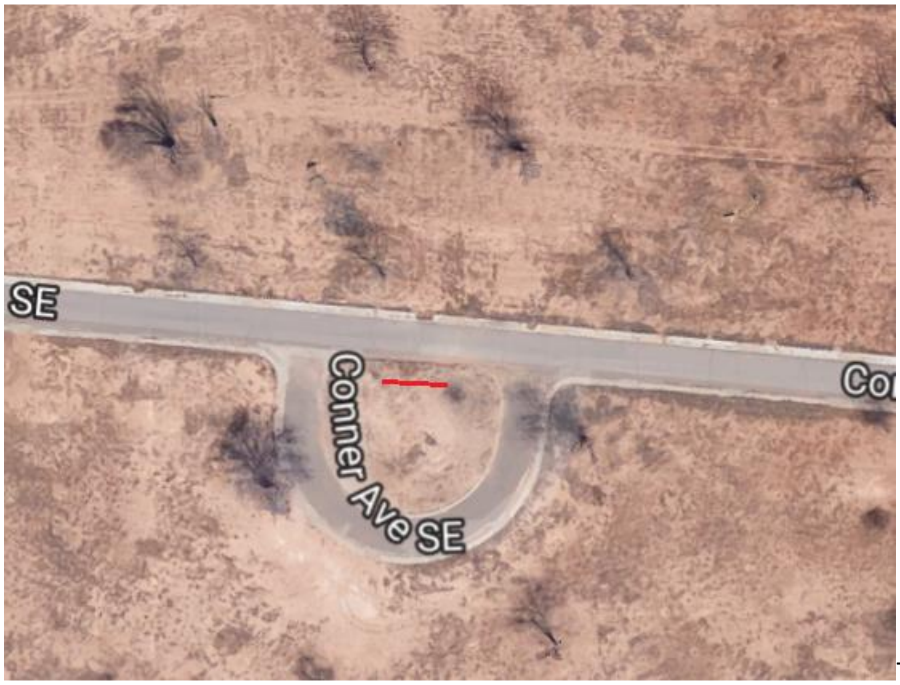
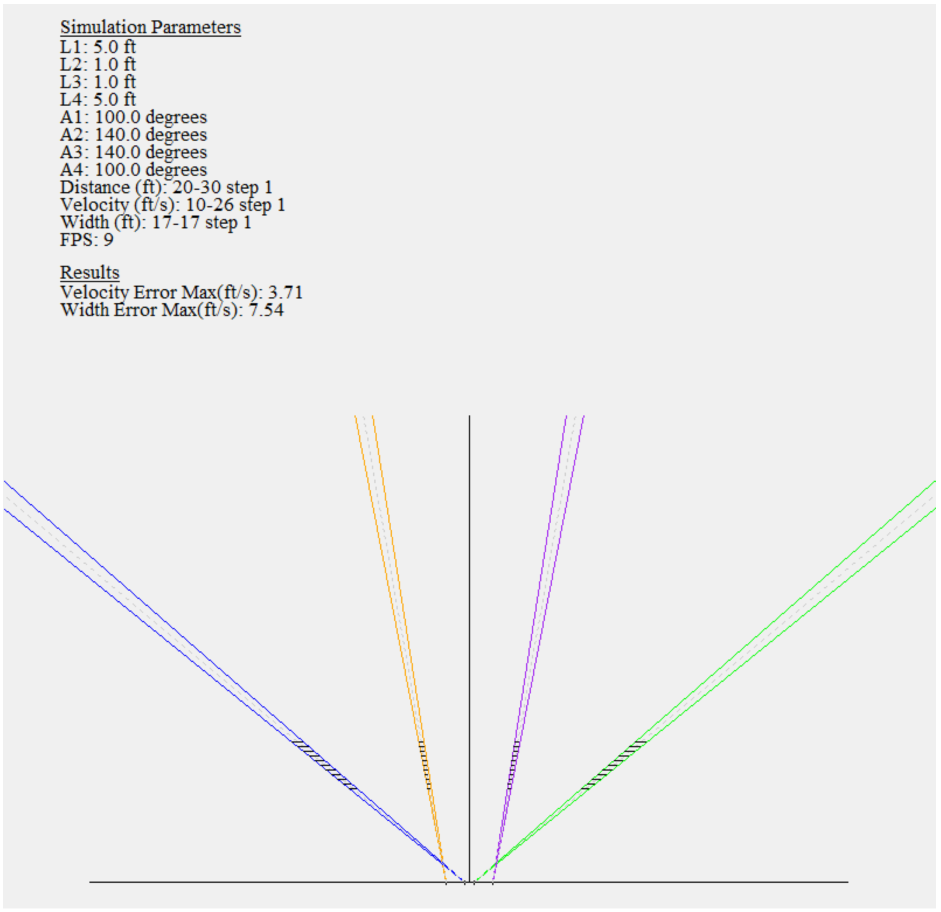
First Data Gathering Report

Matthew DeKoning

The first test site took place in an abandoned neighborhood on Kirtland Airforce base from 9 am to 12 pm. The location features a straight road with a loop to easily turn around.

