Bee Translator App Design Document

Overview of Project Idea:

The Science Museum of Virginia wants to improve their beehive exhibit. In order to accomplish this, they have expressed the need for an interactive application. The primary goal of this application is to collect video data of bees waggling and mapping their dance to an actual location of nearby plants. Through this, we hope to educate visitors on the communication between bees and the many uses of AI and Image Analysis.

Implementation of the Project:

To create a functional application that accomplishes this goal, we will need a camera to observe and collect video data on the hive. Our application will provide a simple framework for museum staff and visitors to view and learn about the intricacies of the bee waggle. In addition, the application will informatively walk users through the calculations used for detecting and tracking the bee waggle and show them a sample result of the path of the waggle and location referred to by the bee. We are awaiting clarification on some components of the user interaction.

Project Timeline:

Fall Semester Goals and Accomplishments:

- Requirements gathering
  - Meeting with Jeremy and Elli to learn about the project
  - Meeting with Brandon to ask about the camera setup
  - Project Proposal Document uploaded to Edusourced
  - Researching the bee waggle dance and seeing what other groups have done in the past
- Getting a camera to get sample footage of the hive
  - Sample footage used to check the lighting conditions, getting data about how many pixels the bees will take up, the distance of the camera from the hive, the necessary video resolution, and video frame rate.
- Weekly status reports uploaded to Github
- Researching technologies to begin writing the parts of the app
  - Chelsea and Paul will work on the front end which will include the demo portion of the app
    - Tools used will include Ionic and JavaScript
  - Christopher and Vinit will work on the back end components
    - Tools used will include OpenCV for working with the camera and Python
- Design Document uploaded to Edusourced

Possible Winter Break Goals:

- One large component of our work over winter break will be learning how to use the technologies for the project. We also need to confirm our current project plan with Jeremy and if there are any major changes from that meeting, that will also impact our plan. If we follow the document in its current form, we would work on implementing the modules in the order they are written.

- Create program to initialize and set camera angles and settings (parallax and resizing)
- Vinit and Christopher
    - Start working on the bee detector using OpenCV and Python
- Chelsea and Paul
    - Start creating initial UI (Idle screen and loading screen)

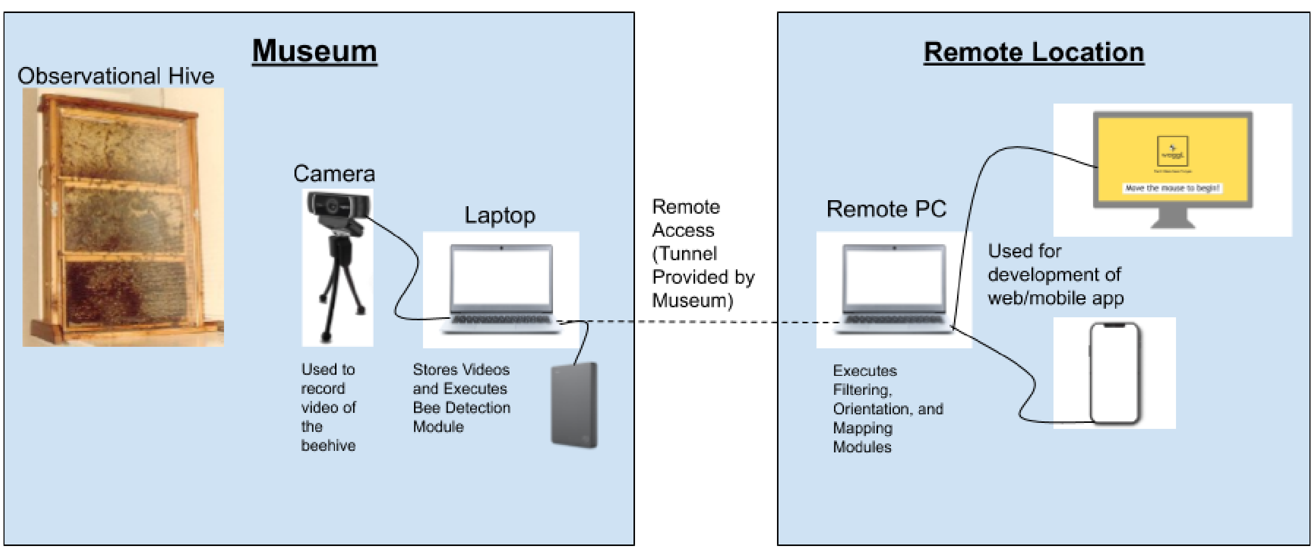
Possible Spring Semester Goals:

