Live Image Intensity Tracker

Yazan Sawalhi†
Mechanical and Aerospace Engineering
The George Washington University
Washington DC, USA
ysawalhi@gwu.edu

Abstract

This project presents an IoT solution designed to track image intensity and detect saturation issues in experimental imaging setups. A Raspberry Pi 4 is integrated with a PiNOIR camera to acquire live images and extract their intensity profiles while also evaluating saturation conditions. The intensity data is displayed via a cloudflare-tunneled Flask web server using Plotly to visualize the profile of the curve. Additionally, MQTT integration facilitates remote monitoring by providing quick notifications regarding image saturation and computed statistics such as the Full Width Half Maximum (FWHM). The primary goal is to streamline data collection processes and enhance image quality assessment while actively conducting experiments.

Keywords

IoT, Image Intensity, RaspberryPi, Flask, MQTT, Signal Processing, Research

I. INTRODUCTION

When taking a photo with a camera, proper lighting is a must in order for the photo to survive deletion. How much light is too much light will depend on various factors, but the key is to make use of the entire range of light that the camera is able to capture, which is dependent on the number of bits the camera has. There is no use to flood the scene with light, that will result in an entirely white image and will surely get deleted. This is where light intensity comes in. An image is made of a series of pixels, each assigned a light intensity value from 0 to 2 raised to the power of the number of bits (minus 1). The scenario where there is a completely white image is a result of saturation. All the pixel values of the image read the max intensity value, which means that there is information being lost from the scene being captured due to the amount of excess light. For everyday photography, turning off the flash and retaking the photo will do. But in the research landscape, this is an issue that can hurt the quality of one's data. This project attempts to address this issue in experimental settings.

II. BACKGROUND AND LITERATURE REVIEW

This project is inspired by a specific issue from the lab I work in. One imaging technique the lab deploys is plenoptic imaging, which uses a microlens array (MLA) to get multiple perspective (elemental) views of the object, which allows access to angular information at the expense of spatial information. The system currently in use has 7 elemental views with one in the center and a hexagonal array of views surrounding it. Camera alignment is crucial for getting the best quality data, however, with measurements on the order of microns, this is not a straightforward task. One indicator of an aligned system is the intensity profile between the elemental views. The view directly left of the center view should have a similar intensity profile as the view directly right of center. If that is not the case, minimal camera adjustments must be made to correct that error.

What makes this project ideal for this application is how it saves time and storage space. The current process for determining if the system is misaligned is by recording our data, exporting it from the software, and using an application like ImageJ to look at the intensity profiles. The project being proposed will cut that process down to running a code and searching for the webpage that displays all the relevant intensity data at that moment, which will help inform system adjustments on the fly. The plot will also reflect whether or not there is any saturation. This is also important for this application, as it is extremely difficult to detect if the images are saturated with the naked eye. This will show up after we export and analyze the data, making us frustrated that we have to redo the experiment.

I have worked closely with Steven Williams on his thesis about measuring the wall shear stress using Fourier Integral Microscopy [1]. This technique involves placing the MLA at the Fourier plane to achieve orthogonal elemental views that retain telecentricity. The area that gave Steve some trouble was when performing the Point Spread Function (PSF). In short, the result of the PSF will be used in a deconvolution scheme that removes the blurring that appears in the data. This involves shining a light through an incredibly small pinhole to create a point source and translating the camera axially towards it in order to characterize the axial and angular resolution. Where this becomes a problem in Williams' system is when there is misalignment with the camera relative to the pinhole, as even the slightest of errors will propagate with the translation of the camera. This causes, as Williams puts it, a "discrepancy in peak values" among the elemental views [1]. He puts this down to small misalignments, which can be used to explain any deviation between theoretical and measured values.

III. METHODS USED FOR IOT DEVICE IMPLEMENTATION

The profile of the PSF for our application is in the form of an Airy Disk shown in Figure 1. So the focus on the device will be on imaging an airy disk to report back any saturation as well as any relevant statistics. The most notable statistic is the Full Width Half Max (FWHM) of the curve, as it tells us the lateral resolution of our imaging system [2]. Additionally, a plot of the intensity profile will be displayed along with another version of the plot that underwent a gaussian fit to smoothen it out.