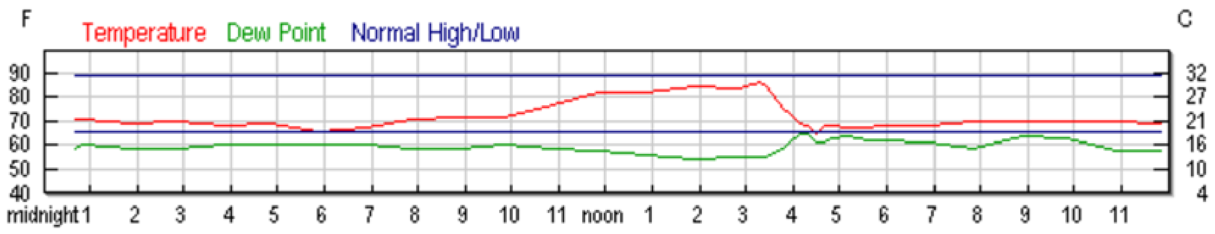


The sensor array was set up 25 feet from the straight road inside of the loop. The farthest two cameras were configured five feet from center focused at an angle 140 degrees from center and the middle two cameras were placed one foot from the center at an angle of 100 degrees from center.


Simulation Parameters
L1: 5.0 ft
L2: 1.0 ft
L3: 1.0 ft
L4: 5.0 ft
A1: 100.0 degrees
A2: 140.0 degrees
A3: 140.0 degrees
A4: 100.0 degrees
Distance (ft): 20-30 step 1
Velocity (ft/s): 10-26 step 1
Width (ft): 17-17 step 1
FPS: 9

Results
Velocity Error Max(ft/s): 3.71
Width Error Max(ft/s): 7.54

The temperature rose from approximately 70 degrees Fahrenheit to slightly over 80 degrees Fahrenheit. A fairly linear increase of 10 degrees from 10 am to 12 pm.



https://www.wunderground.com/history/airport/KABQ/2017/7/31/DailyHistory.html?req_city=&req_state=&req_statename=&reqdb.zip=&reqdb.magic=&reqdb.wmo=

A dodge caravan van (17 feet wide) was driven at 15 – 25 miles per hour in both directions along the straight road as one test subject; a person walking in front of the cameras was used as the second test subject.

The Raspberry Pi, cameras, monitor, and input devices were powered with a large battery. A powered USB hub compatible with the Raspberry Pi had not yet been found, so a script was created to trigger the data gathering program on a timer, allowing the keyboard to be unplugged and the fourth Lepton to be plugged in (a Raspberry Pi only has 4 USB ports, so without a hub those are monopolized by the cameras).

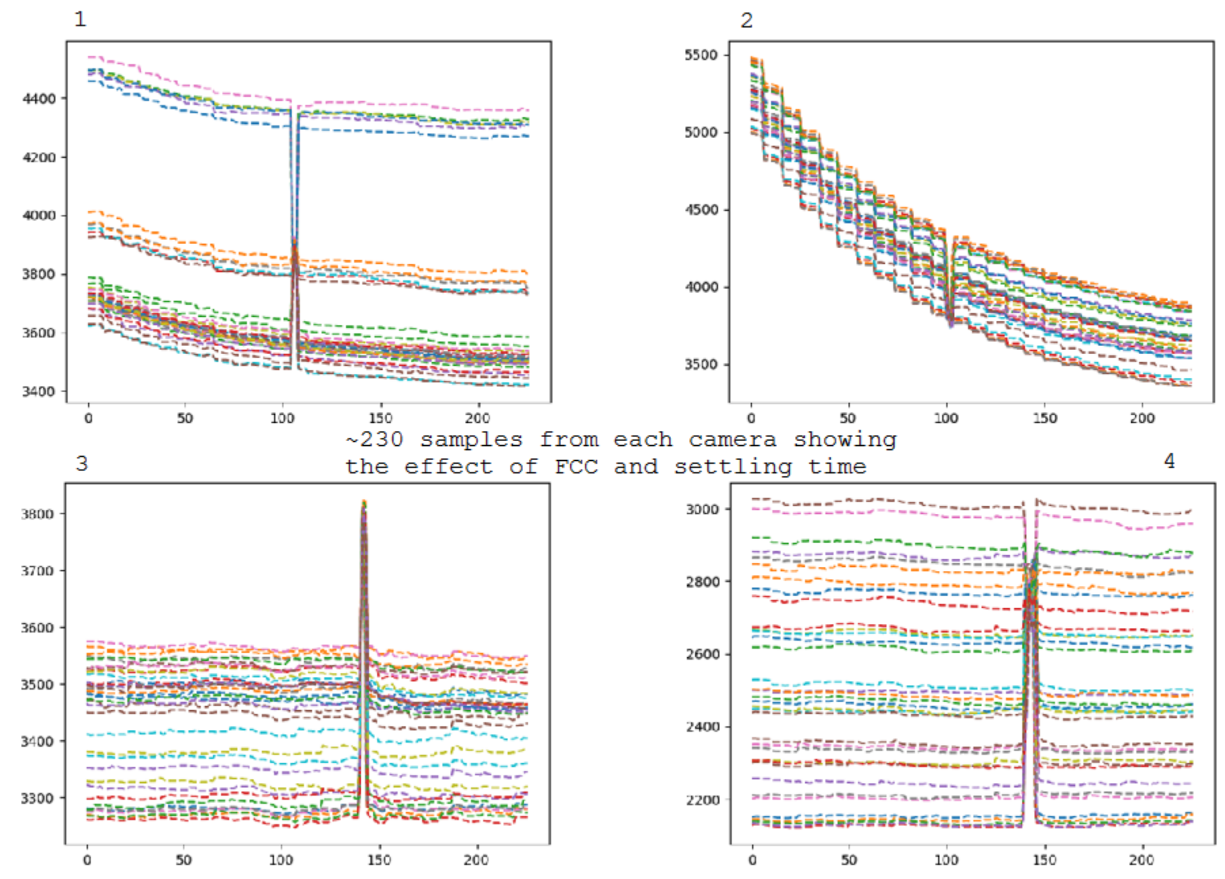
The algorithm tested gathers a window of thirty two samples for each of the central six by six pixels of the cameras. The mean, median, and standard deviation (extracted via the Median Absolute Deviation calculation) are tracked and incoming data is tested against a certain statistical threshold to determine if it has changed due to an intrusion or simply a shift in ambient temperature. The threshold used here was six sigma (giving a probability of 99.9999998026825 percent that an incoming temperature reading would lie within the acceptable bounds, given the thirty two previous samples have created a reasonable statistical model for the temperature drift the camera is experiencing). When an intrusion is detected one of two things could happen: the alerted pixel count for this camera is incremented and the value is added to the thirty two value window, or the alert flag is incremented and the value is not added. The latter method attempts to leave the running statistics undisturbed by the intrusion, but the effect of the rise or fall in temperature on the value it eventually returns/settles to is unknown. However, the implementation of discarding intrusions was flawed during the first test and a pixel's standard deviation could become zero. Given any statistically unlikely data will not be added to the running average, this freezes the window and causes constant alerting.

