Senior Design

Automatic Face Tracking Camera

System Test Plan V2.1

07/27/22

Summer 2022

Team 12: All Stars

Nicholas Frazer

Dylan Peters

Hesham Alaidarous

Michael Ankony

Advisor: Paul Watta

1. Introduction

1.1 Purpose

The purpose of this document is to describe the procedures, execution, and reporting of the tests of the automatic face tracking camera. The tests described within this document will serve to test that all set requirements have been met.

1.2 Objectives

The objective of this test plan will be to evaluate the device and its performance to hold it accountable to the defined system requirement specification. This test plan will describe the tests that will be performed to evaluate individual requirements and how they will be executed and recorded. Data will be gathered to accurately determine if the device has met its requirements.

1.3 Scope of the Test Plan

1 Camera FOV
2 Camera FPS
3 Camera detection accuracy
4 Camera rotation range
5 Camera speed
6 Image Aspect Ratio
7 Displaying/storing images

1.4 Test Case Definition

1.4.1 Camera FOV (Requirement 3.1.1)

1. Test Case Description
To determine the field of view (FOV) we measure it directly with a tape measure. We will point the camera at a wall. Looking through the camera (monitor), we can measure horizontally across the entire field of view right at the edge of visibility, to obtain the width/height. Width and height will be measured in cm.

2. Test Environment
All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set
The camera will have a specified FOV, which is determined by the model of the camera, and is measured in degrees.

4. Expected data values and results
The expected data will be the width and height of the FOV measured in centimeters and used to calculate degrees. The expected value of this will be at least 60 degrees horizontally and 40 vertically.

5. Test Procedure:

- Test Initialization:
  - Connect motors to Arduino and the Arduino to the Raspberry Pi.
  - Initialize motor positions to be centered on the horizontal and vertical axis.

- Test Steps:
  1. Run Test Code to rotate the horizontal motor to its maximum clockwise direction.
  2. Measure degrees rotated from the starting position.
  3. Repeat the first two steps for the maximum counterclockwise direction.
  4. Repeat the steps for vertical rotation.

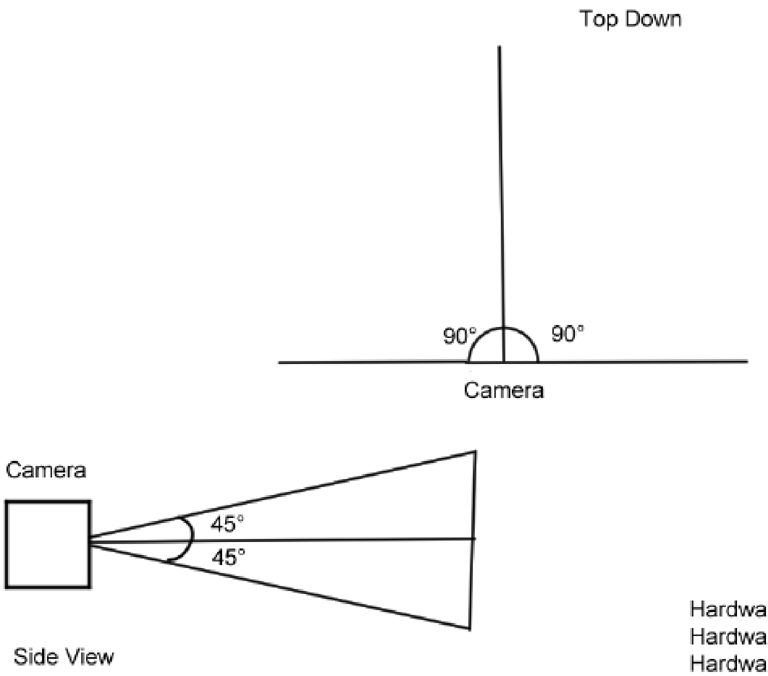
- Test Data Log:
    - The maximum degree difference from the starting position will be recorded for the clockwise and counterclockwise changes for both the horizontal and vertical motors. Each direction and motor will be run 2 times.

- Anomaly Report:
    - Measurements below the required maximum field of view will be recorded and steps will be taken to correct the error.

- Suspension Criteria and Resumption Requirements:
    - In the event that the field of view is not within the required range there may be serious hardware errors.

- Test Termination:
    - If motors fail to move within specified parameters the test will be terminated. If motor rotation is within an acceptable range the test will be terminated.

