Line Length Detector

Team 6: Alyssa Lessing, Brittany Fields, Jason Follis, Sarah Mathew
Faculty Advisor: Jin-Woo Choi
Instructor: John Scalzo
Louisiana State University Electrical and Computer Engineering Capstone Project

Problem Statement

For students with busy schedules, time is a commodity and lunchtime often becomes an unavoidable time constraint. Due to the large student body and limited dining options in the LSU student Union, long lines are bound to form.
We wish to provide students with an easy-to-use interface that will give them accurate line length/wait times in order to help manage their time. This system will use sensors in order to gather line length information. It will then send data wirelessly and the data will be interpreted and displayed on an accessible interface. This will allow students to make better dining decisions based on the amount of time they have.

Engineering Requirements

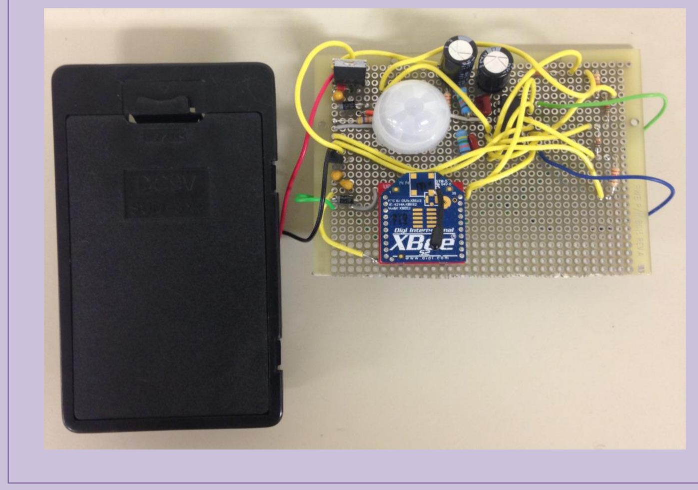
- The system will communicate wirelessly.
- The accuracy of the line length count will be above 80%.
- The power source should be able to last at least one work week.
- Each sensor module should cost less than $150.
- The interface should take less than 3 minutes for a user to acquire information on line length.
- The sensor enclosure should be less than 6 X 6 X 6.

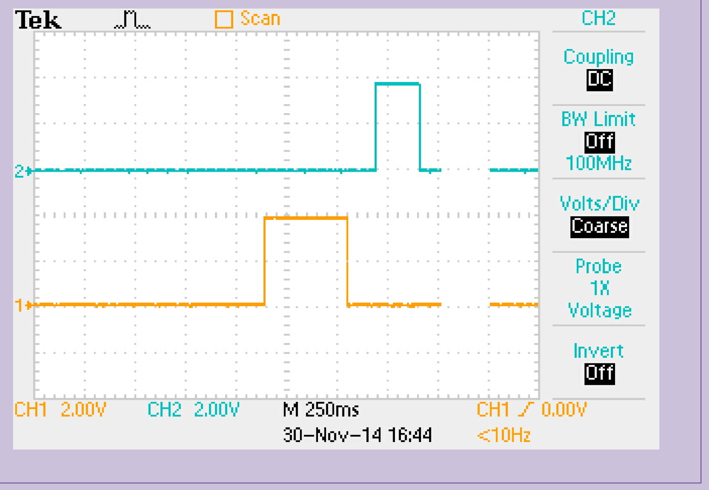
Outcome of Testing and Project

Once our subsystems were performing properly, we wirelessly integrated the subsystems together. We first connected one in line sensor node, then a second, and then one entrance people counter module. With all subsystems integrated, we could test the entire system's performance regarding each requirement.
The outcome of this integrated system showed that all subsystems were able to communicate wirelessly to produce an average wait time and approximate percent full description on the website. During our testing, accuracy was above 80% , for both sensor modules, as well as the wait time and percent calculations. The power sources for two of our subsystems were able to last one workweek. The power supply for the in line sensor and the power supply for the data processing module were both able to last one workweek, while the power supply for the people counter was unable to last the entire duration. Our cost requirements were met, as the in line sensor node, people counter module, and data processing module all cost less than $150 each. The website was programmed to refresh data every 30 seconds, so real-time data was able to be acquired by users in less than 3 minutes. The sensor enclosures were all less than or equal to 6 X 6 X 6 inches cubed in volume; however, the people counter module was unable to fit in a 6 x 6 x 6 enclosure.