A variety of outputs for the tests were possible: camera number, time stamps, and pixel count changed (intrusion information), pure data – the value of each pixel analyzed paired with its index (camera, location within the six by six window) and statistics (mean, median, standard deviation), or pure data and intrusion information. The most useful proved to be both pure data and intrusion information.

The algorithm did not perform well on the first test for a variety of reasons. The first and largest reason is the settling time of the cameras was not taken into account. Since the USB plugs were constantly being swapped out to allow editing and script starting with the keyboard or run the algorithm with all four cameras, the cameras being swapped were not given time to adjust to the background temperature.

All graphs were created using Python and pyPlot.

Problems



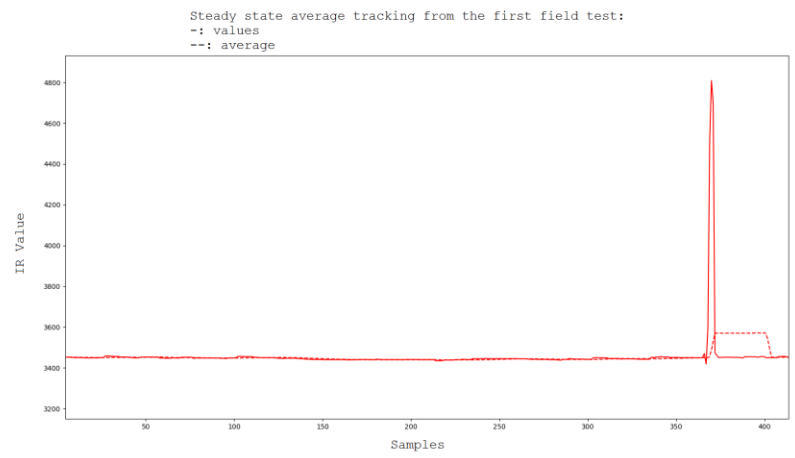
~230 samples from each camera showing the effect of FCC and settling time

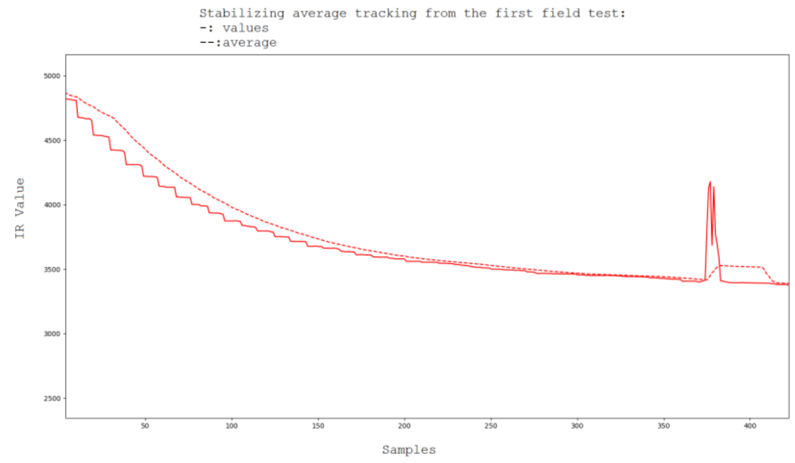
Figures 3 and 4 show cameras that have been allowed to adjust to the ambient temperature landscape they are viewing. Figure 4 shows a good range of temperature scenery being monitored, yet the intrusion around sample 140 still clearly changes each pixel value. Figures 1 and 2 show the cameras response as they perform flat-field correction to adjust to the ambient surrounding temperature. Figure 2 in particular shows the effect of the FCC, which is triggered ever second or every nine frames.

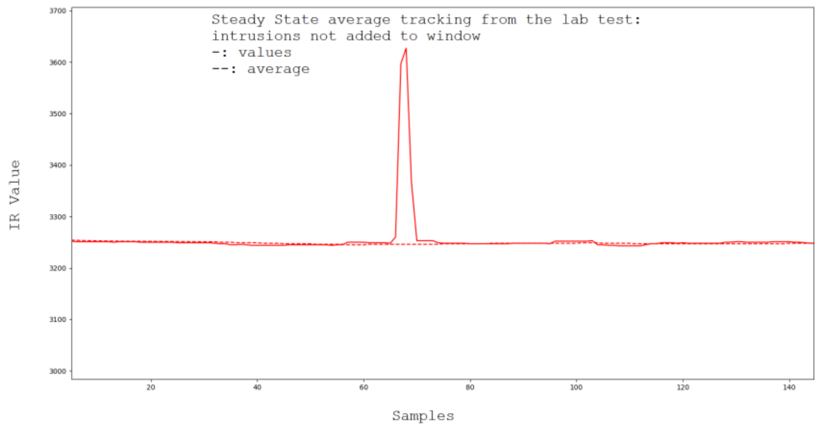
These graphs were obtained during post-processing. In the field I was aware the algorithm was not behaving well, but unsure of where to begin in terms of fixing it. After I began looking at the data, I set up the cameras in the lab, all 90 degrees from the ‘x axis’ and spaced 3 feet and 1 foot from the center. With this setup I determined the cameras need at least a minute if not two to settle before reading their data, as well as found the standard deviation bug. A powered USB hub that allowed all peripherals to always be plugged in was also located.