- Test Data Log Format:
    - The maximum range of each motor, per trial, in each direction will be logged in an excel spreadsheet.



Hardware Component 1: Camera
Hardware Component 2: Raspberry Pi
Hardware Component 3: Motors
Software Component 1: Motor Control

Test Diagram 1.4.1: Component and Modules Used to Validate Requirement 3.1.1

1.4.2 Camera FPS (Requirement 3.1.2)

1. Test Case Description

The FPS of a camera can be measured directly through software. The frames per will be monitored for a period of at least 10 min and sampled every second. The average FPS will then be calculated. This will be measured in average frames per second.

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

For the first part of this test, we will not input anything, just let the camera run. The second part of this test will input a tester's face to test the load on the system.

4. Expected data values and results

The expected data for this test will be at least 10 frames per second and will be considered successful if it remains above 10 frames per second on average for the entire process of the camera starting to track a user to it taking a picture.

5. Test Procedure:

- Test Initialization:
  - Raspberry pi will be connected to a temporary test display.
  - Device will first be tested with no user present.

- Test Steps:
  1. Run the main program with test code included, measuring the frames captured in one second by incrementing a counter for every frame captured from the camera and displaying and resetting the counter after 1 second has elapsed.
  2. Record one minute of FPS readings and find the average.
  3. Introduce the user in the center of the camera's field of view.
  4. Repeat steps one and two.

- Test Data Log:
  - Each run will record one minute of the FPS every second. There will be two runs, one including a user and one without. Depending on the consistency of the FPS values there could be more runs as required.

- Anomaly Report:
    - Incidents that below expected results, will be recorded with other data. Troubleshooting will occur and steps taken will also be noted

- Suspension Criteria and Resumption Requirements:
    - Suspension will occur if FPS numbers are drastically under required values (4 FPS or less).

- Test Termination:
    - Test termination will occur if FPS values are too low to track faces or the camera is not detecting a face.

- Test Data Log Format:
    - Individual FPS values will be logged for each run in an excel spreadsheet along with the calculated average of each run.

1.4.3 Camera Detection Accuracy (Requirement 3.1.3-3.1.7, 3.4.1-3.4.3)

1. Test Case Description

A user will be placed within a static range designated within requirement 3.1.3 and will move at constant speeds within the range given in requirement 3.1.4. Under the above conditions within various ranges, head orientation, and speeds the device will be activated at least 50 times and we will record whether the camera could successfully identify the user, ignoring false positives.

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

The range of usable radius of the device is within 1 to 3 meters of the device. The user will be able to move from 0 to 0.8 meters per second within the device's range. The user can tilt their face within 25 degrees horizontally and vertically looking towards the camera.

4. Expected data values and results

This test will be measured in the percent of successes. The expected value of this test will be an average of an 80% success rate.

5. Test Procedure:

- Test Initialization:
  - Raspberry pi will be connected to a temporary display
  - User will be placed within the center of the camera's FOV and will remain still.

- Test Steps:
  1. The program will be run along with test code to measure frames with detected faces and those without.
  2. Frames that detected a face and frames that did not will be totalled and recorded.
  3. The user will then begin moving at a speed of 0.5 m/s horizontally across the center of FOV.
  4. Steps one and two are repeated for new conditions.
  5. The user will then begin moving at a speed of 0.5 m/s vertically across the center of FOV.
  6. Steps one and two are repeated for new conditions.

