Tina Paper In Development

Your Name

October 7, 2024

Abstract

This is the abstract of your paper. Briefly describe the purpose of the research, the main results, and the conclusions.

1 Introduction

This is the introduction section where you introduce the topic and cite some papers [Mohammadi et al., 2024].

2 Methods

Describe your methods here. You can use equations like this:

2.1 Hardware Design

2.1.1 Mechanical Structure

The cartesian robot was purchased from an open source parts supplier, OpenBuilds. They sell various sets of gantry robots, motor controllers, and power supplies which can be conveniently purchased from one supplier. All metal components are cut and sized, and have very detailed assembly videos, an active community forum, and customer support to fall back on. The OpenBuilds ACRO system circumvents the need for most power tools, machining, electrical connectors, and technical knowhow that would otherwise be necessary to build a robust robot. All other parts were designed using SolidWorks and printed on an FDM 3D printer using ABS material. Mechanical assembly can be completed entirely with common hand tools and metric fasteners.

2.1.2 Computer

The computer which runs the graphical user interface (GUI), gantry control, camera streaming, and image stitching is the NVIDIA Jetson Orin Nano Developer Kit (Orin Nano). Using a computer which is compatible with CSI cameras such as the Raspberry Pi HQ Camera was essential to maintain cost effectiveness. The Orin Nano, like others in the Jetson line of products, has hardware accelerators for common GPU tasks such as image streaming, which is vital to retain CPU power for GUI tasks and robot control.

2.1.3 Camera

Unlike the Gigapixel or CaptuRING, the camera is a machine vision camera rather than a handheld professional camera. Choosing the Raspberry Pi HQ Camera provided many lens options, significant forum support, and includes an Obsolescence Statement that the camera will be in production until January 2030. Combined with a 180mm C-mount microscope lens, the images can reach above 20,000 dots per inch (DPI).

2.2 Software Design

All software was written in Python VERSION and a variety of packages for the GUI, image stitching, and calculations (Should I include all the packages in writing?). Care was taken to segment the software design into abstracted object oriented code.

2.2.1 Graphical User Interface

The GUI to control the machine launches from a Python script. Included are a viewer to see the live stream from the camera, buttons to jog the machine throughout the XYZ coordinate system, setting the size of samples, and running the process to capture and stitch images. The system was designed to allow for multiple samples to be prepared in a queue for bulk digitization.

2.2.2 Sample Digitization

The digitization of a sample can be reduced to a few processes - capturing a grid of overlapping images, image focusing, and image stitching.

Inside the GUI, the program is populated with the height and width of the sample as well as the height and width of the image. The user then navigates to the center of the sample and can choose to begin imaging the sample or add the sample to a queue for bulk digitization. This provides the machine enough data to calculate the necessary rows and columns for traversing the entire surface area of the sample while maintaining a level of overlap between adjacent images. Image overlap is what allows for images to be stitched together as there is nonzero error when the robot translates from one coordinate to another.

To stitch adjacent images, feature based image stitching from a Python package Stitch2D was used. The underlying functions were based on the processing libraries OpenCV and NumPy. Multiple other image stitching packages were tested but none were able to successfully piece together our samples. (***Sounds kinda bad to not have data on a comparison as well as not having a metric to see how 'well' an image is stitched... but this is non trivial. Can we get away with it?***)

One major difficulty with using the Raspberry Pi HQ Camera is the lack of a lens with auto-focus. An in focus image in this regime is dependent on the focal length of the lens set by the user and the distance from the sample. The microscope lens has a very small depth of field and goes from in-focus to out-of-focus in less than a 1mm step in the distance to the sample. Two cooperative solutions were implemented to achieve a focused image for the entire surface area of the sample - taking images from multiple distances from the sample and automatic control.

2.2.3 Image Stack Focusing

When the sample is placed on the table to be digitized, it is assumed that there will, at best, very slight variations in height across the surface area of the sample. A levelling table assists in reducing this error although it will never be nonzero. At each location (X, Y) on the grid of overlapping images, 11 images are captured in equally distanced Z steps across a 1mm range. Each of these images have a normalized variance score calculated - the maximum value represents the most in focus image [Mir et al., 2014]. In practice, stopping and starting the motor causes enough vibration to require upwards to two seconds of pause time before the motion blur stops to take each photo. Instead of stopping and starting each image, an additional distance is added to the Z-motion to allow the stepper to accelerate and reach constant velocity. At this speed, images are then taken with constant time steps instead of constant distance steps across the 1mm range. This eliminates the motion blur due to acceleration and dramatically reduces imaging time.

$$\frac{1}{MN\mu}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f(x,y)-\mu)^2$$

2.2.4 Automatic Control

While the image stack provides a basis to obtain an in focus image, it is statically limited to the 1mm range in Z-values. Instead of spending significant time guaranteeing that the sample is within the Z-value range across the entire surface area, a PID control algorithm takes the wheel [O'Dwyer, 2000]. The information previously calculated in the image stack is sufficient to inform the PID controller of how to adjust the initial Z-value for the next image stack. The controller is constantly trying to make the most in focus image be at the center of the image stack, index value 5. This allows for the most flexibility moving to the next location of the image grid.

3 Results

Present your results here.

3.1 Scans of Cookies and Cores

3.2 Functional Limits

4 Discussion

Discuss the implications of your results here.

4.1 Strengths and Opportunities

4.2 Opportunities for Improvement

5 Conclusion

Summarize your key findings here.

References

Hashim Mir, Peter Xu, and Peter Van Beek. An extensive empirical evaluation of focus measures for digital photography. page 90230I, San Francisco, California, USA, March 2014. doi: 10.1117/12.2042350. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2042350.

Fatemeh Sadat Mohammadi, Seyyed Erfan Mohammadi, Parsa Mojarad Adi, Seyed Mohammad Ali Mirkarimi, and Hasti Shabani. A comparative analysis of pairwise image stitching techniques for microscopy images. *Scientific Reports*, 14(1): 9215, April 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-59626-y. URL https://www.nature.com/articles/s41598-024-59626-y. Publisher: Nature Publishing Group.

Aidan O'Dwyer. A Summary of PI and PID Controller Tuning Rules for Processes with Time Delay. Part 1: PI Controller Tuning Rules. *IFAC Proceedings Volumes*, 33(4): 159–164, April 2000. ISSN 1474-6670. doi: 10.1016/S1474-6670(17)38237-X. URL https://www.sciencedirect.com/science/article/pii/S147466701738237X.