BU Memo Senior Design ENG EC 463

To: Professor Pisano, Professor Alshaykh, Professor Kia

From: John McCullough, Shantanu Bobhate, Brian Tan, Bandhan Zishanuzzaman, Teng Zhang

Team: 2

Date: 11/11/2015

Subject: RightAlert First Deliverable Test Plan

**1.0 Recorded Video Object Detection**

**1.1 Description and Goal**

In order to implement the first version of our detection algorithm, we needed footage that would attempt to simulate the actual conditions for our project. We used a stationary laptop camera to film pedestrians walking alone, pedestrians walking side by side, and a biker. The data would then aid in differentiating between bikers and cars. The algorithm uses the OpenCV library. The program begins by taking an initial photo of the background. No moving objects should be in this image. The next frame of a video or live feed is then compared with the background image. The absolute difference of this image is provided to us in a pure black and white pixel image. Each pixel in each subsequent frame is compared with the background frame to see if there is any change. If no change is present, the pixel is black. With a change, there is a white pixel. A “bounding box” is created using the top, left, right, and bottom most white pixels. This box identifies an object. Using the box to extract necessary data, we want to prove that this algorithm can detect the difference between ‘Class A’ and ‘Class B’ objects (Class A being above a threshold width and Class B being below it). The user is provided the option to use an input video or a real time camera. Relevant characteristics regarding an object that crosses a line is output to a text file.

This test demonstrates our ability to both detect and classify objects through the use of image processing. This is directly relevant to our final design. In our final design, we will have to detect moving objects on the road and be able to classify such objects as cars, bicycles, pedestrians, etc. Being able to identify objects properly will allow us to accurately provide drivers turning right at intersections an alert when a biker is present on the driver's right hand side.

**1.2 Procedure**

In order to create the parameters to determine whether an object is classified as type A or B, test footage of objects of both types was recorded. For type A, the larger objects such as cars and trucks, we recorded two of us walking side by side down the hallway. This would represent an object with a width that is much greater than a pedestrian or biker. For type B objects, those that the alerting system would be looking for, we had two sets of videos. The first was of one of the team members walking down the hall and another was of a team member riding a bike down the hall. To obtain the threshold width which separates a type A object from type B, the widths of each of the subjects in the videos was taken. To make sure the collected widths were collected at the same depth of the hallway, a y-value in the frame of the footage was selected. Once the bottom of the bounding box of an object passed this y-value, its width was recorded. A value in between the type A and B width was selected as the threshold width.

**1.3 Verifiable Result**

When the algorithm is run on the previously recorded video, it displays a bounding box around the object and a red line near the bottom of the frame which acts as the y-line. Once the bottom of the object passes the redline, the console will output whether the object i of type A or B class.

**2.0 Live Video Object Detection**

**2.1 Description and Goal**

The live video demonstration will serve as a means to test the parameters that determine object detection. These parameters were determined based on previously recorded video. Based on the success or inaccuracy of the algorithm, the parameters will be altered to decrease the probability of a false detection.

**2.2 Procedure**

In order to display the algorithm’s capabilities, it will have to be tested on a number of different cases. The algorithm will be run on a live video feed from a laptop web camera. The camera will be set up outside of room 109 approximately 5 feet from the door and pointed down the hallway. For the first test, there will be one person walking from down the hall towards the camera. For the second test, two people will come down the hall. Finally, the third test will have a biker ride down the hall.

To begin the algorithm, run it and enter “c” when prompted for the file name. This will target the webcam as input. A window will pop up showing the camera feed. To select where in the frame to detect the moving object, simply click the location with the mouse. The algorithm will output the data into a text file when the object has passed the threshold selected by the user.

**2.3 Verifiable Result**

After running the algorithm on a number of different scenarios we will have written specific data in “output.txt” unique to each test case. Each test will yield its:

* timestamp
* class of object detected (Class A or B)
* width of bounding rectangle
* area of bounding rectangle

The timestamp will help us to differentiate the new tests with older data previously recorded. Depending on the outputted class of object, we will know whether our algorithm responds correctly to our previously set parameters. If our algorithm incorrectly identifies an object, we look to its outputted width of its bounding rectangle. Since the algorithm categorizes objects based on this width, we will know to modify this threshold for more accurate results in the future. In addition, the algorithm uses a “minarea” variable to filter out reflections and shadows which would normally show up up on the video feed. The outputted area of the bounding rectangle from our tests will provide a better threshold for this filtering. By the end of our test, if we see a large amount of false detections we will be able to use the data collected to improve our algorithm.