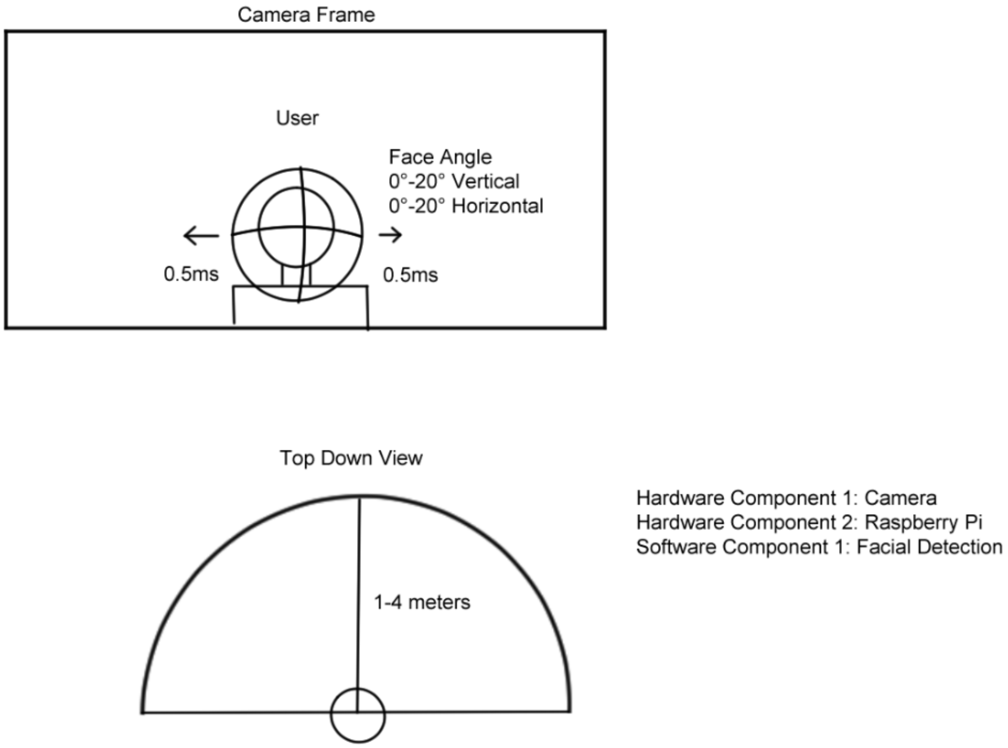
- Test Data Log:
  - This test will be measured in the number of successes and number of failures.

- Anomaly Report:
  - If the number of failures outweighs the number of successes, this result will be noted and steps will be taken to resolve this problem.

- Suspension Criteria and Resumption Requirements:
  - If the number of successes are not within the specified numbers the test will be suspended.

- Test Termination:
  - The test will be terminated if the number of failures outweighs the successes or if the number of successes is at or above the required number.

- Test Data Log Format:
  - Number of successes and failures will be logged in an excel spreadsheet for each run and a percent success rate will be calculated.
  - Percent Detection Formula:
    - % Success = [(Number of Successful Runs - Number of Unsuccessful Runs - False Detections) / Total Test Runs] * 100


Camera Frame
User
Face Angle
0°-20° Vertical
0°-20° Horizontal
0.5ms
0.5ms
Top Down View
1-4 meters
Hardware Component 1: Camera
Hardware Component 2: Raspberry Pi
Software Component 1: Facial Detection

Test Diagram 1.4.3: Components and Modules Used to Validate Requirements 3.1.3-3.1.7 and 3.4.1-3.4.3

1.4.4 Camera Rotation Range (Requirement 3.2.1)

1. Test Case Description

The range of the camera's rotation can be directly measured to ensure that its rotations are within specification. This will be measured in degrees.

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

We will input the range of the camera rotation, which will be 90 degrees, horizontally left and right, and 45 degrees, vertically up and down, of its set starting position.

4. Expected data values and results

This test will measure the range in degrees. The expected value of the camera's range is 180 degrees horizontally and 45 degrees vertically.

5. Test Procedure:

- Test Initialization:
  - Connect Arduino to laptop
  - In the Arduino program, set a starting position of 0 degrees for azimuth and 45 degrees for elevation.
  - Line a protractor up to the zero location for azimuth..

- Test Steps:

  1. Insert a command for azimuth to move to 105 degrees. Record actual degree traveled using protractor.
  2. Insert a command for azimuth to move to -105 degrees. Record actual degree traveled using protractor. Return azimuth to zero.
  3. Set elevation to 45 degrees. Line up protractor (0 degree) with elevation assembly.
  4. Repeat steps 1 and 2.

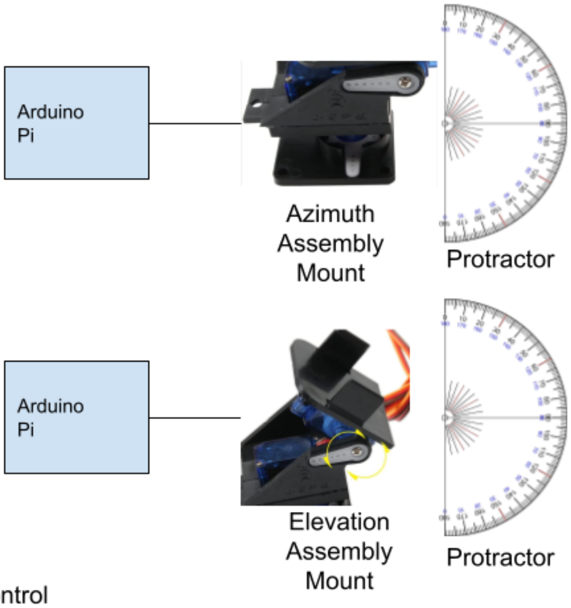
- Test Data Log:
  - ○ Each run will record the degree to which the camera was rotating. There will be three runs in each direction (right, left, up, and down).

