SHaZam the Magic Lamp: IR-Based Gaze Tracking and Light Direction

Chaim Halbert, Dexter Scobee, and Edward Zhao
EE149A/EE249A Project Milestone Update

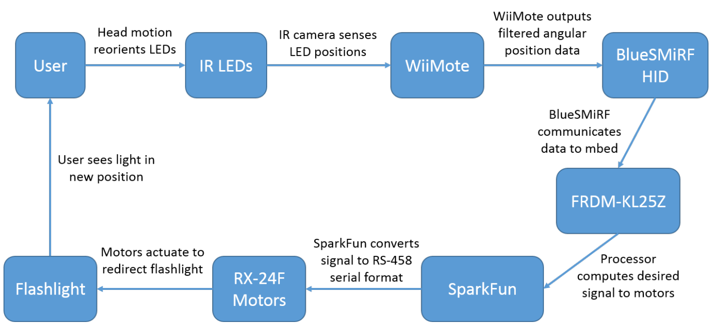

Fig. 1: SHaZam ConOps. This diagram represents the interconnections between components for the SHaZam system

I. Project Vision

The goal of this project is to design a lamp that will redirect its light to follow a user's gaze. The magic lamp will consist of a standard adjustable desk lamp with a microcontroller embedded in its base and servo motors connected to its light source. Additionally, the lamp will utilize a WiiMote to track IR LEDs affixed to the user's face or hat. The WiiMote will communicate relative positioning data to the microcontroller via Bluetooth, which will in turn direct the motors to aim the light source. The controller will behave according to a state-machine that models states such as "Track" or "Stay." Figure 1 shows the logical connections between system components and how they are intended to interact.

II. Project Milestones

A. Hardware

The microcontroller to be used for this project will be the mbed FRDM-KL25Z Freedom board. All team members possess their own board, and all have successfully demonstrated the ability to build and execute programs for the microcontroller in their own development environments, which will allow for parallelized development.

Originally, we were going to use a BlueSMiRF Gold to connect to the WiiMote. However, the Gold only connects via serial, while the WiiMote requires an HID interface. To that end, we have purchased a BlueSMiRF HID that has arrived today. In case this does not work:

1. We also have a Raspberry Pi, USB Bluetooth dongle, and an Alamode shield for servo control.
2. We could also buy a USB mini OTG cable to interface the USB dongle with the mbed.

To date, we have been able to connect the WiiMote via Bluetooth to a laptop, and we have been able to communicate with the BlueSMiRF Gold via the mbed board. Now with the BlueSMiRF HID in our possesion, we plan to complete the transmission of data from the WiiMote to the mbed via the BlueSMiRF HID.

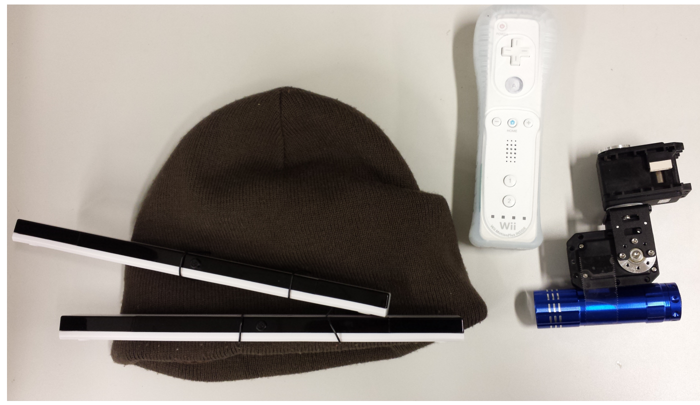

Fig. 2: SHaZam Hardware. From left to right, IR LED assembly for user wear, WiiMote with internal IR camera, Dynamixel 2-axis motor assembly with attached flashlight

For motor actuation, we will be using an assembly of two Dynamixel RX-24F motors to form a two-axis gimbal assembley (see Figure 2). These are "smart servos" which are controlled by sending data via a serial connection. In our application, we will send desired angular positions to the motors via RS-458 (the SparkFun will convert data from the mbed board to an RS-458 signal).

B. Software

We have created Makefiles (tested on Mac) to enable program compilation without internet access, and to allow us to use our own IDEs. We also have hello world programs for:

- Blinking the mbed's LED
- Receiving debugging messages over serial-USB
- Sending and receiving data via serial to the Bluesmirf Gold

We also have the capability to communicate with the WiiMote via Bluetooth connection to a laptop, and we have obtained a "USB2Dynamixel" adapter which allows interfacing a computer to the Dynamixel motors to prototype motor control software. In terms of data filtering and motor positional control software, our peripherals already provide a useful suite of functionalities. The WiiMote performs blob tracking and filtering onboard to provide smooth angular position estimates for the IR LEDs (there actually is no mode in which the WiiMote will transmit raw pixel data). Similarly, the Dynamixel motors are smart servos, so the control logic to maintain a position is internal to the servo and we only need to provide the position to be held.