Validations

The statistics (aside from steady state standard deviation) worked well.



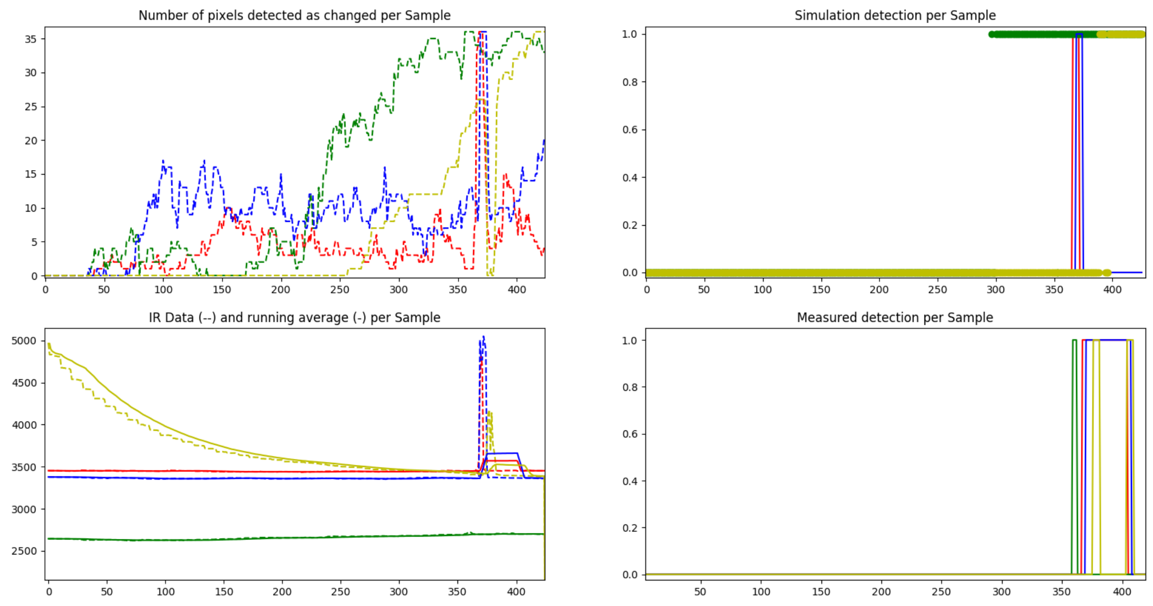




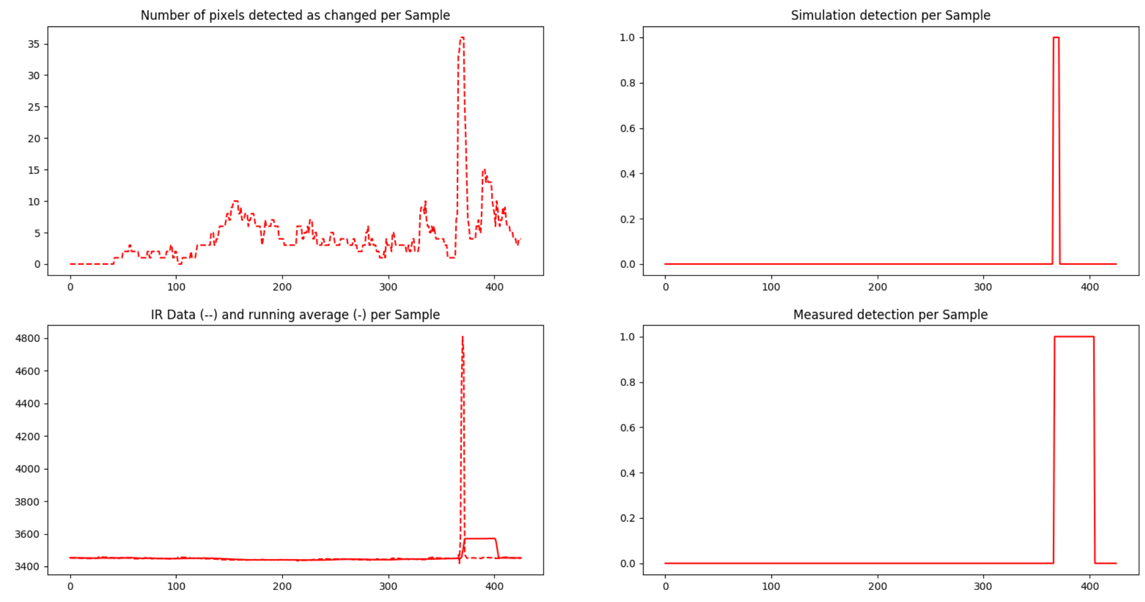
Results

The following graphs show data gathered and algorithm results versus simulated results run over the same data. Both simulation and the Raspberry Pi algorithm used a statistical threshold of six sigma, but the field test had a changed pixel to detection threshold of 25 while the simulation had 30.

