FRI Project Proposal

Smitha Nagar, Mayuri Raja, and Xinyu Zhao

April 2, 2017

1 Introduction

1.1 Problem

Due to the complexity of the BWI bot's planning algorithms, or, while driving the bot manually, the necessity of having a hand on the keyboard while operating it, both human and robot must move slowly through the building – slow enough that the human may daydream, lose focus, and fall asleep standing. To remedy this, we propose taking advantage of the robot's ability to play sounds, and allowing the robot to play ambient music based on its environment. This will keep the human driver more focused (maybe even pumped up and dancing), and will make the experience of working on the robot more enjoyable.

We hope that another happy side effect will be that the people in the same building as the BWI bot will regard its presence more positively, in the way that children are excited when they hear the music of an ice cream truck.

1.2 Previous Experience

- **Smitha:** My experience has been with this class - the turtlesim project was my introduction to ROS, and the image and vision processing project for homework 5 was how I learned more about ROS and OpenCV.
- **Mayuri:** My experience is limited to what I have learned in this class through doing the projects and readings. I learned about OpenCV with the image processing project, and I also learned more about how nodes communicate with each other. I have also attended several talks on computer vision, and although these talks were higher level with fewer implementation details, they have lended some intuition as to how to detect objects in an image and filter colors.
- **Xinyu:** In high school, I was part of a robotics competition team; I helped with implementing vision processing/tracking and movement (controlling the motors). My recent experience has been with Turtlesim (simple pathfinding algorithms) and vision processing on the BWI bot, like the others have mentioned.

2 Proposed Approach

Our goal is to use the robot's sound-playing, vision, and localization capabilities in tandem to get the robot to play music based on its environment.

The first essential step to achieve this purpose is to ensure that the robot can **analyze its environment**. This has two components:

- Vision: We will use what we have learned about computer vision, specifically two dimensional vision with the cameras, to break the video feed down into its component colors. Then, using the color makeup of the image, we will determine what mood the image has. If the image consists of primarily warm colors, then the robot will play happier songs with faster beats and major keys. If the image

consists of cold colors, then the robots will play sadder songs with slower tempos and minor keys. Using the OpenCV library, we will use color filtering to calculate the percent of warm colors and cold colors in the image.

- Localization via proximity sensors: We will process the point-cloud data from the robot's proximity sensors to determine the size of the room the robot is in, as well as the presence of doors nearby. If the room that the robot is in is small, i.e. the sensors detect walls on all sides in close proximity, then the robot is most likely in an office. If there are only two walls on either side, then the robot is most likely in a hallway. Doors can be detected by looking for "breaks" in nearby walls.

Second, we will use the robot's location to play **location-based music.** This will not apply to all locations, of course, and will rely on the robot having been localized correctly. We will use the current navigation system and the robot's current knowledge of offices, bathrooms, and elevator locations to play incidental music at each of these locations. Location-based music will take precedence over image-based music. This relies on the robot's ability to differentiate between different rooms as offices and classrooms, so we will add clarification between locations to the robot's map if necessary.

Third, to **play music**, we will play MIDI files either from a database of files stored on the robot or, optimally, from a dynamically generated MIDI file that the robot generates based on its environment. For playing music on the robot, we will have to ask the mentors and TAs about the hardware available to us. If the robot has external speakers, then we will use the ROS audio_play or sound_play nodes to output the sound files to the speakers. If we are using a database of music, then we will pre-program "mood" attributes for each song, and the robot will pick one in the appropriate category to play based on environmental factors. We will also store a database of songs to play based on location.

Fourth, to **generate music**, we will first examine hand-picked collection of ambient MIDI files and divide these into different categories based on the "mood" they they best fit. Using this, we can use a Markov process of generating notes and sounds based on the environment. Ideally, when determining what sounds to play next, the program will use its surroundings to determine what "mood" of sound to play and then generate the next note of the Markov chain from the provided ambient MIDI files. We will provide MIDI files with single notes and chords so that there is some variation in what music the robot creates. CFugue is a C++ library that can be used to play notes and generate MIDI files, so we will possibly use this library for music generation.

3 Evaluation of Success

As we work through our project according to the timeline and learn more about what each aspect of our project entails, we will better understand what is feasible to accomplish within our given deadline. We will measure our success according to the following:

Basic Viability: At the very least, the robot will play assigned songs according to the colors in its environment. When surrounded by cooler colors, the robot will play a sad song, and when in warmer colors, a happy one. Determination of "cool" and "warm" environments will be based on vision processing and image thresholding. It will switch between tracks after a song completes or once the environment changes, whichever comes first. The robots color detection and warm-cool determination will work best at extreme instances of warm and cool colors and will provide different results each time for more moderately-colored environments.

Moderate Viability: This will cover basic viability, as well as location-based (incidental) music. For example, when the robot is near a bathroom, it will play a "flush" sound, and when it is in an office, it will play office music. This will involve potentially adding recognizable locations to the robot's map (such as the bathroom) and using localization to select the appropriate sound to play. The robots color detection will be better honed and will reliably produce the same verdict (warm or cool) for the same environment each time it passes through.

Semi-Advanced Viability: This will cover moderate viability, along with procedurally generated music. Using a collection of one-note MIDI files, the robot will generate music using individual notes to form major or minor chords in an appropriate key and octave according to how the environment is classified. Success of the music-generating algorithm will be judged based on recording the robots generated music and backwards-analyzing the music with the same algorithm we use to generate the music to determine if the robots generated music was sufficiently happy or sad. We will also judge its aesthetic quality by comparing to the happy and sad songs we have selected for the music database.

Advanced Viability: After reaching the criteria for semi-advanced viability, we will, time permitting, improve our music generation algorithm to produce music that is pleasing to hear, rather than just a random sequence of notes. Success of this stage will be harder to judge, since it is more of an aesthetic bar rather than a numerical one. We will bring in outside opinions to determine whether the music is actually pleasing to hear.

4 Timeline

- April 3rd:
  - Submit project proposal.
  - Verify viability of approach with instructor and mentors.
- April 4th:
  - Start researching how to play sounds on robot.
  - Talk to mentors and TAs in lab about hardware available for playing sound on robot.
- April 7th:
  - Start code base.
  - Start working on image processing and color segmentation of image.
- April 12th:
  - Finish image processing. Robot should accurately classify different environments as warm/cool.
  - Start curating sounds/songs and port into robot program.
- April 17th:
  - Have robot play music while running. Test that robot can play music from files stored on the computer and can accurately classify environments.
- April 18th:
  - Research how to gather and process environment data (proximity sensors, identifying specific locations.)
  - Start coding logic for incidental music.
- April 27th:
  - Finish location/incidental music. Test on robot in the lab.
  - Research MIDI generation and Markov chains.
  - Begin implementing environment-based music generation.
- May 1st:

  - If time permits, tweak procedural music generation algorithm to make the music cohesive and melodic.

- May 8th:
  - Evaluate success and do final debugging.
  - Begin writing final report.
- May 11th:
  - Submit final report.
  - Begin preparing for presentation.
- May 13th:
  - Give presentation.