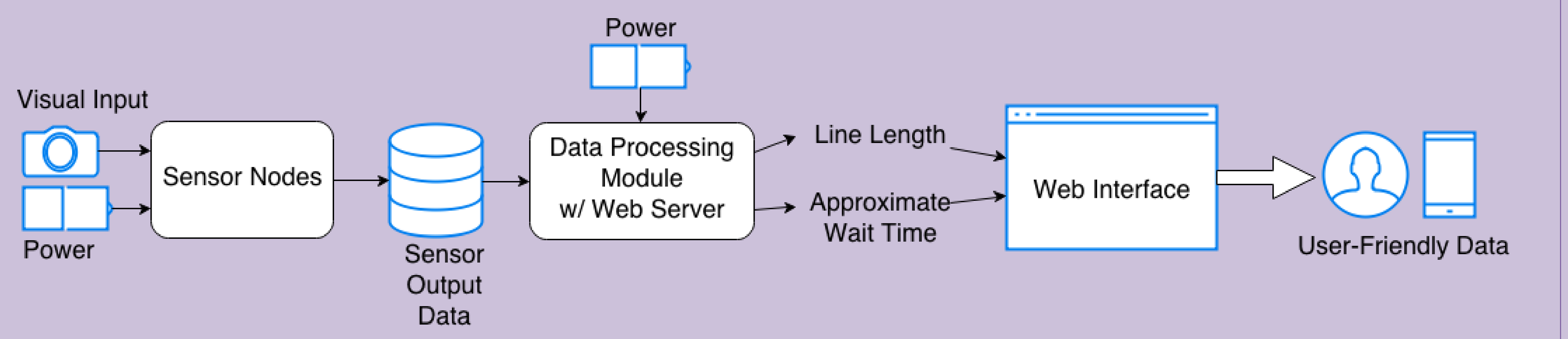
In-Line Sensor Module

The in-line sensor module uses a pyroelectric sensor, which consists of two sensing elements that are arranged side by side in order to cancel temperature variations that are not produced by a human. An infrared filter is placed in front of the dual sensing elements in order to detect a certain range of infrared radiation. When a human walks in either direction, the pyroelectric material will detect the human infrared radiation and a charge will be generated, which will then be converted to a voltage by an FET transistor that is located inside of the sensor. The sensor module incorporates an amplifier stage to amplify the analog output of the infrared sensor, a bandpass filter to eliminate noise, and a comparator stage that converts the analog signal to a digital output. An XBee chip is mounted on each board, which will take the digital output of the circuit and send it wirelessly to the XBee coordinator on the Raspberry Pi. The digital output of the in-line sensors can be seen below. The output signal 1 represents a person walking through the first sensor and the output signal 2 represents the same person walking through the second sensor.





