VIEWPOINTE

Bringing to you a vision, a vision for the future

Meet the Team

Tim
Kim

Omkar
Lonkar

Nathan
Niu

Krithika
Thanigaivelan

Ryan
Wenger

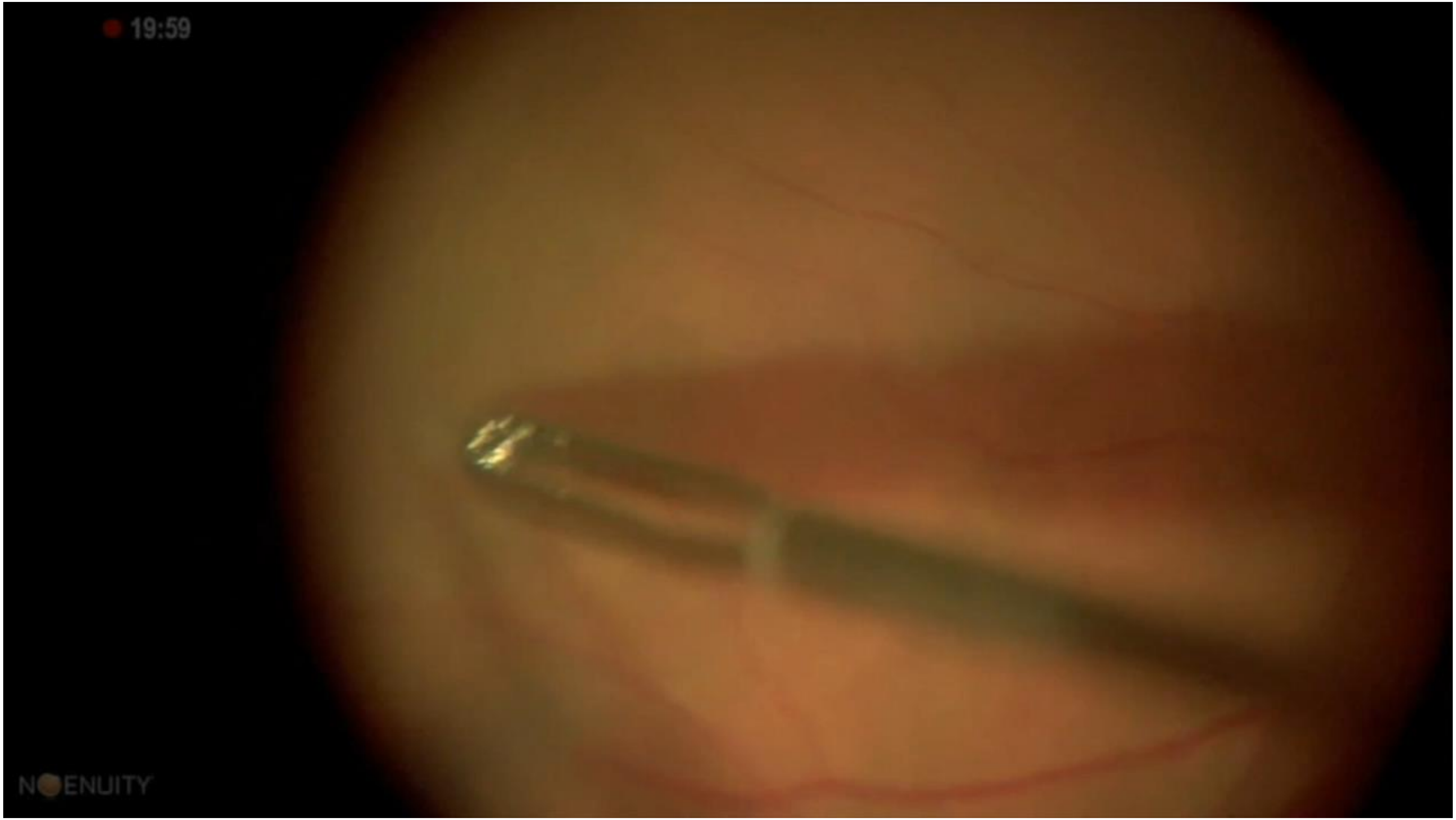
What are we trying to solve?