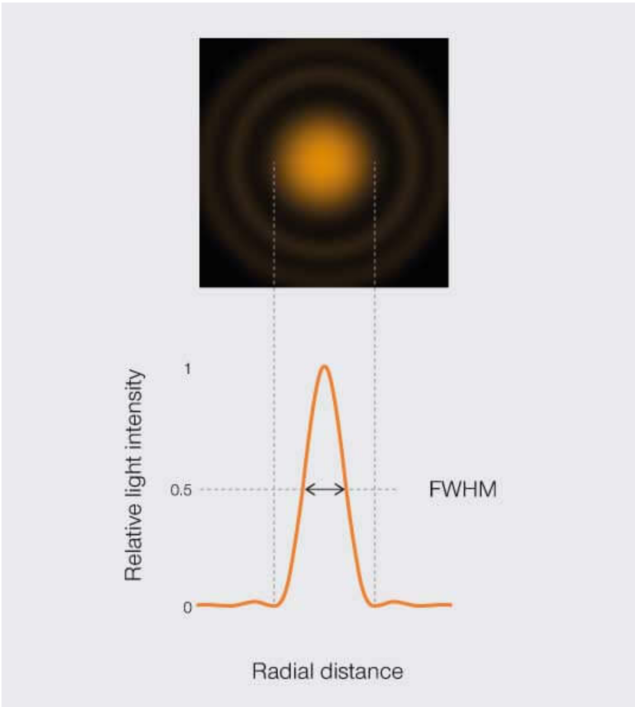

Figure 1: The profile of the image is an Airy Disk [3]

A. Hardware Components: The device is light on hardware, as it only incorporates two components. The schematic is shown in Figure 2.

- **Raspberry Pi Model 4B:** Processing unit that allows the code to interact with the sensors.
- **PiNOIR Camera:** For taking the image of the airy disk

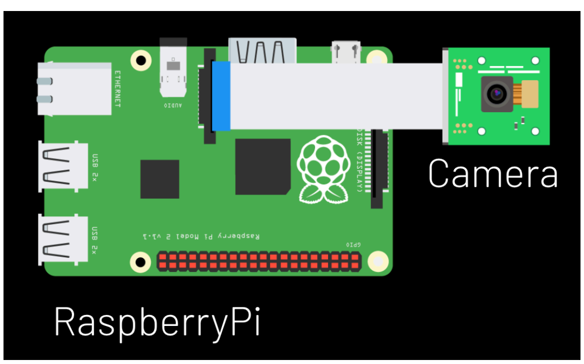

Figure 2. Schematic of the IoT device [4]

B. Edge Computing: This is key for an IoT device. The advantage of these types of devices is the proximity to data and their ability to perform calculations on the RaspberryPi without the need to send any data to the cloud. How the device achieves this is as follows:

- The OpenCV library is used to read images as a grayscale 2D array of intensity values.
- Linear algebra is used to grab a 1D slice across the center of the Airy Disk. This array is further truncated by grabbing every 80th value to reduce the required processing power.
- Signal processing for a cleaner signal using a Gaussian fit from scipy.
- Saturation is determined using a truth statement of whether the maximum intensity is reported on multiple occasions.
- If the "no saturation" criteria is met, the FWHM will be calculated using numpy.

C. IoT Layered Model: The project follows the 5-layered model as shown in Figure 3 and as detailed below:

- **Business:** Market for researchers that could do with on-the-fly data of their images
- **Application:** A localhost web server will be created using Flask with live plot of intensity values using plotly
- **Middleware:** RaspberryPi with Python (OpenCV library to read image as intensity values, numpy and scipy for computing)
- **Network:** Webpage will be tunneled to the internet using cloudflare and paho-MQTT will be used to publish relevant information on the profile (whether there is saturation and the FWHM)
- **Perception:** Camera picking up live images, where a 1D slice of the intensity values of the image are used as the data to be processed and plotted

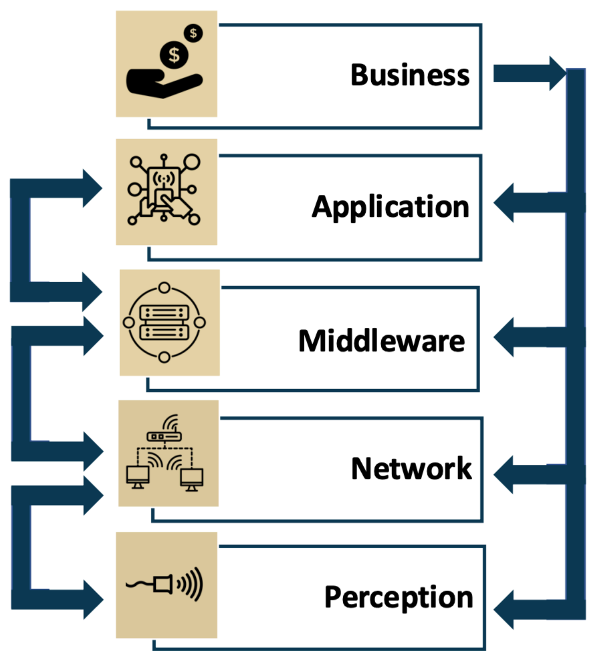

Figure 3. Architecture model used to describe the IoT device [5]

