRbot system can be seen at `http://biorobotics.fi-p.unam.mx`.

## 5 Acknowledgment

This work was supported by PAPIIT-DGAPA UNAM under Grant IN-107609

## References

Chang, P., and Krumm, J. 1999. Object recognition with color cooccurrence histogram. In *Proceedings of CVPR '99*.

Chen, T.-W.; Chen, Y.-L.; and Chien, S.-Y. 2009. Fast image segmentation and texture feature extraction for image retrieval. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 854 –861.

Ekvall, S., and Kragic, D. 2005. Receptive field cooccurrence histograms for object detection. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 84 – 89.

Figueroa, J.; Contreras, L.; Pacheco, A.; and Savage, J. 2011. Development of an object recognition and location system using the microsoft kinecttm sensor. In *RoboCup*, 440–449.

Figueroa, J.; Savage, J.; Bribiesca, E.; and Succar, E. 2012. Recognition of static gestures using three-dimensional chain codes using dominant direction vectors. In *Intelligent Environments (Workshops)*, 231–241.

LaValle, S. M. 2006. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. Available at http://planning.cs.uiuc.edu/.

Llarena, A.; Savage, J.; Morales, Á. F. K.; and Escalante-Ramírez, B. 2012. Odometry-based viterbi localization with artificial neural networks and laser range finders for mobile robots. *Journal of Intelligent and Robotic Systems* 66(1-2):75–109.

Lowe, D. G. 2003. Distinctive image features from scale-invariant keypoints.

Microsoft. 2010. Kinect - xbox.com - xbox.com. `http://www.xbox.com/en-US/kinect`.

Microsoft. 2011. Speech server - microsoft corporation. `http://www.microsoft.com/speech/speech2007/default.mspx`.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257 –286.

Riley, G. 1994. *CLIPS Reference Manual Version 6.0. Technical Report Number JSC-25012.* Software Technology Branch, Lyndon B. Johnson Space Center, Houston, TX.

Savage, J.; Weitzenfeld, A.; Ayala, F.; and Cuellar, S. 2008. The use of scripts based on conceptual dependency primitives for the operation of service mobile robots. RoboCup 2008 Symposium, Suzhou, China.

Schank, R. C. 1975. *Conceptual Information Processing*. North-Holland Publishing Company.

Thrun, S., and Bücken, A. 1996. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*.