Problem Description

- Vitreoretinal (cataract) surgery requires accuracy, and can be affected by:
  - latency
  - image quality
  - ergonomics

Content warning:

Depictions of surgery and incisions


19:59
NGENUITY

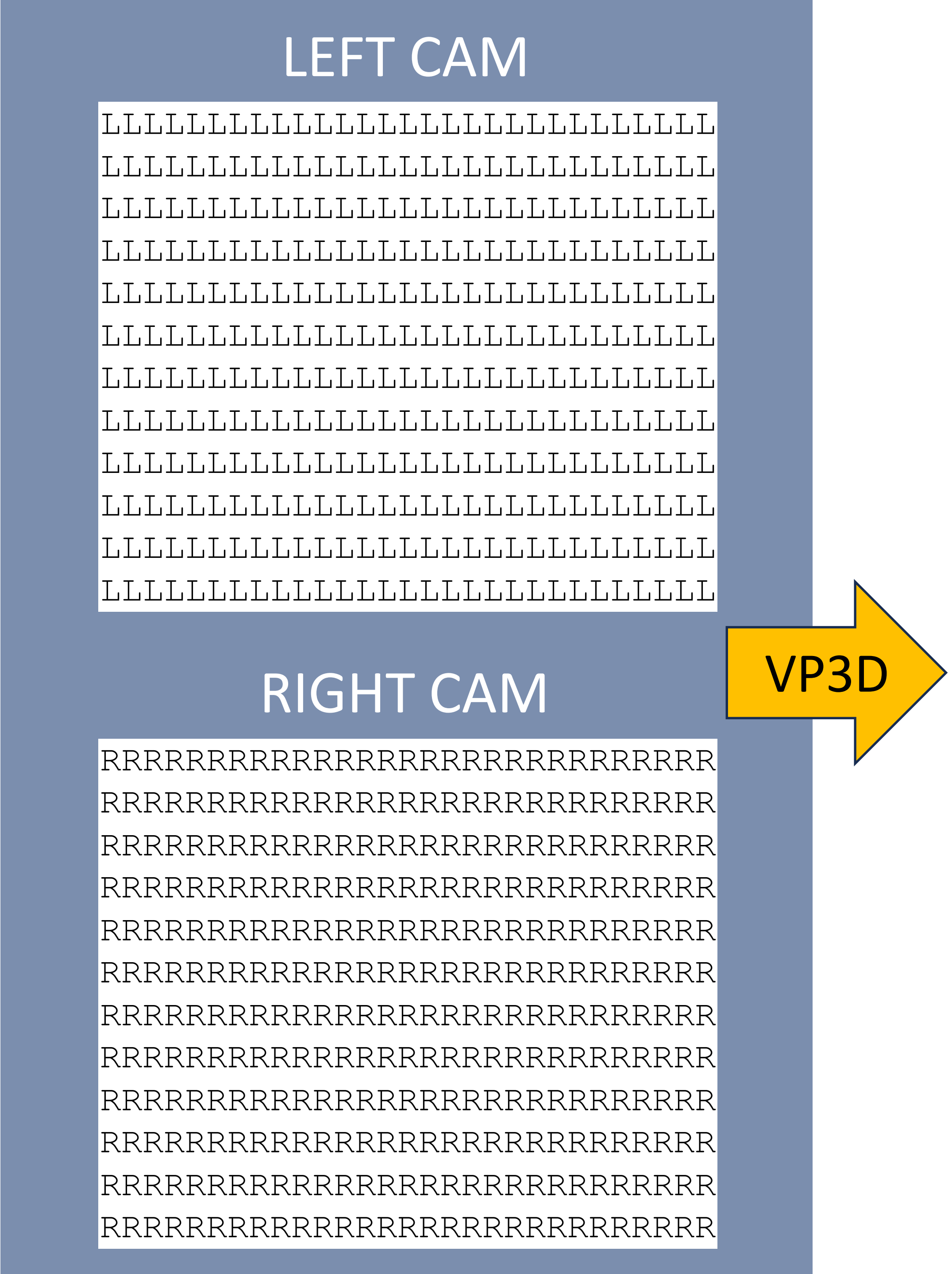
Overview

NGENUITY® provides eye surgeons with a 3D, HD video feed of patient's eye

Currently feeds are scaled/stitched together with discrete graphics processing

USB protocol used between camera and computer introduces latency

Our aim: cut the latency produced by the computer by processing in real time on an FPGA



LEFT CAM

```
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLLLLLLLLLLLLLLL
```

