ECE 437

Sensors and Instrumentation

Final Project Report

Real-Time Object Tracking System

Yicheng Zhou (yz69@illinois.edu)

Qiran Pang (qpang2@illinois.edu)

TA: Bill Yang

November 30, 2024

1. Introduction

We developed a real-time object-tracking system with the sensor board for the ECE437 final project. With the acceleration/magnetic sensor and image sensor incorporated with the FPGA, we can acquire at least 20 frames per second from the image sensor while updating the 3-dimensional acceleration/magnetic measurements at least 120 readings per second. With images acquired from the sensor, we achieved object tracking using the KCF (Kernelized Correlation Filter) tracker such that after selecting an object with a rectangle region of interest in the beginning, when the object moves to the side, the board will detect and rotate to center the object again in the camera.

2. Spatial and temporal noise of image sensor

The image sensor has two types of noise: spatial and temporal.

Spatial Noise

Spatial noise is the variations in pixel values across the sensor at a given moment when illuminated uniformly, defined by the standard deviation normalized by the mean of all pixels. To measure it, we show pictures of pure color on my phone and put them in front of the image sensor to make sure the sensor is illuminated uniformly. Then, we take a frame and calculate its standard deviation and mean, therefore getting std/mean as the spatial noise. We test it repeatedly several times to calculate the average and we get 0.9966 spatial noise with a pure green picture.

```
if i == 50:
    print("standard deviation: ", np.std(arr))
    print("mean: ", np.mean(arr))
    print("spatial noise: ", np.std(arr) / np.mean(arr))
    print("SNR: ", 20 * np.log10(np.mean(arr) / np.std(arr)), " dB")
```

Temporal Noise

Temporal noise is the variations in pixel values for the same pixel across multiple frames when the illumination of that pixel is constant. We measured it by taking several frame readings and calculating the standard deviation of the intensity of the row-50 col-50 pixel of the frames as we did for lab 10. After repeating measurements and taking the average, we get a temporal noise value of 0.5745.

```
intencities_50_50 = np.append(intencities_50_50, arr[50][50])
```

```
print("temporal noise: ", np.std(intencities_50_50))
```

```
spatial noise:  0.9966390269568716
temporal noise:  0.574456264653803
```

3. SNR (signal-to-noise ratio) of image sensor

The signal-to-noise ratio of the image sensor is calculated by equation:

$$\mathrm{SNR} = 20 \times log_{10}(\frac{\mu_{signal}}{\sigma_{noise}})\ dB$$

μ_{signal} = mean intensity, σ_{noise} = noise standard deviation

Therefore, we made measurements by showing a pure white image in front of the image sensor and measuring the intensity. We got μ_{signal} to be 115.688 and σ_{noise} to be 28.059. Therefore, we can calculate the SNR to be 12.3042 dB.