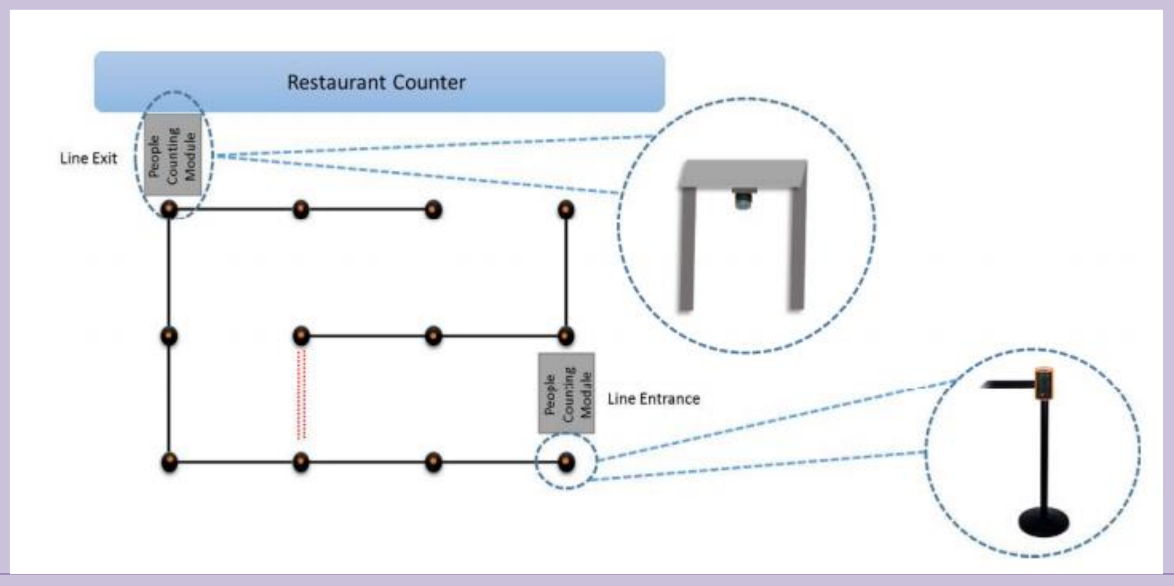
System Behavioral Model



The Line Length Detector gathers visual data from the restaurant lines, using sensors in sensor nodes that are distributed throughout a restaurant line. Each sensor node contains its own power supply, and transmits its output data to one central data processing module. This module has a separate power supply, which is to be plugged into the wall, and processes and analyzes the sensor output data in order to extract line length and approximate wait time for the restaurant line. This data processing module also hosts a web server that displays the data to a webpage. The line length information that is extracted is displayed on an accessible and easy-to-navigate web interface.
The means of communication between the sensor nodes and data processing module is wireless transmission, as stated in our marketing requirements. The data processing module and each sensor node contains a wireless device that transmits and receives data. These sensor nodes communicate using a star topology, with the wireless device in the data processing module performing as a coordinator for wireless communication, and the sensor node wireless devices acting as end devices in the network. The data processing module calculates average wait time, number of people in line, and approximate line length; this information is ultimately displayed on a website.

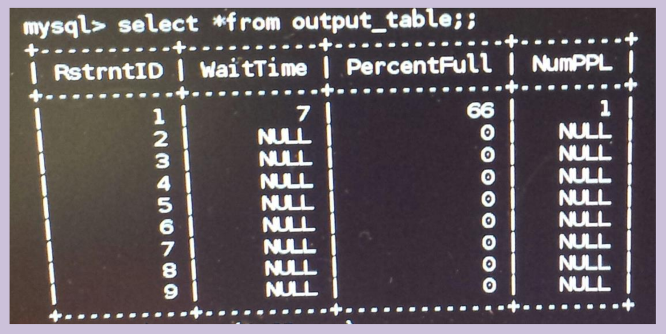
Overall Design Concept

Our Line Length Detector system includes two types of sensing nodes: people counting modules and in-line sensor nodes. People counters are placed at the entrance and exit of the line to detect when people enter or exit the line. The in-line sensor nodes are modules containing bidirectional passive IR sensors, which help to determine an accurate line length percentage. These sensor modules are placed in strategic locations throughout the line on line stanchions. The system uses a data processing module to translate the data that is collected by the two types of sensors and host a website to display the processed data to end users. The entire system communicates using XBee Series 2 radios that are configured for the ZigBee protocol for wireless transmission. An overhead view of our design concept is pictured below.



Data Processing Module

The data processing module is designed to convert raw data from the sensors to user-friendly information that can be displayed on a website. The data processing module consists of a Raspberry Pi, which is connected to an XBee radio via the Universal Asynchronous Receiver/Transmitter (UART) pins on the Pi. The XBee receives transmitted count data and infrared data from the people counter and the infrared sensors respectively and serially communicates this information to the Raspberry Pi microcontroller. The microcontroller then calculates approximate wait time and approximate percentage full line based on the data, which consists of sensor id, its output, and the timestamp associated with each sensor id. The program to calculate the wait time and percent full is written using a combination of C++ and SQL. C++ is used to calculate the wait time and percent full, and SQL provides the connection to the MySQL database using queries. The programs' results are then stored in a MySQL database output table (pictured below). An Apache web server, which is used to host and display a website, is hosted on the Raspberry Pi.



