Map-Making with a Four-Legged Mobile Robot

Benjamin H. Willard and Kurt D. Krebsbach
Department of Mathematics
Lawrence University
Appleton, WI 54912-0599
{benjamin.h.willard, kurt.krebsbach}@lawrence.edu

Abstract

Most approaches to map-making with an autonomous mobile robot rely on a set of standard tools, such as laser range finders, sonar arrays, precise shaft encoders, or GPS uplinks. While it is difficult to construct a map without such tools, there is still utility in endowing low-cost, legged robots with map-making capabilities in order to exploit their unique advantages. This paper describes an attempt to construct a recognizable map of an indoor hallway environment with a Sony ERS-7 AIBO outfitted with suboptimal mapping components.

Introduction

Many map-making studies (Thrun, Burgard, & Fox 2000; Stewart *et al.* 2003), have employed laser or sonar range finders on a wheeled robot, but map-making with a robot lacking these standard features has received relatively little attention. The Sony AIBO is one such modern robot. The AIBO is an excellent robot platform for undergraduate research because of its low cost and robust feature set, but the AIBO's sensors are less than ideal for the construction of *good* maps; i.e., maps that mirror the physical features of an environment to within an acceptable level of precision and accuracy. A good map can also be used for navigating and localizing within an environment.

While simply choosing a robot that is better suited to mapping would eliminate many difficulties, there are still good reasons to focus on mapping capability for suboptimally outfitted robots. For instance, because the AIBO is used so widely at the undergraduate level, a good mapping system for the AIBO would benefit a number of undergraduate projects that could use it as a starting point for more involved projects. The AIBO mapping problem is also interesting because of the challenge posed by legged locomotion. Legs allow options for movement that wheels cannot easily provide such as traversing rough terrain or climbing trees (www.msl.ri.cmu.edu/projects/rise/) but legs also introduce noisy and inaccurate odometry. Minimizing error in leg-based odometry is an interesting subproblem that arises from map-making with the AIBO, and is useful because legged robots can perform important navigational tasks that cannot be performed by wheeled robots.

In this paper we describe an undergraduate student project that attempts to construct a good map of an indoor environment using only the standard features of a Sony ERS-7 AIBO. Specifically, we are programming the AIBO to autonomously construct a map of an enclosed hallway area and later use that map to navigate to specific points in the hallway. We are currently trying to accomplish this task without giving the robot explicit knowledge of general hallway structure such as corridors and junctions.

The AIBO Platform for Mapping

Gathering useful odometry and range data from the ERS-7 sensors is a difficult and problematic task. Instead of wheels, the AIBO has four legs – legs that introduce a host of problems for determining odometry accurately. The AIBO's legs are quite prone to slippage, especially when rotating. The legs complicate the measurement of rotational distance when turning, and they make it difficult to maintain an accurate position estimate for the robot.

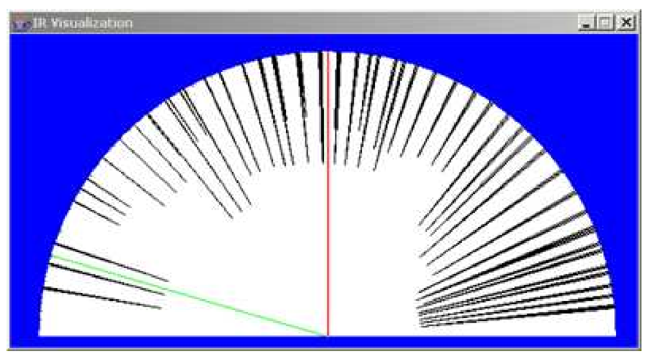

Figure 1: The infrared view module displays data collected from the most recent pan of the robot's head. Lines are drawn from the point at which an obstacle is detected to the bounds of the sensor range to better visualize the robot's perception of its environment.