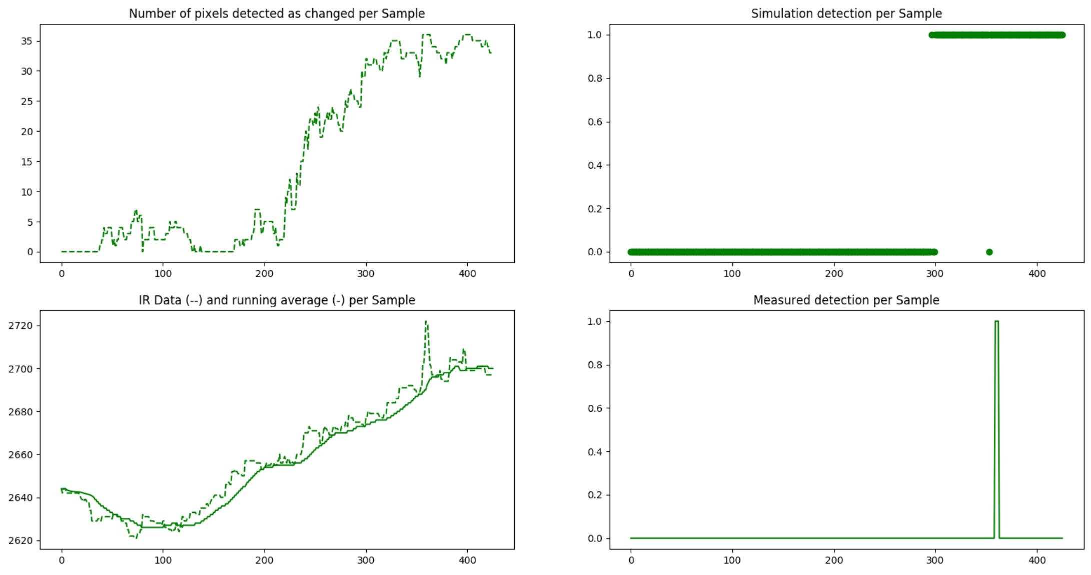
Field Test: Car 15 mph at 25 feet



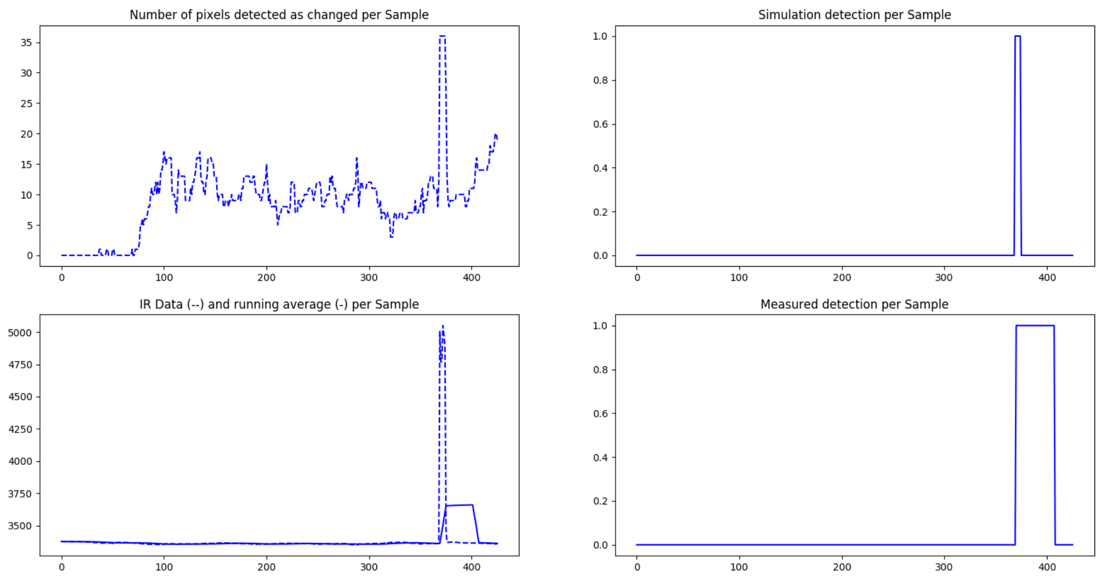
Field Test: Car 15 mph at 25 ft Camera 1