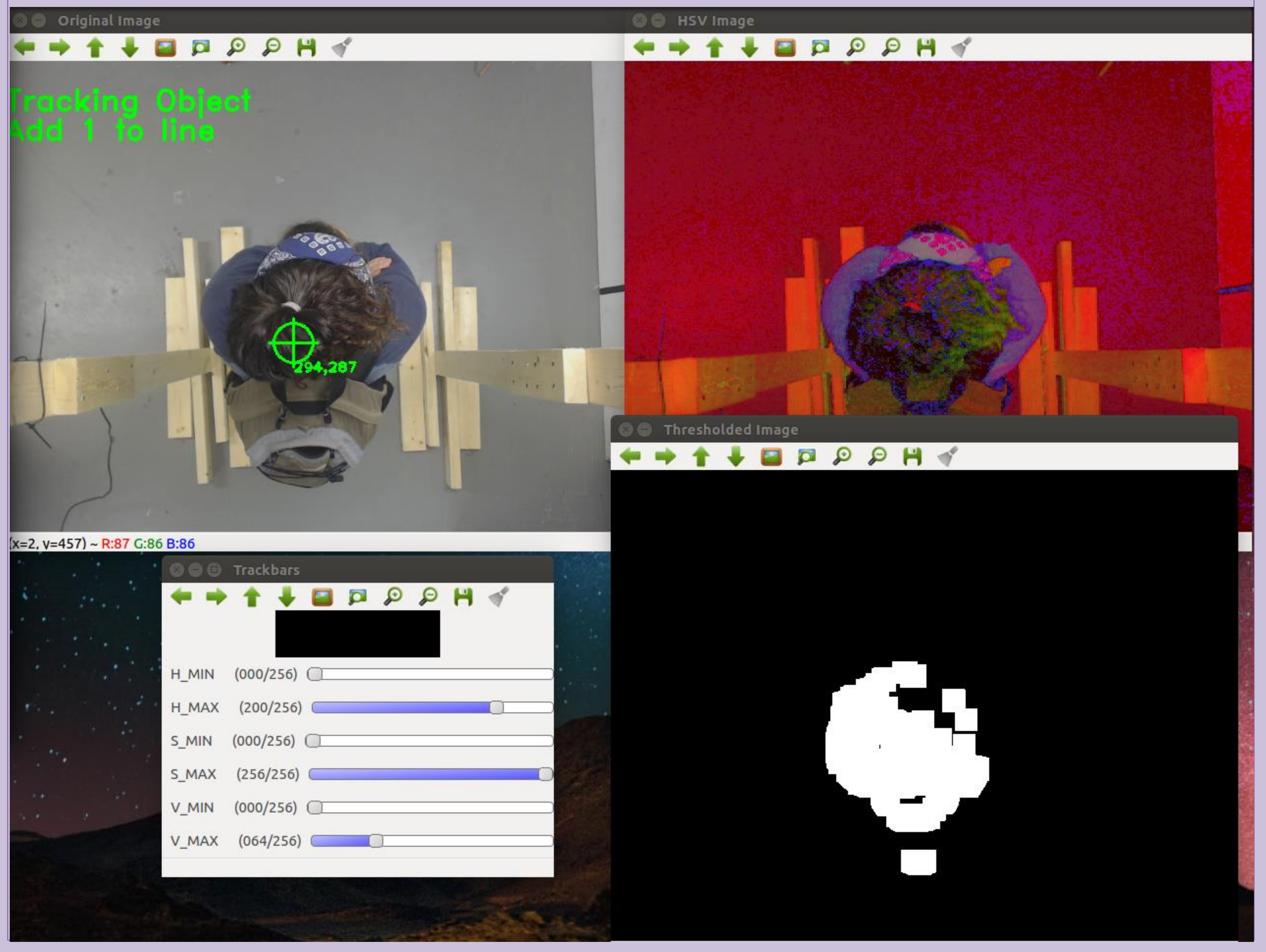
mysql> select *from output_table;;

RstrntID	WaitTime	PercentFull	NumPPL
1	7	66	1
2	NULL	0	NULL
3	NULL	0	NULL
4	NULL	0	NULL
5	NULL	0	NULL
6	NULL	0	NULL
7	NULL	0	NULL
8	NULL	0	NULL
9	NULL	0	NULL

People Counter Module

The people counter module sits at the entrances and exits of the line. As the name suggestions this module counts the total number of people standing within the line at any given time in addition to providing accurate arrival and departure times. After considering multiple alternatives to gather this data a camera mounted looking down on the entrance or exit was determined the best solution. A microprocessor, the BeagleBoneBlack, parses the incoming video stream and outputs a simple binary output to increment the line count. Since all processing is done within the module, video is never transmitted, providing security in addition to low transmission bandwidth needs.

While using an off-the-shelf webcam and microprocessor, the code within the BeagleBoneBlack was written exclusively for this application. Leveraging the power of the OpenCV image library the code performs several transformations. First the RGB color space is converted to HSV and then threshold parameters applied to produce a binary image. This binary image is improved by applying morphological operators, cleaning up the rough edges. From this region, some basic checking is done to determine if it should be considered and if so the centroid is calculated. With an (x,y) coordinate, previous frames are compared with the input frame and a line count is either recorded or not.



Website

The website application for the Line Length Detector is the means for users to access and view the processed data regarding the line length. The web application for the Line Length Detector is written using a combination of HTML, PHP, JavaScript, and CSS. HTML and CSS provides the front end website encoding; PHP allows for dynamic page rendering and also provides the data link between the stored values in the database output table and the display on the webpage; JavaScript is used to display real-time data at an appropriate refresh rate so users don't need to manually refresh the webpages to obtain updated information.
The website has six webpage types available to users: the dashboard, individual restaurant detail webpages, the about page, the map webpage, the login page, and the administrator update page. The dashboard is the homepage of the website. The restaurant details, about, login, administrator update, and map webpages are query-generated pages that can be accessed from the dashboard. Below is shown the a restaurant details page, which shows the line length percentage and the approximate wait time for a particular restaurant.