- Anomaly Report:
  - Incidents that below expected results, will be recorded with other data. However, multiple test runs will allow us to determine frequency of occurrence.

- Suspension Criteria and Resumption Requirements:
  - Test suspension will occur if motors do not move if commanded. Will be resumed if they respond to commands

- Test Termination:
  - Test termination will occur if the motors do not move on command.

- Test Data Log Format:
  - Test data will be logged in excel. Each entry will be given a standard of error in degrees. Graphs will be used to display the percentage of runs that meet the requirements.



Module: Motor Control
Hardware Component: Servo Motors
Hardware Component: Arduino
Hardware Component: Raspberry Pi
Software Component 1: insertDegreeAngles

Test Diagram 1.4.4: Component and Modules used in Test Case 4 to validate Requirement 3.2.1

1.4.5 Camera Rotation Speed (Requirement 3.2.2)

1. Test Case Description

This requirement can be tested by having a user move within the range of the camera and if the user remains within the center of the camera's captured image then it is matching the user's speed.

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

The data inputted for this test will be reliant on the test subjects speed and position which will remain within the specified range. The test subject will move at various speeds within the range.

4. Expected data values and results

The expected value of this test will be measured in meters per second. The expected result of this test is to have it match the user's speed.

5. Test Procedure:

- Test Initialization:
  - Connect Arduino to laptop.
  - One tester will sit 1.5 meters away from the camera.
  - Mark 1 meter right and left of the seated position of the tester with tape.
  - Find 1 meter above and below the top of the tester's head with a meter stick.
  - Tester will sit still during camera power on and through initialization.
  - Set protractor starting azimuth at 0.

- Test Steps:

  1. Set azimuth to 0 and center the camera on the still tester. Align protractor to azimuth assembly (see test procedure 1.4.4).
  2. At a speed of 0.1 meters per second, move to the tester's left, making sure that the tester's image is within the center of the camera image. If the camera has reached the maximum range (90 degrees), record a success, otherwise a failure. Repeat this step four times.
  3. Repeat camera initialization steps. Repeat step 1, except for moving to the tester's right, making sure that the tester's image is within the center of the camera image. If the camera has reached the maximum range (90 degrees), record a success, otherwise a failure. Repeat this step four times.
  4. Return azimuth to 0 degrees. Then, set elevation to 45 degrees. Align protractor with elevation assembly (see test procedure for 1.4.4).
  5. Repeat camera initialization steps. At a speed of 0.1 meters per second, the tester will stand up, making sure that the tester's image is within the center of the camera image. If the camera has reached the maximum range (45 degrees), record a success, otherwise a failure. Repeat this step four times.
  6. Repeat camera initialization steps. Repeat step 3, except the tester will be moving down. If the camera has reached the maximum range (45 degrees), record a success, otherwise a failure. Repeat this step four times.
  7. Repeat steps 1-4, with the tester moving at 0.8 meters per second.