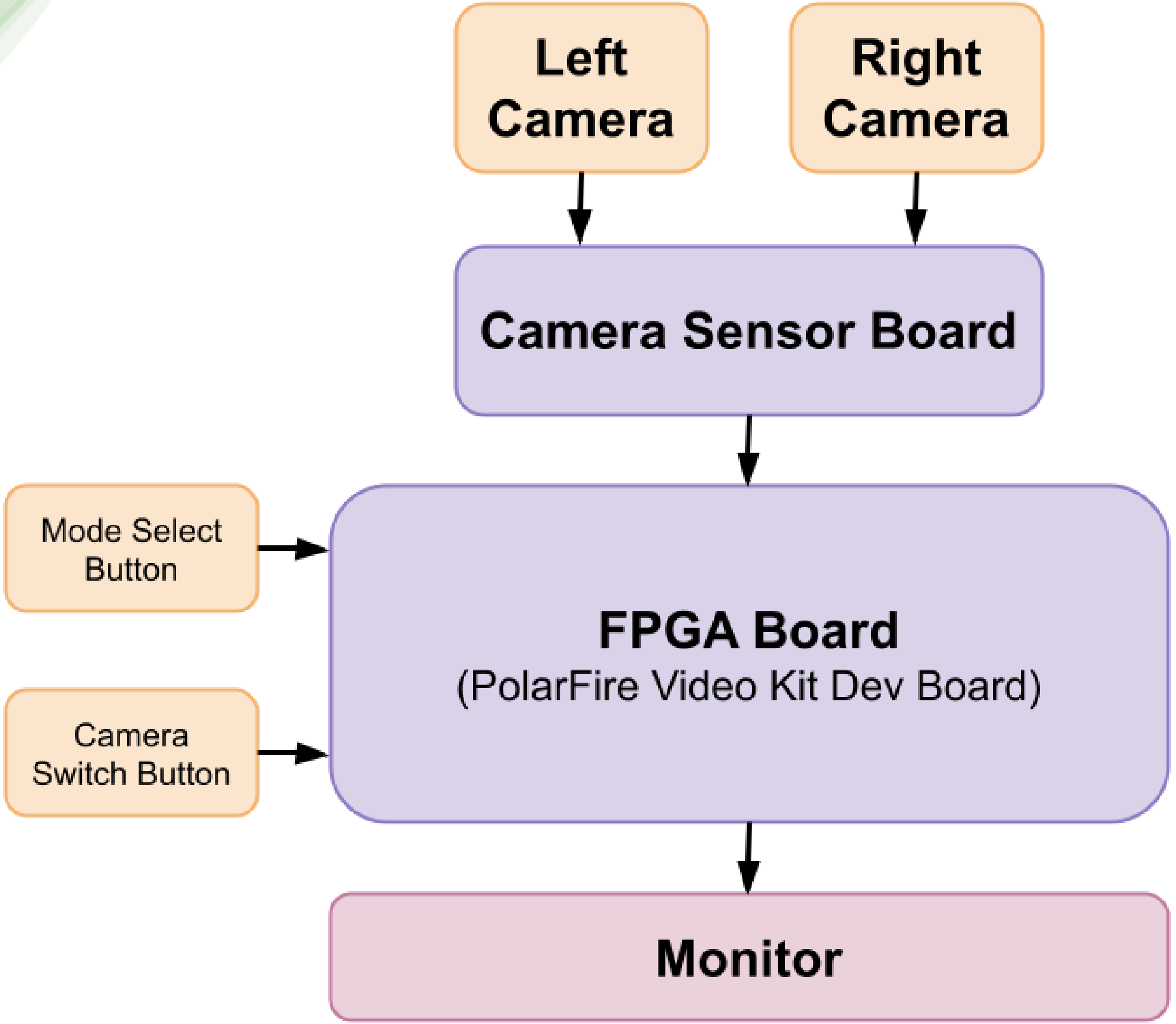
RIGHT CAM

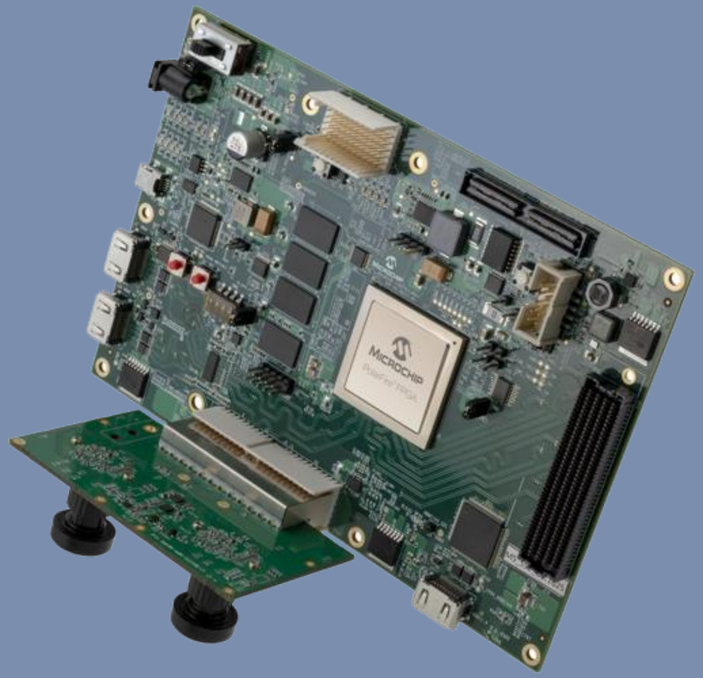
```
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

VP3D

Hardware + Components

Block Diagram – Hardware





PolarFire Video/Imaging Board

MPF300T-1FCG1152E FPGA

- 32 bit RISC-V CPU
  - Single core soft-processor
- Reprogrammable GPIO Buttons
- MIPI CSI-2, DSI, and CS interfaces
- HDMI 2.0, 1.4

High-level System Design Overview

All of the video processing is done in specialized “hardware” (Verilog IP cores)

The configuration of peripherals and communication protocols is done by a softcore RISC-V CPU on the FPGA

Software Breakdown

Software Flow – Microsemi Demo Code

- Analog data from both cameras sent through image sensor board to FPGA board and converted to binary data
- Data for both cameras buffered in DRAM simultaneously
- Data for both cameras read from DRAM and sent through video pipelining process simultaneously