- The specifics of this section will depend on what we are able to do over winter break.
- Create a demo program and an interactive user interface
    - The features of the demo program are included in the UI overview section
- Create a prototype of the back end components of the app.
- Create Report
- Create Poster
- Create Presentation

Hardware Overview and Costs:

- What we have ordered:
    - [Camera](#) (currently in Vinit's possession so we can learn how to use it) ($119.99)
    - [Tripod](#) (got lost by VCU, being reordered currently) ($25.99)
- Potential hardware needed:
    - [2TB HDD](#) - Store preprocessed and processed data and video ($110)
    - Laptop or Desktop - Communication between camera, hard drive, and our home computers

Setup:



Software Overview:

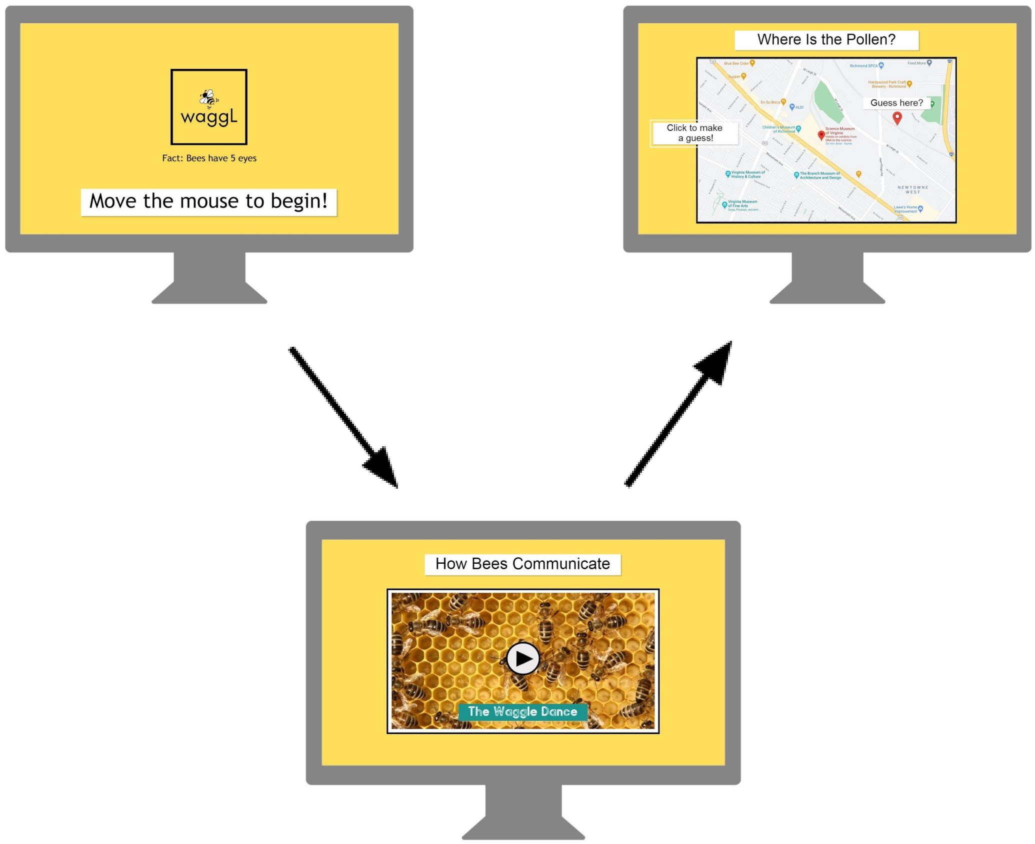
Programming languages: Python, JavaScript

This program will be a web app and it will be presented on a computer.