To gather range data, the AIBO must rely on two of its three infrared sensors. Each of the infrared sensors is tuned for a specific distance range. The short-range sensor detects

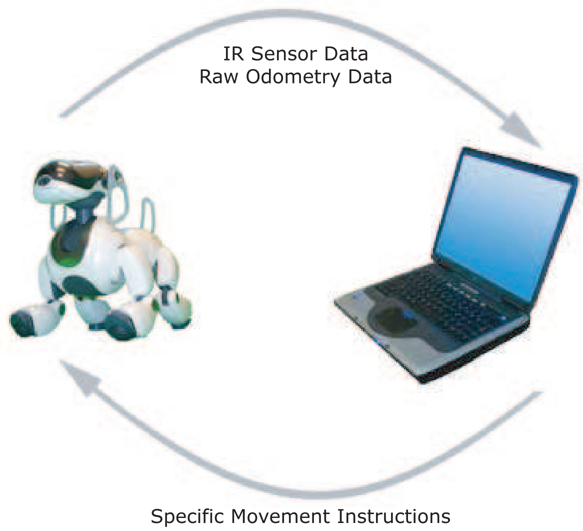

Figure 2: The robot sends sensor information to the laptop, and the laptop plans the robot's actions.

obstacles from 50mm to 500mm and the long-range sensor detects obstacles from 200mm to 1500mm. The chest mounted infrared sensor is used to detect sudden drops such as stairs. The short- and long-range sensors are located on the AIBO's head and face directly forward. To acquire a 180-degree view of the environment in front of the robot, it is necessary to pan the head from side to side. This method of gathering range data is much different from the use of fixed sensors such as sonar, as it is impossible to acquire a broad view of the environment at one specific time. By panning the head, it is possible to create a snapshot view of the environment represented as distances to obstacles, but each point is viewed at a slightly different time since the head cannot rotate instantly. If the robot is walking forward and panning its head to acquire range data at the same time, the problem only worsens. Figure 1 shows a graphical representation of an infrared snapshot, as constructed by our program.

Mapping Framework

The mapping framework developed during this project is composed of two subsystems, as depicted in Figure 2. One subsystem runs on the AIBO itself, and the other operates on a more powerful laptop. The two subsystems communicate via a wireless network link. The AIBO-based subsystem is built in C++ with the help of the Tekkotsu development framework for AIBO robots created at Carnegie Mellon University (Tira-Thompson 2004). The laptop-based subsystem is written in Java for maximum portability.

The Tekkotsu-based code running on the AIBO is responsible for controlling the robot's limbs, recording the speed of the robot, and collecting IR data by panning the head and reading from the IR sensors. The slave program running on the robot receives all higher level commands from the program on the laptop. For instance, the robot itself does not decide where to move. Instead, the robot sends sensor information to the laptop which then plans the actions the robot should take. Then, the laptop transmits instructions to the robot via the wireless network. The robot executes all instructions it receives from the laptop, and the cycle continues.

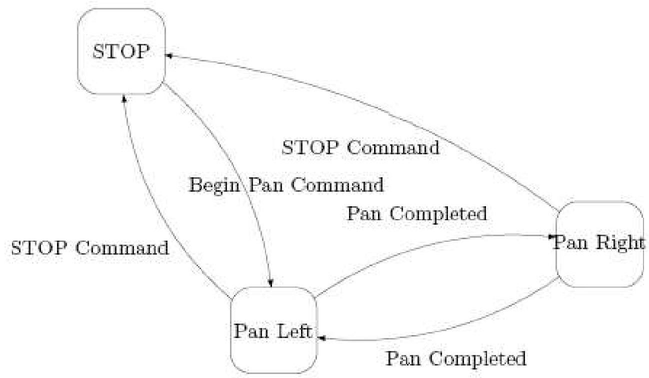

Figure 3: The Tekkotsu state machine that drives the robot's constant head panning is quite simple. A command from the laptop-based controller activates the panning which will continue until a STOP command is sent by the laptop.

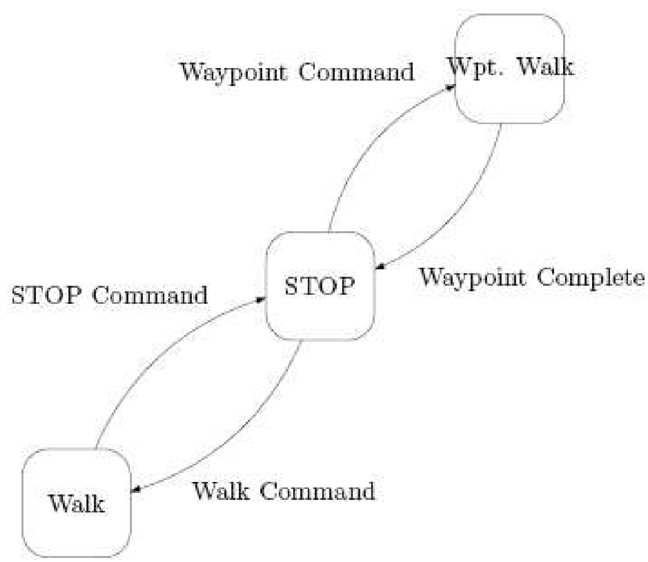

Figure 4: Locomotion is driven by this Tekkotsu state machine. A command from the laptop-based controller either sets the robot on a course specified by a Tekkotsu waypoint or instructs the robot to begin walking in a particular direction until a STOP command is sent.

