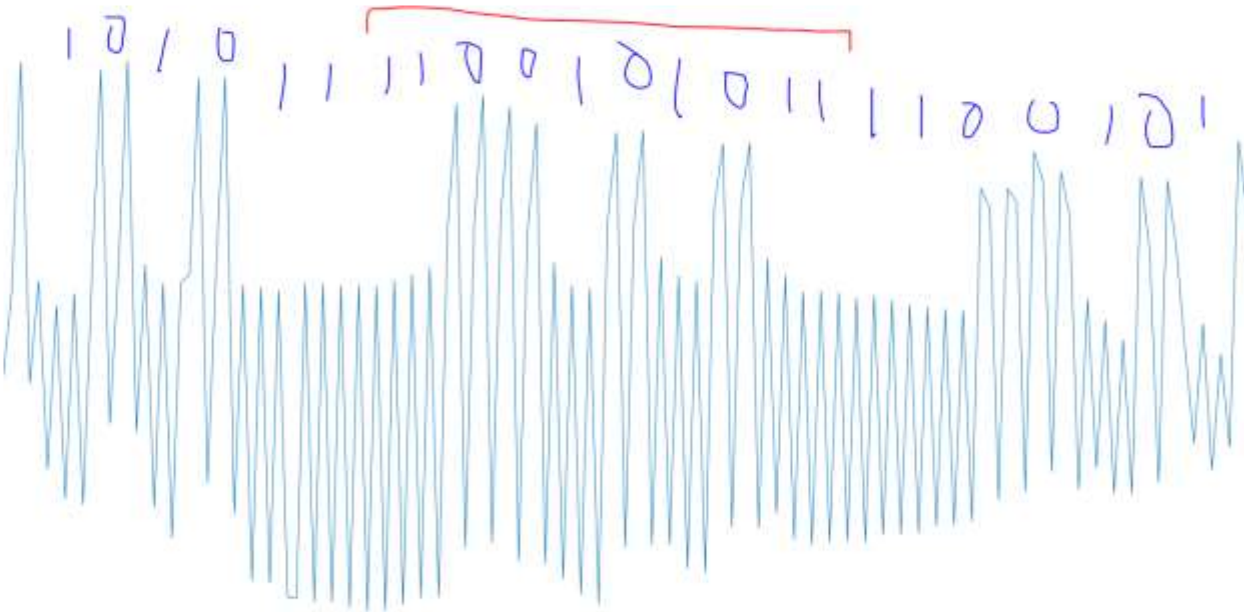
Alex Ching
CSE 562 SPR21 A3

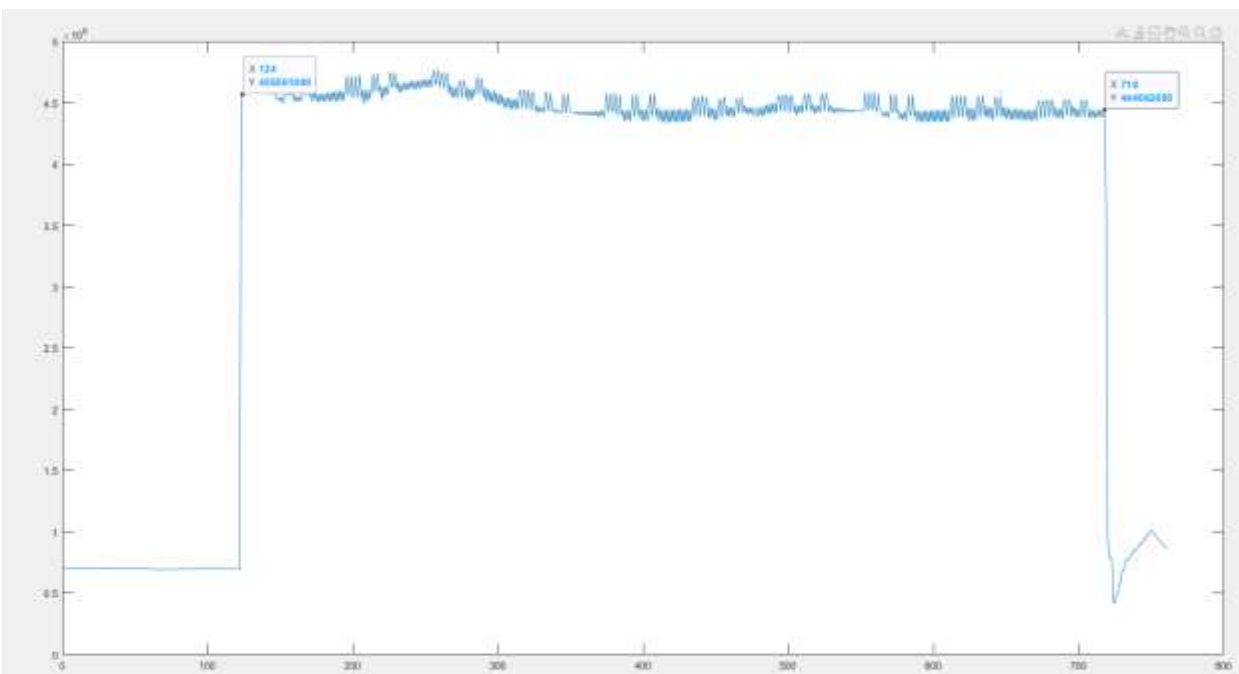
I created a script in Matlab to create a video from a static image. I am just using one color and varying the value of the blue in this example to adjust the alpha. From the paper, I interlaced the images in the order that would represent a zero or one in six frames.