Possible Software Modules - listed in the order in which they would be developed

- UI Module and Demo/ Information module (Design (app/website?) for educators and visitors to use)
    - Detailed in the UI Overview
    - Ionic Framework - Web App
- Bee Detection Module
    - Use OpenCV to communicate with the camera
    - Using a general detection algorithm to detect if a bee is waggling. Partition the video in ~2 minute sections of video depending on the presence of a bee waggling
    - This could be done either offline or live depending on the storage system we use (laptop vs. desktop vs. external drive, etc.)
- Filtering Module
    - Filter out false positives from the bee detection module snippets
    - This could be done either manually or via a convolution neural network with an Adam optimizer as described in our research
- Orientation Module
    - Orient waggle for mapping module via a Fourier Transform
    - Calculate the position of the sun via the solar azimuth angle calculation
    - Find the position that the bee’s waggling is referring to
- Mapping Module
    - Plot the location that the bee is waggling about onto a static map centered on the museum

UI Overview:
A web app will walk the users through an interactive explanation of the bee's communication. While informative, this app will function primarily to entertain children as they look at the beehive. We plan to use Ionic to be supported on mobile devices as well.



The app's default state will be a splash screen that cycles through different bee facts until the user moves the mouse or clicks the screen. Once the user begins, they are guided through an explanation of the bee's communication dances. Several times, they are prompted to answer questions or interact with the app through an age friendly interface. Pictured above, one of these interactions will feature a guessing game where the user places a pin on a map. Like many other applications in the museum, this gives the users a chance to "leave a mark" on something at the exhibit.

Math Overview:
This document describes the waggle dance in plain english.
When determining how far away a source of food is, we need to know the position of the sun, the time of day, date, length of the waggle run, and angle of the dance relative to the sun
This document describes margins of error regarding the length and angle of the bee corresponding to certain distances and angles away.
This document also describes the correlation between the bee waggle run and the distance away of the food source.

This research paper describes a project similar to ours and is the source for several of the formulas listed below

This paper also helped us decide how several components of the app will work, including our use of the Fourier Transform and Convolution Neural Network

This piece of code helps with calculating the position of the sun given a time of day, location, and date. A library for Python called pvlib.solarposition can do this as well.

Calculates the approximate location of the flowers based on the waggle of the bee:

$$r_R \sim f_d^{-1}(d_w).$$

$$\theta_R \sim atan2\left(\sum_{j=1}^{n} sin\alpha_{wj}, \sum_{j=1}^{n} cos\alpha_{wj}\right), n = 2k.$$

$$p_R \sim \frac{d_w}{d_r}.$$

r_R =: Distance between hive and pollen source
d_w =: Average waggle dance duration
d_r =: Average return run duration
f_d =: Approximation of calibration curve
θ_R =: Conversion factor
p_R =: Ratio between d_w and d_r
α_w =: Average orientation of dance

Calculates a score for bee detection between frames:

$$score\left(\overline{B}_{ij}^{n}, r\right) = \sum_{m=1}^{b}\left(\left(\overline{B}_{ij}^{n}(m)\cdot cos\left(2\pi r\frac{m}{s_r}\right)\right)^2 + \left(\overline{B}_{ij}^{n}(m)\cdot sin\left(2\pi r\frac{m}{s_r}\right)\right)\right. \tag{4}$$

b =: Sliding window width of screen
B_{ij}^{n} =: Vector that stores the last window width of intensity values for each pixel
$\overline{B}_{ij}^{n}$ =: Normalized version of B_{ij}^{n}
r =: Frequencies of the "waggle band"
s_r =: Video sample rate (Hz)

Contacts:
Christopher Flippen
- Email: flippenc@vcu.edu
- Phone: (804) 774-9906

Vinit Patel:
- Email: patelva8@vcu.edu
- Phone: (804) 433-5758

Paul Cochran
- Email: cochranpf@vcu.edu
- Phone: (804) 896-3043

Chelsea Greco
- Email: crgreco@vcu.edu
- Phone: (804) 814-7872