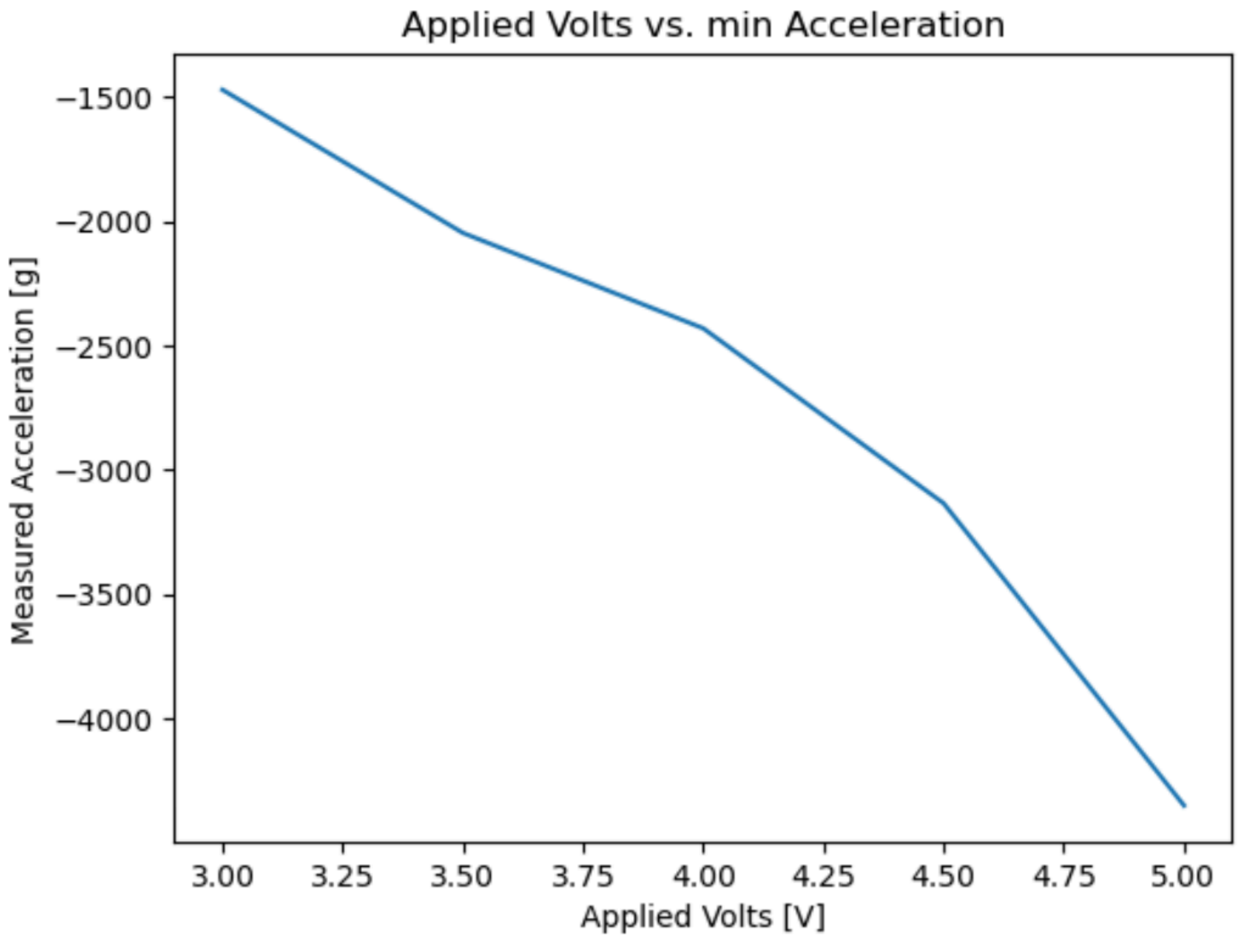
4. Acceleration/magnetic sensor noise

```
print("x acceleration reading noise: ", np.std(x_a_list))
print("y acceleration reading noise: ", np.std(y_a_list))
print("z acceleration reading noise: ", np.std(z_a_list))
print("x magnetic reading noise: ", np.std(x_m_list))
print("y magnetic reading noise: ", np.std(y_m_list))
print("z magnetic reading noise: ", np.std(z_m_list))
```

The calculation of acceleration/magnetic sensor noise is simply the standard deviation of multiple measurements while the acceleration and magnetic field stay the same. We chose to take 100 measurements to calculate the noise and we got the following result:

x acceleration reading noise: 56.06853
y acceleration reading noise: 63.85904
z acceleration reading noise: 77.36062
x magnetic reading noise: 1.46027
y magnetic reading noise: 1.42464
z magnetic reading noise: 1.43175

5. Motor acceleration under different applied voltage



The acceleration is is negative direction thus we can see it negatively trends with increasing supply voltage, from which we can derive roughly $|a| = V_{supply} \times 1500 - 3000$

6. Tracking algorithm and its accuracy

The algorithm is implemented on the CMS300 image sensor with default exposure time, thus the accuracy and reliability of the algorithm is significantly impacted by the contrast of the desired target and background (brightness, distance). With the required light source as example, as long as no sudden significant movement occurs, the algorithm can accurately track the object. Furthermore, even in the event that the target is out of frame or moved too fast that the algorithm lost track, it can generally regain tracking provided with a few static frame.