The Tekkotsu program is based on two state machines. One state machine is responsible for controlling head movement and the other is responsible for controlling locomotion. Both state machines are embedded within a Tekkotsu *behavior* that receives commands from the network link and passes them on to the appropriate state machine. Boxes represent states, while arcs represent state transitions based on sensor readings or commands sent over the network. Figures 3 and 4 illustrate the two state machines. The modular nature of this setup makes the Tekkotsu portion of the framework fluid and easily expandable. It is a simple matter to add nodes to the state machines or even to add an entirely new state machine to a behavior.

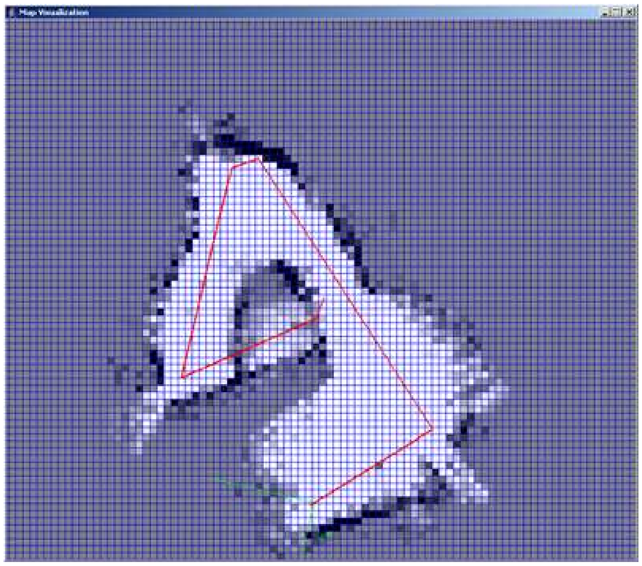

Figure 5: The occupancy grid module builds a map as the robot wanders through the environment. Darker squares indicate a greater likelihood that the area is occupied by an obstacle or a wall.

The Java-based controller's main function is to plan the robot's high-level actions. We designed the core of this program to accommodate the easy addition and removal of features. The unchanging role of the controller is to establish a connection with the robot and to interpret sensor data that is gathered from the robot. It is possible to add modules that can access the sensor data from the robot and send commands back to the robot. Currently, the Java controller has three distinct modules: a view of the IR sensor data from one sweep of the robot's head (Figure 1), an occupancy grid-based map (Figure 5), and a control module that uses information from the other two modules in order to steer the robot down the hallway. This framework provides a useful foundation for constructing a usable map in a hallway environment.

Project Status

The current map-making process uses two important types of information from the AIBO, namely the instantaneous velocity of the robot and infrared range values. The robot's velocity is tracked by the Tekkotsu framework and is accessible in the Tekkotsu WorldState data structure. The infrared sensors are also easily accessible from Tekkotsu, but it is necessary to pan the robots head continually to maintain an up-to-date snapshot of the environment. As sensor data reaches the laptop, it is fed to the occupancy grid module.

This occupancy grid is based on a model presented in Robin Murphy's text (Murphy 2000). Murphy's sensor model is specifically intended for a sonar array, so it was necessary to eliminate sonar-specific assumptions to adapt the model to work with the IR sensors. The IR model differs from the sonar model because it is essentially a one dimensional line instead of a cone-shaped field. Specifically, the occupancy grid uses the idea that each grid unit has an associated value which indicates the likelihood of occupancy. For each obstacle that is detected within a particular grid unit, the occupancy value is incremented. Likewise, all of the units that lie on a straight line between the robot and the detected obstacle are assumed to be empty, and occupancy values are decremented. The result of this process is shown in Figure 5.

We are currently refining our occupancy grid-based mapping process. The AIBO's infrared sensors are accurate enough for our purposes in a hallway environment, but the legged nature of the odometry has proved far too inaccurate for a simple dead reckoning approach. At a minimum it is necessary to fuse the data from the IR sensors with the odometry data to verify the robot's movements within the environment by confirming the change in positions of observed obstacles and walls.