C. Modeling and Algorithms

We have developed a finite state machine that models the behavior we want for our project, as well as mathematical models detailing our gaze tracking algorithm.

For our state machine, we have two main states, “On” and “Off.” Within the On state, we have two sub state machines, one dedicated to pitch, and one dedicated to yaw, to control the respective servo motors. Inside the pitch and the yaw submachines we have three further substates; “Calibrate,” “Stay,” and “Track”. Figure 3 illustrates the flow between these states within the pitch sub state machine. Within the Calibrate state, we want to standardize our measurements before beginning any tracking. This state will act as an initialization state and will only execute once. Then, depending on the angle of the user’s head, we will either move to and remain in the Stay state or advance through to the Track state, which begins moving the servo motor in accordance to the user’s head movements. We have designed guards to avoid actuating the motors when the user may not intend for his motion to be tracked by SHaZam. To this end, we set minimum and maximum limits on the change in desired lamp orientation (based on change in user head position) to which the system will respond. Note that these are not instantaneous changes, but the change over a fixed, short period of time (on the order of 0.2 seconds).

We have also derived a mathematical model that will allow us to reconstuct the state of the LED configuration (which reveals where the user is looking) by obtaining angular position measurements from the WiiMote.[1] The data provided are x- and y-angle measurements, corresponding to the estimated angular position of the LEDs in two orthogonal planes. Figure 4 shows the geometry of reconstructing the state in one of these measurement planes. Note that the state of the LEDs in a plane has three degrees of freedom (shown in Figure 4 as u, v, and ψ), so three angular measurements are needed to unambiguously reconstruct the state. Once the state in the first plane is known, however, the range data (v) is known for the second plane as well, so only two angular measurements are needed to reconstruct the state relative to the second plane.

III. Path Forward

At this point in the project, we believe that we have all of the basic components to successfully implement our design. With the new BlueSMiRF module in hand, we will move forward with creating and testing the necessary software to collect data from the WiiMote on the mbed processor. In parallel, we will use the USB2Dynamixel to move forward prototyping motor control from a laptop. Once motor control functionality is established, the SparkFun board should enable the same control from the mbed, serving as a means to convert data from the mbed UART into the RS-458 format required by the Dynamixel motors.

[1]This model is an expansion of work done by Johnny Chung Lee (Google, formerly CMU). See http://johnnylee.net/projects/wii/

In addition to continuing to develop our electronics interfaces, we will also iterate on the designs of user-facing components. We will look to create a more comfortable but less mutable fixed LED array for the user to wear so that the distances between LEDs can be better characterized. Finally, the motor assembly will be mounted on a lamp-like frame to enable the desired experience for the user.

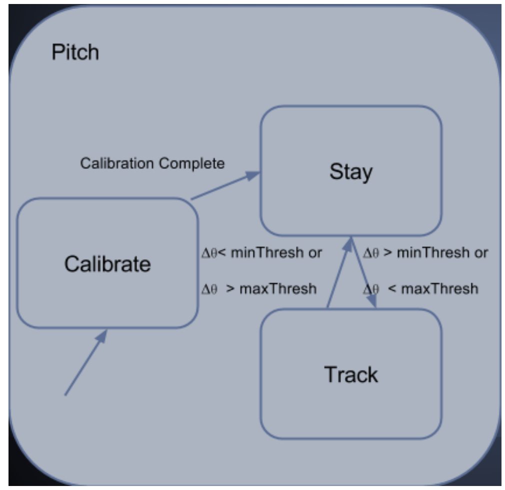

Fig. 3: State Machine. The state machine controlling the motors’ pitch orientation is designed with three states: “Calibration,” “Stay,” and “Track.” Note that there will be thresholds to prevent undesired tracking. The control for yaw follows an identical model

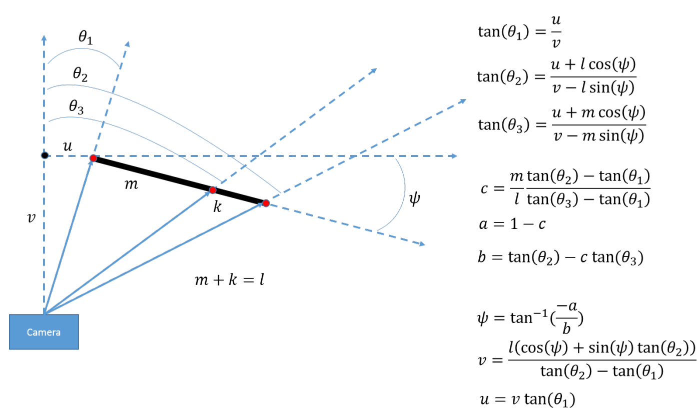

Fig. 4: State Reconstruction. By obtaining angular position measurements of three LEDs with known offsets, it is possible to fully reconstruct the state in a plane. Range data (v) obtained from one plane of measurement can be applied to aid state reconstruction in the other