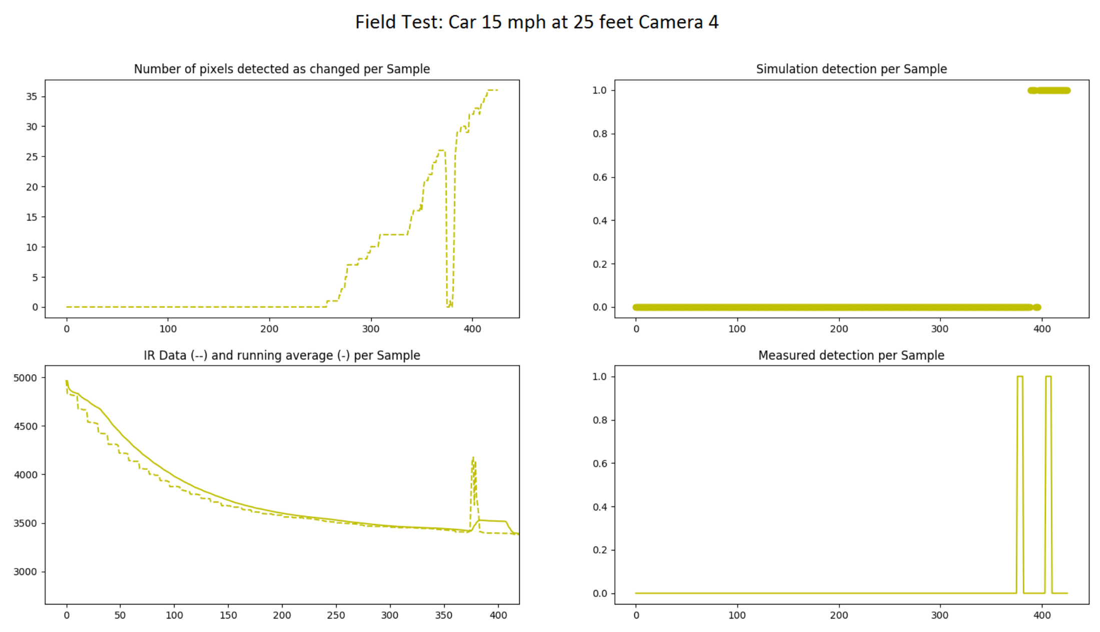


Field Test: Car 15 mph at 25 feet Camera 2



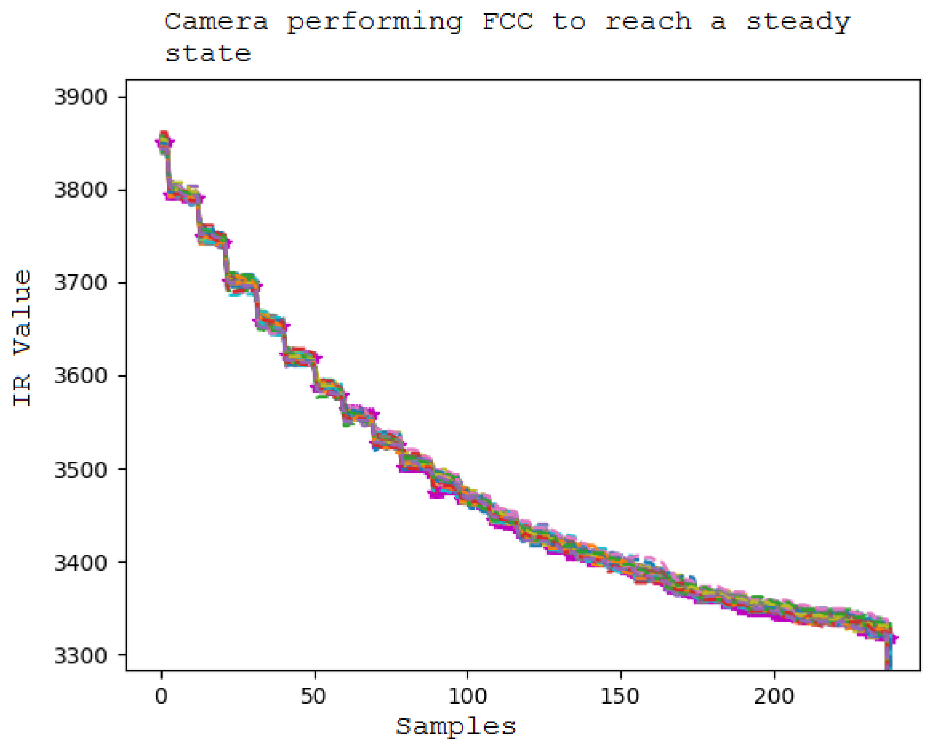
Field Test: Car 15 mph at 25 feet Camera 3

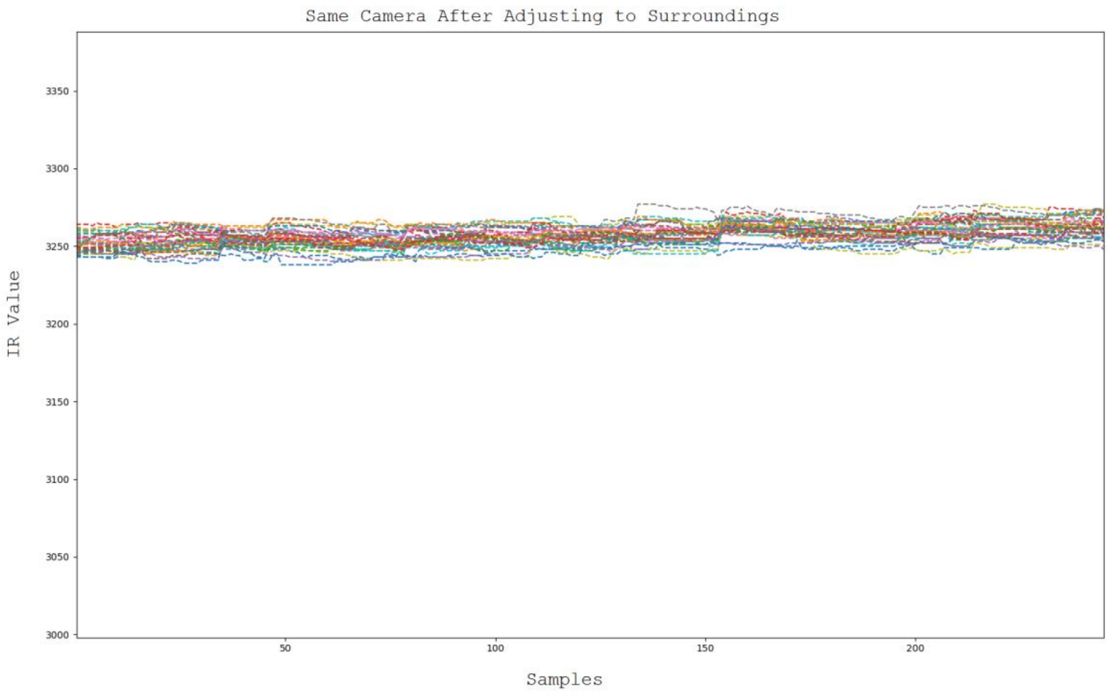




Investigating Discontinuities

The data below was captured in an air conditioned (more thermally stable than outside) lab when trying to determine the cause of the discontinuities in the previous graphs.

