The Sony AIBO:
Using IR For Maze Navigation

Kyle W. Lawton and Liz Shrecengost
The goal of the project was to allow the AIBO to autonomously navigate and map an unknown maze.
Outline

- The Sony AIBO
- Tekkotsu
- Our Project
- Conclusion
The Sony AIBO

AIBO stands for Artificial Intelligence roBOt. It also means “companion” in Japanese. The first-generation AIBO was launched in 1999.
### Comments

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>384 MHz MIPS Processor</td>
<td></td>
</tr>
<tr>
<td>32 MB RAM</td>
<td></td>
</tr>
<tr>
<td>802.11b Wireless Ethernet</td>
<td></td>
</tr>
<tr>
<td>Memory Stick Reader/Writer</td>
<td></td>
</tr>
<tr>
<td>20 joints</td>
<td>18 PID joints with force sensing</td>
</tr>
<tr>
<td></td>
<td>2 boolean joints</td>
</tr>
<tr>
<td>9 LEDs</td>
<td></td>
</tr>
<tr>
<td>Video Camera</td>
<td>Field of view 57.6° wide and 47.8° high</td>
</tr>
<tr>
<td></td>
<td>Resolutions: 208x160, 104x80, 52x40</td>
</tr>
<tr>
<td></td>
<td>Up to 25 frames per second</td>
</tr>
<tr>
<td>Stereo Microphones</td>
<td></td>
</tr>
<tr>
<td>IR Distance Measure</td>
<td></td>
</tr>
<tr>
<td>X, Y, and Z accelerometers</td>
<td></td>
</tr>
<tr>
<td>4 Buttons</td>
<td>2 pressure sensitive, 2 boolean</td>
</tr>
<tr>
<td>Sensor updates every 32 ms</td>
<td>4 sample per update</td>
</tr>
</tbody>
</table>
Tekkotsu

- An open source program created at Carnegie Mellon University
- Handles routine tasks and allows the user to concentrate on their unique application
- Designed to make adding new functionality easy
Pertinent Abilities of the AIBO

- **Pre-programmed Walk**
  - Walking is an extremely complicated process

- **Infrared Sensors**
Infrared

- The AIBO measures distance based on how long it takes IR light to get back to it.
- It can only measure things within a very short range (no closer than 100 mm and no further than 900 mm).
- Measurements are taken every 32 milliseconds.
Maze Generation

- Virtual maps were generated to simulate the process of exploration.
- As the AIBO actually moves, it stores which walls it actually sees and updates its own map.
Maze Navigation

- Unexplored cells are preferred to explored ones.
- When the AIBO reaches a dead end, it is able to back-track.
The robot is very likely to go off course as it is traveling through the maze.

Alignment comes in two steps:

- Position to the center of the path
- Orientation parallel to walls
Processing Alignment

- Pan head to get the distances and angles of walls
- Use this information to determine your relative position in the maze
- The picture to the left represents actual data from a maze
Integration

- The program uses a finite state automaton to transfer between the different behaviors.
- Each state is called when the previous one has completed its motion.
Results

- The maze navigation uses a grid, but the AIBO is not confined to move likewise.
- The preprogrammed walk had to be recalibrated in order to account for the different surfaces that it was walking on.
- It is able to explore a maze and account for commonly occurring anomalies.
What Else Could Be Done

- Adapt for different maze types
  - Different wall thicknesses
  - Curved walls
- More efficient navigation
  - Removing excessive stops
  - Panning head while walking
  - Cutting corners
The real world is very different from our maze.

Accounting for errors is critical for robust behaviors.

Real World Applications

- Building Navigation
- Search and Rescue
Thanks to Advisor Ethan Tira-Thompson and TA Jack Shi!
We would also like to thank Dr. David S. Touretzky for the use of the equipment and The Robotics Education Laboratory for the use of the maze walls.