I made an arbitrary sequence ‘1100101011’ that would take up one total second with each bit being one frame in a video that runs at 60Hz which is the frequency my monitor runs on. I repeated that sequence for 10 seconds with black on either side.

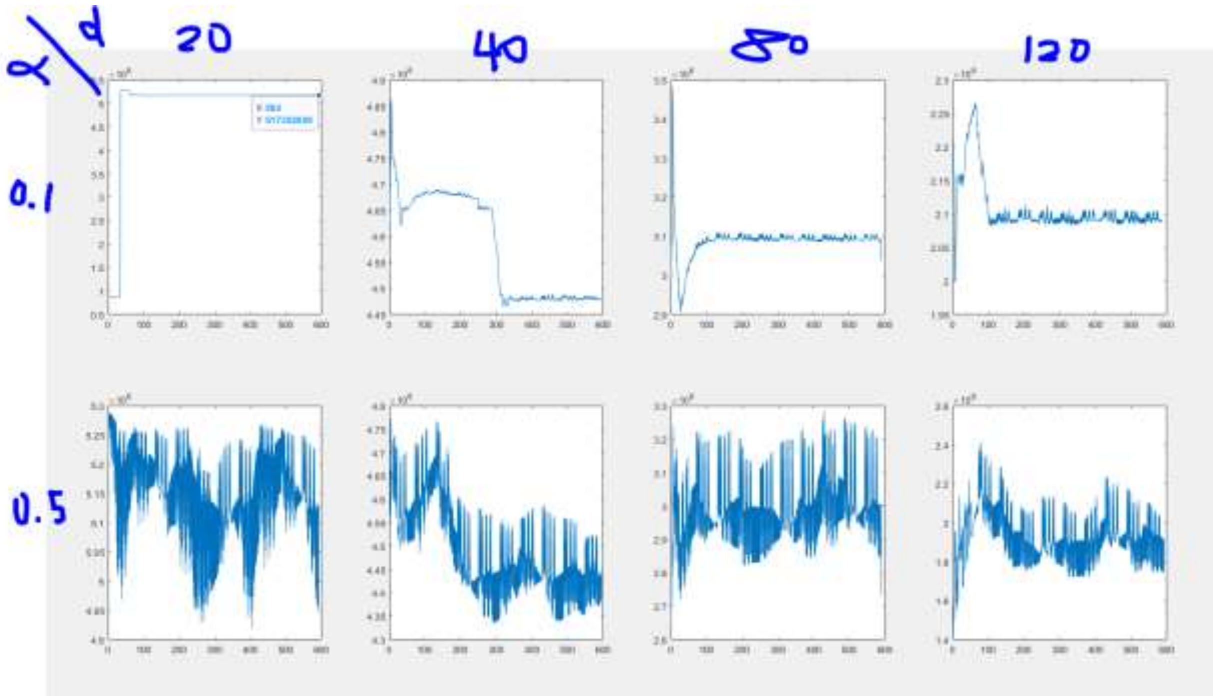
For the receiver, since the image is static and is just a color, there is really no need to do scene detection. It was easier to do the analysis as a post process so I used another Matlab script to analyze the video data. Looking at the raw average value of the frames as a sanity test you can see that the arbitrary sequence I chose can be seen by the naked eye by the difference in the frequency. The first time I took the video, the data varied widely, but I retook the videos using a tripod and turned the other lights off to help steady it. This actually might have hurt it in another way which I explain later.



To make things easier, I wrote a preprocessor to get the sum of the images for each video and converted it into a single vector. Since I had black on either side of the image data, I could plot each vector and clip out all of the non high blue frames. This made the data analysis cleaner.



Once clipped, you can look at the data a little better.