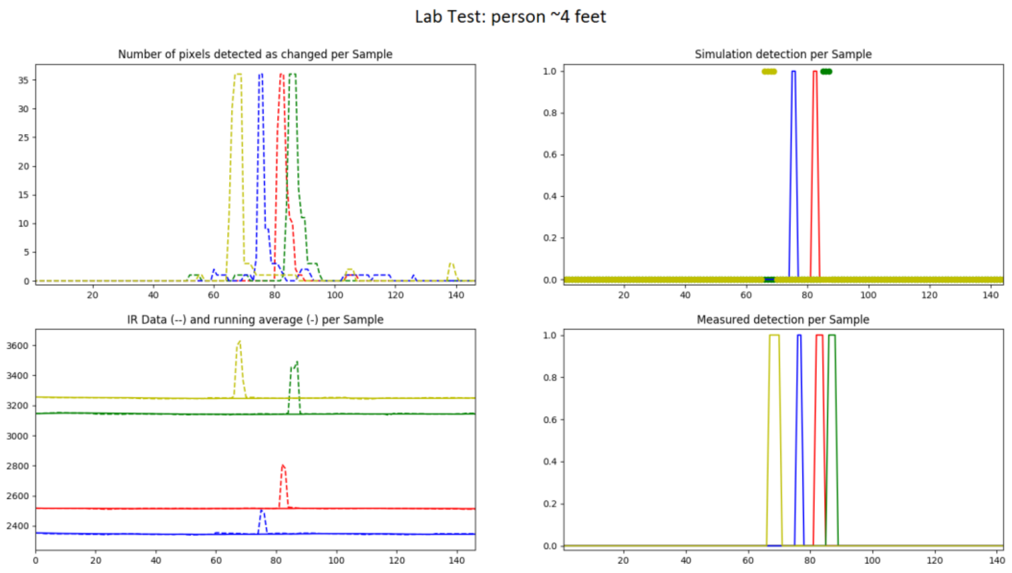




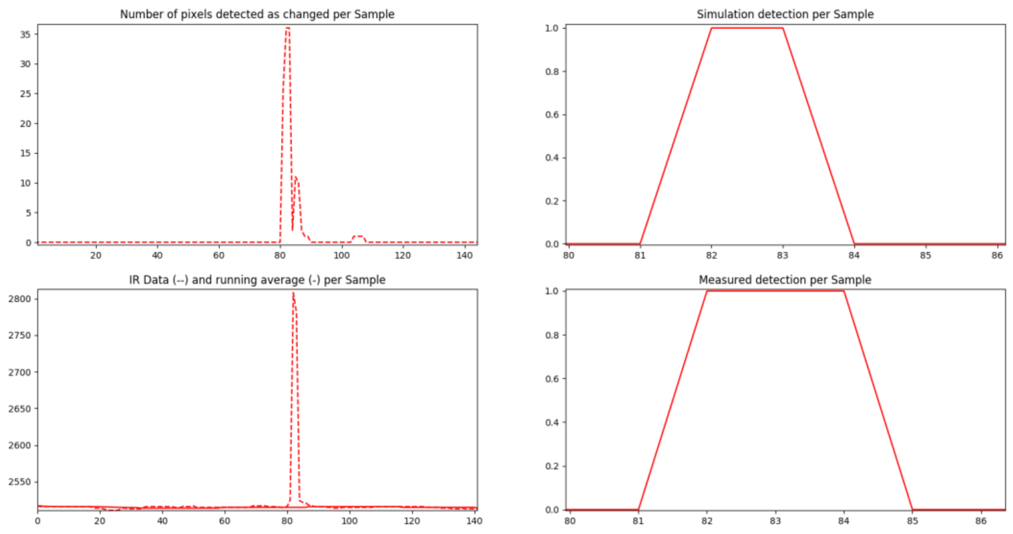
Given the powered USB hub will now allow cameras to remain plugged in during all testing, the rough estimate of 90 seconds for camera stability is acceptable, the cameras will be plugged in for multiple minutes before any future tests, so the problem should be resolved.