IV. DISCUSSION OF YOUR IOT DEVICE AND RESULTS GENERATED USING THE DEVICE

Using only a PiNOIR camera integrated with a Raspberry Pi, a device was created that is capable of capturing, analyzing, and presenting image intensity data. A python script was developed that creates a Flask server where Plotly can plot intensity profiles for quick visualization before being tunneled out to the internet using Cloudflare. The MQTT service effectively reported saturation conditions and the FWHM instantaneously. This gives researchers quick insights into their experimental setup that guide the necessary adjustments without the hassle of exporting and processing data outside the experiment. Results of the plots and MQTT messaging are shown in Figure 4.

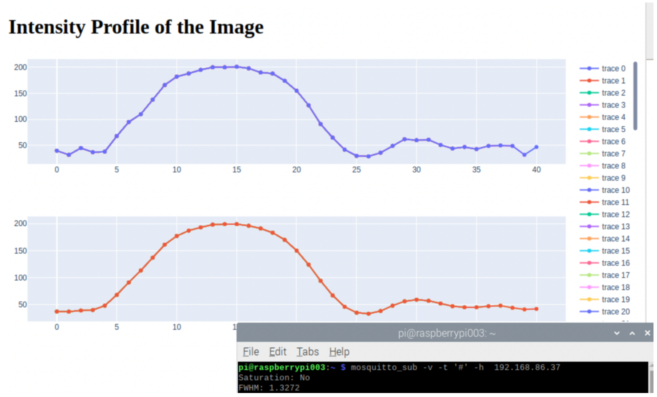

Figure 4. Results that include a Flask webpage showing the plot of the raw data followed by the plot of the data with a gaussian fit. In the bottom right corner, a terminal window is open that is subscribed to the MQTT channel that sends the saturation status and the FHWM.

Challenges included managing the latency of the camera to ensure quality images were being run through the code. This called for steady hands and practicing beforehand when performing the demo. Additionally, the image provided over 2000 data points for the one array, which was seen to have been overloading either Plotly or Cloudflare. The solution involved grabbing every 80th element of the array to reduce the array size to 41 data points. This was enough to still visualize the profile and extract the FWHM.

V. CONCLUSIONS, SUMMARY, FUTURE WORK

This project successfully developed an IoT solution capable of real-time detection and reporting of image saturation. The device is capable of advancing the work of Williams and other researchers by enabling the achievement of accuracy at higher levels. A contributor to errors in the data comes from the misalignment of a system. With the small scales that Williams is working in, it is impossible to detect misalignments and saturation with the naked eye. Devices such as this one can go a long way to assist researchers in their data collection.

This project serves as a prototype in tracking the intensity of the image in real time to allow for quick adjustments to the system without having to go through the laborious process of recording, extracting and analyzing the data before finding out that adjustments need to be made. Future developments could expand system flexibility to accommodate a wider variety of imaging applications. There is also potential for incorporating machine learning techniques in automated alignment corrections and advanced predictive analytics. With IoT devices, the possibilities are endless and they have the ability to aid many facets of life, including that of the research sector.

ACKNOWLEDGMENT

I would like to thank Dr. Kartik Bulusu for providing the necessary hardware and teaching me how to use the software for this project.

VI. REFERENCES

[1] S. Williams, "A Novel Compact Fourier Integral Microscope Design for Wall Shear Stress Measurements." Order No. 31556315, The George Washington University, United States -- District of Columbia, 2024.

[2] S. W. Hell, S. Lindek, C. Cremer, and E. H. K. Stelzer, "Measurement of the 4Pi-confocal point spread function proves 75 nm axial resolution," Applied Physics Letters, vol. 64, no. 11, pp. 1335–1337, Mar. 1994, doi: 10.1063/1.111926.

[3] Abberior Instruments GmbH, "What is resolution? Part one!," abberior.rocks, Sep. 17, 2022. [Online]. Available: https://abberior.rocks/knowledge-base/what-is-resolution-part-one/. [Accessed: Apr. 20, 2025].

[4] D. Wilcher, "How to Build a Raspberry Pi Camera," All About Circuits, Aug. 1, 2016. [Online]. Available: https://www.allaboutcircuits.com/projects/how-to-build-a-picamera/. [Accessed: Apr. 20, 2025].

[5] Bulusu, Kartik V. (2025, February). Week-6 Presentation: MAE 6291 Internet-of-Things for Engineers. [Course lecture notes, codes and presentations]. Department of Mechanical and Aerospace Engineering, The George Washington University.https://gwu-mae6291-iot.github.io/spring2025/