S.P.O.T.  
Smart Path-planning 
Object-avoidance Technology  

Steve Chaya, Rachna Jethwani, Vivek Parmar, 
Dan Rubin, Brandon Sheffer (Team Lead)
Project Overview

- Current Problem
  - Sony’s AIBO can be remotely controlled using only primitive commands (Move forward, backward, left, right)

- Project Aim
  - Improve control by ensuring the robot navigates autonomously to a specified object
Project Description

- Create an Internet based GUI that will allow the user to
  - View the video feed
  - Make the AIBO move towards an object by clicking on it
- Use object detection, obstacle avoidance, and path planning to ensure that AIBO reaches specified object
Purpose

- The AIBO can be used to remotely monitor the house
  - Checking up on the kids
  - Checking if the house has been broken into
  - Catching your wife cheating on you with the milkman

- A framework for algorithms
  - Facilitate future robotic development
  - Eliminate navigational pitfalls of obstacle avoidance and path planning
System Evolution

- Environment Mapping
  - Have the AIBO render a 3D version of its environment for navigation to objects outside its field of view.

- Person Locator
  - Combine person recognition with Environment Mapping to make the AIBO find people.

- Extensibility
  - Possible to extend the S.P.O.T. Framework to robots other than the AIBO.
Application Components

- Client
- Server
- Robot
Client

- Interface for users to control the AIBO
- Four major functions
  - Provide streaming video and display object selection
  - Initiate movement towards the selected object
  - Allow for manual movement
  - Display status information
- Minimal system requirements
S.P.O.T. Server

- The bridge which connects the AIBO and the Client

- Three major functions
  - Process video from the AIBO
  - Process user input from client
  - Plan path to target

- Must be capable of running Apache and MATLAB
S.P.O.T. E.R.

- S.P.O.T. External Robot
- Two major functions
  - Provide the video stream
  - Navigate the environment
- SONY ER7 AIBO with the URBI Framework
URBI

- “URBI is a Universal Real-time Behavior Interface and gives you a simple but powerful way to control any robot or complex system, using a convenient and easy to use scripting language that can be interfaced with several popular programming languages (C++, Java, Matlab,...) and OS (Windows, Mac OSX, Linux).”

**URBI Forge** <http://www.urbiforge.com>
Webots Simulator

“Fast prototyping and simulation of mobile robots”

Cyberbotics <http://www.cyberbotics.com>
User Interface
Image Processing

1. **Send Parameters**
   - Movement Request
   - Target Parameters
   - Image, Input Parameters

2. **Process Image**
   - Video Stream

3. **Web Server**
   - Video Stream
   - URBI Commands
   - Image, Input Parameters

4. **Client**
   - S.P.O.T. Server
   - S.P.O.T.

5. **AIBO**
   - URBI Command
   - Video Stream

6. **MATLAB**
   - URBI Commands
Command Generation

Client

Web Server

MATLAB

AIBO

大海 & 发送 URBI 命令

发送距离
Command Execution

The diagram illustrates the architecture and command execution process. It shows the flow of information and commands between the Client, Web Server, MATLAB, AIBO, URBI, and S.P.O.T. Server and S.P.O.T. System.

1. **Client**:
   - S.P.O.T. Server
   - S.P.O.T.

2. **Web Server**:
   - Target Parameters
   - Video Stream
   - Movement Request
   - Video Stream
   - URBI Commands

3. **MATLAB**:
   - Image, Input Parameters

4. **AIBO**:
   - URBI Command
   - Video Stream
   - Format & Forward Command

The process involves receiving requests, executing commands, and forwarding the formatted commands to AIBO for execution, all coordinated by the S.P.O.T. Server and System.
Stage 1 – Analysis

- What currently exists?
- What are limitations of existing software?
- How can we make it better?
Stage 2 – Requirement Specification

- Web Interface
  - Remote Control
- Web Server
  - Image Processing
  - Obstacle Avoidance
  - Path Planning
- URBI Framework
  - AIBO Communication

Waterfall Process Model
Process Model

- Remaining Stages
  - Design
  - Implementation
  - Testing
  - Maintenance
Project Plan (Stage 1)
Web Interface Design

- Web Interface
  - Ensure remote access
  - Video stream capture for processing
  - Settings control
  - Primitive movement commands
- Unit Analysis Testing
Project Plan (Stage 2)
Object Recognition and Localization

- Image Processing
  - Retrieve mouse coordinates for object selection
  - Create MATLAB routine to generate distance
  - Use IR sensors to verify distance
- Unit Analysis Testing
Project Plan (Stage 3)
Path Planning and Obstacle Avoidance

• Create Path Planning and Obstacle Avoidance Algorithms
  • Design first step routine so the AIBO moves initially in the direction of the selected object
  • Keep track of displacement
  • Use IR to range objects that may be in the way
  • Use Image Processing to do obstacle avoidance

• Unit Analysis Testing
Project Plan (Stage 4)

Project Completion and Testing

- Assemble pieces to complete the product
  - Unit Analysis Testing as each stage is installed
- Full Testing Analysis
- Final Presentation
- Extra features as time permits
Questions?