Further Work

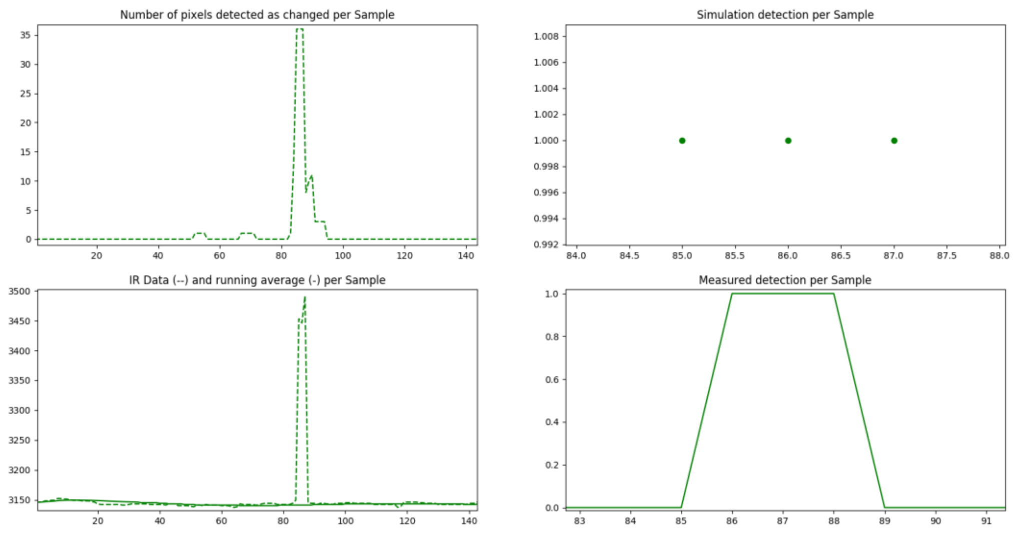
The following are graphs created from data in a lab after debugging the algorithm. The statistic threshold on the Raspberry Pi was changed to 27, aside from that the test remained the same.



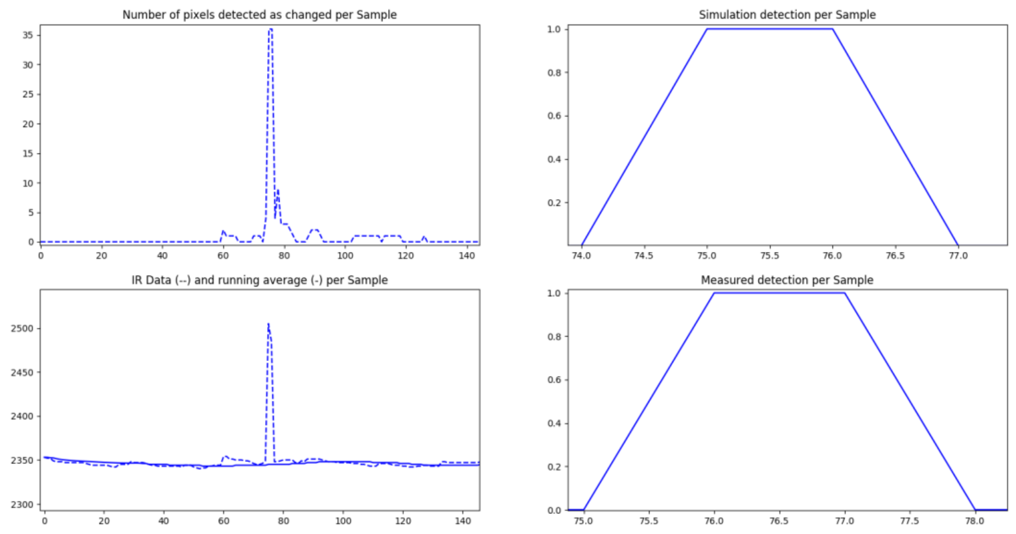
Lab Test: person ~4 feet Camera 1



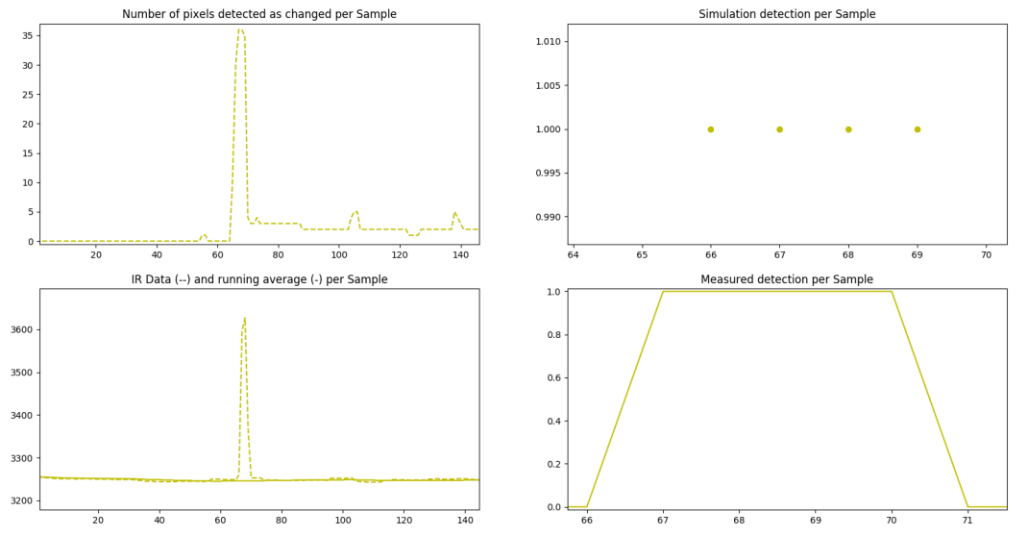
Lab Test: person ~4 feet Camera 2



Lab Test: person ~4 feet Camera 3



Lab Test: person~4 feet Camera 4



This data shows nearly a one to one detection in simulation to detection in practice after adjusting the algorithm and sensor setup based on the knowledge gained from the first test.

False alarm changed pixels – those that are triggered by thermal noise or other interferers – in this scenario are kept beneath five per frame while residual pixels – those registering a changed due to a recent intrusion – are kept beneath sixteen. This suggests the changed pixel threshold could be lowered to around twenty and the results would remain the same. However this is in a much less noisy lab environment. The next step is to do some basic tests outdoors again, then set up another large data gathering day.