BU

Senior Design
ENG EC 464

Memo

UPR

To: Professor Pisano

From: UPR Level Security: Juan Peralta, Eugenia Almandoz, Namir Fawaz, Brock Guzman, Brandon Webster

Team: 24

Date: 2/15/17

Subject: Second Deliverable Test Plan

1.0 Standing Person Detection/Tracking

1.1 Goal:

Goal of this part is to show a working algorithm on the computer webcam that is able to detect and track a standing person. A goal for the project is to be able to detect unauthorized personal inside the substation. The idea is that if a worker is going to be inside the substation there is no need for the robot to be navigating, so they can turn it off using the web application. So the robot doesn't need to distinguish between the person it detects, because if it does detect someone, it is safe to assume they are unauthorized. The robot will be continually performing this task while on operation.

1.2 Procedure:

Since the algorithm is for a standing person and the substation layout is pretty open, we can test the code in the hallway with a person more than 4 meters away. This test will be done using my computer with its webcam running. The matlab script will be ran and a video player will be showing the results.

1.3 Result:

The matlab script will run a video player that boxes a detected person and tracks their movement, as well as it shows the amount points that are considered to be features of a standing person. The more points it shows the more confident the algorithm is that it is actually a person.

2.0 WebApp and AWS Hosting

2.2 Goal:

The goal of this section is to have the majority of our website complete and hosted within AWS on Elastic Beanstalk, RDS, and S3. We will be using Elastic Beanstalk for the computation and hosting of our WebApp server. AWS RDS (Relational Database Service) must be functional and working with our database models. Lastly, S3 must be functional with static file hosting. Within our WebApp, there are a number of functions that must work, representing the final goal functions of our WebApp:

1. Login/Logout and Registration: All users must be logged in to access the website, unauthorized users will be redirected to the login screen. Users must be able to register new accounts.
2. Home page must have live video and an easily readable alerts list, along with links to the rest of the pages.
3. Records page must display all alerts within the database, and provide a link to download the related video (we will use stock video until we have finished the function to convert live stream into a video file).
4. The shell of the Remote Control page should be present, although we are not planning on finishing that yet until we have robot movement done.

2.3 Procedure:

We will display all of these functionalities in two ways. The first method will be to show the WebApp being hosted on AWS. We will demonstrate how one would create an account, and show that the pages are inaccessible unless a user is logged in. We will then show all of the individual pages that are listed above, and demonstrate their functionalities.

Next, we will show the app running on a localhost, in order to demonstrate the "video download" functionality. This aspect of the WebApp is currently only partially complete, because we are still in the process of debugging it when hosted on AWS.

2.4 Result:

The WebApp will run on both AWS and a localhost effectively, and will display all of the pages that we have listed above. The "video download" page will not work on AWS, but will work when demonstrated on a localhost. The login functionality will block users that are not signed in, and new users will successfully be created/logged in.

3.0 Navigation

3.1 Goal:

The goal for navigation is to show that we have a way to calculate a bearing for the robot given its current gps location and a goal gps location. In the future this will be used to send commands to the motors of the robot to turn in the direction of the shortest path to the goal and to find the shortest path after detecting an object.

3.2 Procedure:

We will demonstrate that with an input of a latitude and a longitude the program will be able to read the receiver's location and calculate the distance to the goal via shortest path (using a flat earth approximation since this robot will be working within a plane small enough that we don't necessarily need to consider the curve of the planet).

3.3 Result:

The script will print the angle that the robot would need to turn to travel in the shortest path to the goal given its initial orientation is at 90 degrees north, and the flat earth approximation of the distance it will need to travel. If it reads a different satellite that the ones we are looking for it will print "Nothing cool here" to show that the information from those readings are not necessary for these calculations.