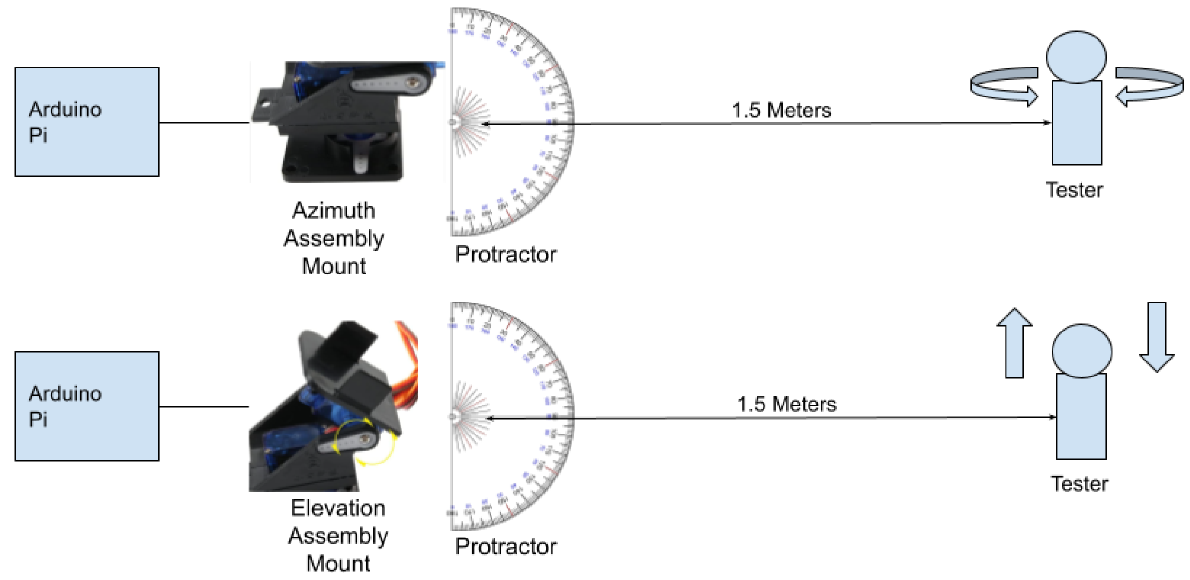
- Test Data Log:
  - This test will be run at least 4 times per direction and minimum/maximum speed (32 times total). Each trial will record the direction, speed, and success of track targets.

- Anomaly Report:
  - Incidents that perform below expected results, will be recorded with other data. However, multiple test runs will allow us to determine frequency of occurrence.

- Suspension Criteria and Resumption Requirements:
  - Test suspension will occur if the motors do not respond to moving faces. Test will resume if the issue is fixed

- Test Termination:
  - Test termination will occur if the motors do not respond to the tester.

- Test Data Log Format:
  - The test data will be recorded in excel. We will compute the average percentage of successful trials and graph them.



Modules: Facial Recognition and Motor Control
Hardware Component: Servo Motors
Hardware Component: Arduino
Hardware Component: Raspberry Pi
Software Component 1: keepFaceTracking
Software Component 2: keepMotorsMoving

Test Diagram 1.4.5: Component and Modules used in Test Case 5 to validate Requirement 3.2.2

1.4.6 Image Aspect Ratio (Requirement 3.3.1)

1. Test Case Description

Image aspect ratio can be measured and is displayed within software

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

There will be no input data for this test

4. Expected data values and results

The aspect ratio of images is measured in the number of pixels height and width. The expected value of this test is 1080x1080.

5. Test Procedure:

- Test Initialization:
  - Raspberry pi will be connected to a temporary display.

- Test Steps:
  1. Images will be captured using a test program.
  2. Images will be taken until 20 images are saved.
  3. Record the image aspect ratio of each trial.

- Test Data Log:
  - This test will be measured in the successes of the aspect ratios of resulting images.

- Anomaly Report:
  - If the number of images saved exceeds 20 images or if any of the images are not in the specified aspect ratio then the test will be considered a failure and will be redone.

- Suspension Criteria and Resumption Requirements:
  - Test will be terminated if no images are being saved.

- Test Termination:
  - Tests will be terminated if images are not being saved.
- Test Data Log Format:
  - Trials will be recorded in a master excel file with calculated success rate.

1.4.7 Displaying/Storing Images (Requirement 3.3.2-3.3.4)

1. Test Case Description

The system will show images within an accessible display. Once 20 images are taken displayed the device will be activated again, if the device saves more images then it will have failed this test.

2. Test Environment

All tests will be conducted in a lab environment with ideal lighting. The background of all tests will be against a static background of a light color (either white or beige). The test environment will be room temperature and will not be introduced to any moisture.

3. Input Data Set

Commands will be input into the Raspberry Pi.

4. Expected data values and results

This test will be measured in the success rate. The expected result is that once the memory has reached its designated limit of images it will cease capturing images.

5. Test Procedure:

- Test Initialization:
  - Raspberry pi will be connected to a temporary display.

- Test Steps:
  1. Images will be captured using a test program.
  2. Attempt to take 25 images.
  3. Note how many images were actually taken
  4. Power cycle Raspberry PI.
  5. Check PI to see how many images were successfully stored

- Test Data Log:
  - This test will be measured in the successful saving of images. Test will be conducted three times.