A notable failed experiment involved fusing data from the AIBO's video camera and infrared sensors in an effort to improve the quality of our occupancy grid. Inspired loosely by (Se, Lowe, & Little 2002), we attempted to use distinct landmarks to nail down particular locations in the environment. The robot was responsible for discovering the landmarks visually, with no initial landmark knowledge. Upon sighting a landmark, the robot would determine the location of the landmark using IR sensors and fix the location on the map. Then, if the robot ever happened to see that particular landmark again, it would compare the previously recorded location of the landmark with the currently observed location. A variety of different methods were tested to smooth out the error on the occupancy grid after the same landmark was observed in different locations, but no attempt proved satisfactory. The main problems with the approach were that the robot had difficulty distinguishing the unique landmarks, and no suitable method was devised to fix the grid upon discovery of a discrepancy between old and new landmark positions. Although we do not intend to revisit this specific approach, we may revisit the idea of fusing sensor data from the video camera, particularly if the infrared sensors cannot provide enough data to accurately represent the environment.

Related Work

Many combinations of platforms and environments for robotic map-making have been investigated, and a great number of general and situation-specific methods for creating and refining maps have been developed. Studies using wheeled robots with laser or sonar range finders and probabilistic approaches such as Bayes or Kalman filters have produced impressive results both in and outside of hallway environments (Fox *et al.* 2003). The Sony AIBO itself has been used extensively at the undergraduate and graduate level for many AI and robotics projects. The four-legged division of the RoboCup soccer league might be the most well known venue where AIBO robots have been used for complicated navigation and localization tasks (Veloso *et al.* 1998).

Past and Future Work

Lawrence University currently owns one ERS-7 AIBO that has been affectionately named LARRY. LARRY provides students interested in AI and robotics with a platform to build projects and conduct experiments. The AIBO was purchased with a grant to support several students interested in learning more about robotics and extending their knowledge from the *Artificial Intelligence* class at Lawrence. So far, LARRY has been at the center of two large student projects. The first project, entitled *Multi-Robot Navigation and Coordination* (Dan Casner and Ben Willard), made use of two AIBO robots (one student owned and one owned by the university) that communicated and coordinated their efforts to rendezvous in a landmark-rich environment. The second project, *Map-Making with a Four-Legged Mobile Robot*, is the subject of this paper. This project is split into two phases. The first phase occurred as summer research, and the second phase takes the form of an Honors Project over the course of the 2006-2007 academic year. The summer was used for literature review, experimentation, and the construction of a development framework for mapping. We are currently extending the mapping framework and refining elements of the mapping process.

Between the extremely error-prone odometry and the noisy, range-limited IR sensors, forming a good map-making strategy presents many challenges. Throughout the remainder of the academic year, we hope to construct a high quality map-making system that will create reasonably good maps of indoor hallway environments. With a map-making framework in place, our short-term goal is to minimize the odometry error produced as the robot walks. Turning introduces more error than walking in a straight line, so we will attempt to refine the way that odometry is computed while turning. Custom calibration of the Tekkotsu walk engine, fusion of data from the odometry and infrared sensors, and perhaps even modification of the leg movements involved in a turn are approaches we are considering for minimizing odometry error.

Acknowledgments

Many thanks to Lawrence Provost David Burrows for supporting this work through a Lawrence *Academic Distinctiveness* grant. Thanks also to Dan Casner for his work in developing the AIBO project at Lawrence, and to Paul Rybski for spending several days advising Lawrence students in their AIBO-based endeavors.

References

Fox, D.; Hightower, J.; Liao, L.; Schulz, D.; and Borriello, G. 2003. Bayesian filtering for location estimation. *IEEE Pervasive Computing* 2(3):24–33.

Murphy, R. R. 2000. *Introduction to AI Robotics*. MIT Press.

Se, S.; Lowe, D.; and Little, J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks.

Stewart, B.; Ko, J.; Fox, D.; and Konolige, K. 2003. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *UAI*, 551–558.

Thrun, S.; Burgard, W.; and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE International Conference on Robotics & Automation*.

Tira-Thompson, E. 2004. Tekkotsu: A rapid development framework for robotics. Master's thesis, Robotics Institute, Carnegie Mellon University.

Veloso, M.; Uther, W.; Fujita, M.; Asada, M.; and Kitano, H. 1998. Playing soccer with legged robots. In *Proceedings of IROS-98, Intelligent Robots and Systems Conference*.