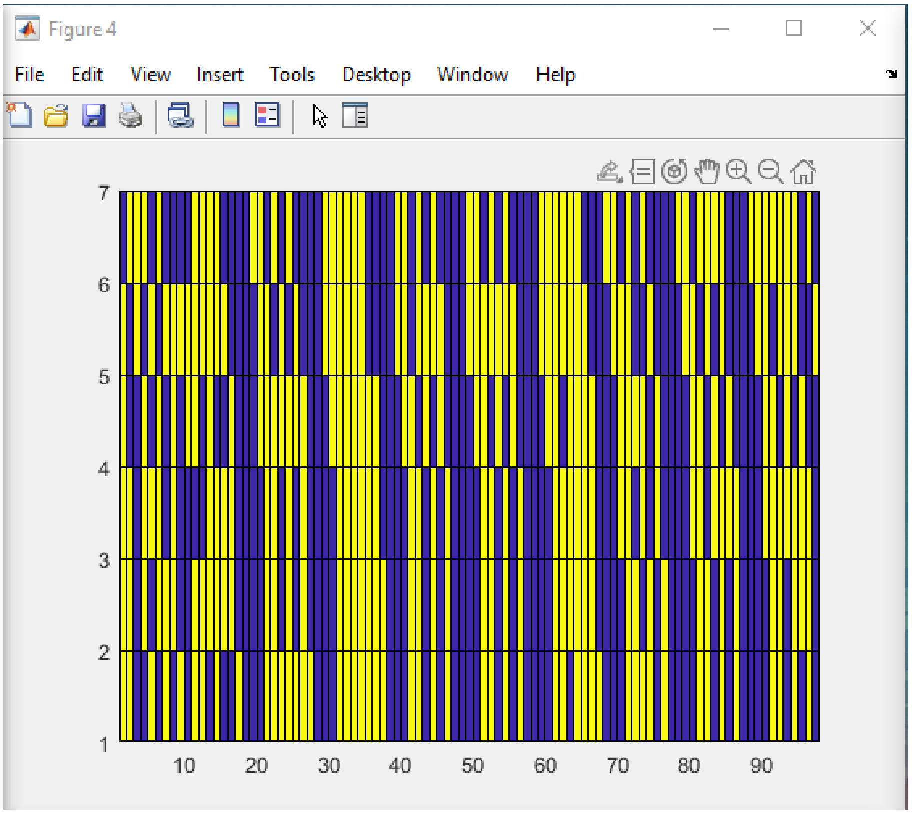


You can see that the bits with the alpha being 0.5 are easier to distinguish since they have a higher amplitude difference. One of the things I think factors into the performance of this system is the fact that the camera autofocus can change the exposure time automatically to get the best image. This might explain why on some of these, the common mode level changed part way into the capture. Also, because I turned off the lights and the screen brightness was so high, the autogain on the exposure may have caused the camera to expose too much which may have reduced some of my dynamic range. This might explain why the data for the alpha = 0.1 was so low.

As the camera moves further away, the error rates will increase because it is likely the camera is not able to distinguish between the change in the image's alpha since the amount of light captured per pixel coming off the screen starts to diminish with distance.

I do a with a window of 6 and choose either 20Hz or 30Hz depending on which one is higher to set the bit as 1 or 0 based on how I coded it into my output video sequence.

I didn't quite understand the pooling for candidate bits so I did a manual test by shifting the frames by one 6 times and plotted it. I then chose the frame shift that produced the best output. For example in this figure representing the data for alpha = 01 and distance = 120cm, with purple being 1, I can see that a frame shift of 2 seems to get me the best result.



After finding the best shift manually for each distance/alpha, I can then just run through the FFT in chunks of 6 frames. This produces the output below. I've highlighted a "perfect" sequence transmission with it decoding 1100101011 and it repeating several times successfully.

Comparing between alpha = 0.1 and 0.5. You can clearly see that alpha = 0.5 had a much lower bit error rate with the highlighted sequence representing 1 sec of data and that data being successfully repeated several times at multiple distances. Alpha = 0.1 has much higher errors as you can see with the areas that output a lot of 0s. This is clearly because the higher alpha difference allows for much more distinct peaks and valleys which leads to a more accurate FFT.

As far as distance, looking at alpha = 0.5. Distance didn't seem to affect it nearly as badly. For alpha = 0.1 distance actually seemed to help it a bit. I believe this was a consequence of my camera's focus affecting the data. If you look at the data from my previous plot, you can see that for alpha = 0.1 at distance of 20 and 40cm, the absolute value of the image seems to shift part way through the recording even though I had the camera mostly stable which I think could be caused by an autofocus attempt. I would expect that with distance, external factors start to matter more which can cause accuracy to begin to diminish with distance.