- Anomaly Report:
  - If the number of images saved exceeds 20 images then the test will be considered a failure and steps will be taken to fix the problem and the test will be redone

- Suspension Criteria and Resumption Requirements:
  - Tests will be suspended if cameras stop capturing images. Will be resumed if it begins capturing images.

- Test Termination:
    - Test will be terminated if no images are being saved.

- Test Data Log Format:
    - Trials will be recorded in a master excel file with calculated success rate.

1.5 Test Analysis Criteria

1.5.1 Test Case 1

1. Defined Criteria for Successful Test Completion
This test will be considered successful if the camera's FOV is within 60 degrees horizontally and 40 degrees vertically. Any value above these numbers is acceptable.

2. Defect Severity Definitions:

Data Type	Mean Degree Error: ± 0.5 Degree	
Degrees of Elevation	>= 50	Excellent
	>= 45	Good
	>= 40	Satisfactory
	< 40	Failure

Data Type	Mean Degree Error: ± 0.5 Degree	
Degrees of Azimuth	>= 80	Excellent
	>= 70	Good
	>= 60	Satisfactory
	< 60	Failure

1.5.2 Test Case 2

1. Defined Criteria for Successful Test Completion
This case will be considered successful as long as the FPS remains at an average of 10 frames per second for the entire duration of a use of the device.

2. Defect Severity Definitions:

Data Type	Mean FPS Error: ± 1 Frame	
FPS	15 >=	Excellent
	>= 10	Satisfactory
	< 10	Failure

1.5.3 Test Case 3

1. Defined Criteria for Successful Test Completion
This test will be considered successful if while the user remains within the set conditions then the device will recognise faces 80% of the time. If on average after 50 test cases the success rate is 80% then it will be considered a success.

2. Defect Severity Definitions:

Data Type	Mean Percent Error: ± 0.5%	
Percentage (of Seconds)	> 90%	Excellent
	> 80%	Satisfactory
	< 80%	Failure

1.5.4 Test Case 4

1. Defined Criteria for Successful Test Completion
As long as the camera begins in its designated starting position the device will be considered successful if it can rotate 90 degrees horizontally left or right from its starting position and 45 degrees vertically from its starting position.

2. Defect Severity Definitions:

Data Type	Mean Degree Error: ± 0.5 Degree	
Degrees of Azimuth	> 100	Excellent
	99 ≤ X ≤ 90	Satisfactory
	< 90	Failure

Data Type	Mean Degree Error: ± 0.5 Degree	
Degrees of Elevation	> 50	Excellent
	45 ≤ X ≤ 50	Satisfactory
	< 45	Failure

1.5.5 Test Case 5

1. Defined Criteria for Successful Test Completion

This test will be considered a success if the camera can match the user’s speed within 0.1 meters per second and 0.8 meters per second at least 80% of the time.

2. Defect Severity Definitions:

Data Type	Mean Time Error: ± 0.2 seconds	
Percentage (of Seconds)	>= 90%	Excellent
	>= 80%	Satisfactory
	< 80%	Failure

1.5.6 Test Case 6

1. Defined Criteria for Successful Test Completion

This test’s scenario is to have an image aspect ratio of 1080x1080.

2. Defect Severity Definitions:

Data Type		
Aspect Ratio	>= 1080x1080	Excellent
	< 1080x1080	Failure

1.5.7 Test Case 7

1. Defined Criteria for Successful Test Completion
This test will be considered a success if the device can store at least 20 images and the device can detect how many images have been stored at least 80% of the time.

2. Defect Severity Definitions:

Data Type	Mean Error: None	
Number of Images Saved	> 20	Failure
	== 20	Satisfactory
	< 20	Failure

Data Type	Mean Percent Error: ± 0.5 %	
Percentage (Completion of Storing Images)	>= 80%	Excellent
	>= 75%	Good
	>= 65%	Satisfactory
	< 60%	Failure

2. Execution Plan

Test Case Order:

- Test Case 4: Camera Rotation Range
- Test Case 1: Camera FOV
- Test Case 2: Camera FPS
- Test Case 3: Camera Detection Accuracy
- Test Case 5: Camera Rotation Speed
- Test Case 6: Image Aspect Ratio
- Test Case 7: Displaying/Storing Images

3. Test Schedule

August 2nd: Case 4, Case 1, Case 2
August 4th: Case 3, Case 5, Case 6, Case 7

4. References

System Requirements Specification
System High Level Design Specification