- Images from both cameras overlayed into "Picture-in-Picture" mode
- Images sent to HDMI output

Software Flow – Challenges

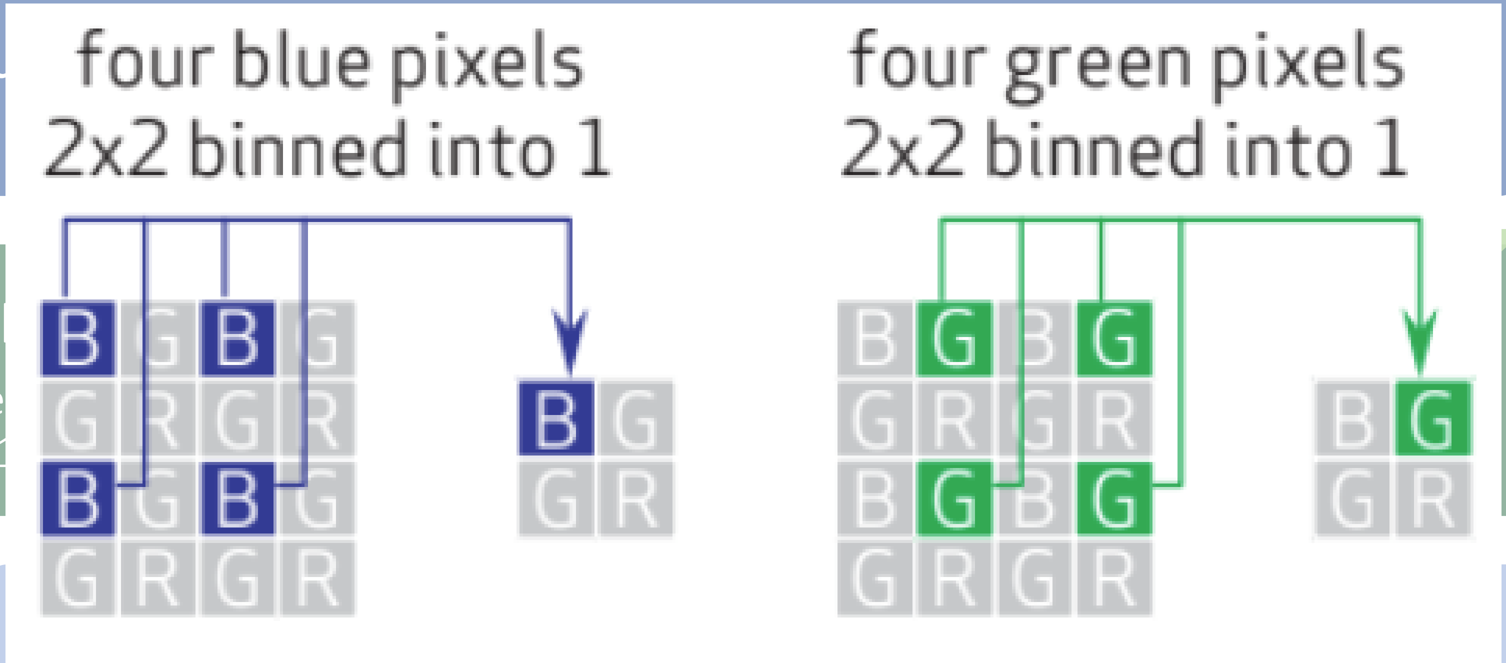
- Unfamiliarity with code base made it difficult to jump right in
  - Not all modules had documentation describing their function
- Had difficulty confirming if and where image data is buffered as well as the volume of data being processed at any given time
  - Unsure of where in the video pipeline to modify the image data without messing up the output image
- Long build times reduced efficiency

Software Flow – 3D Mode Implementation

- Implementation of new submodule following video pipelining process
  - Module logic determines the order at which input image data from both cameras is compiled in the output image
  - Basis to implement all 3D modes
- Change to how memory addresses sent to the arbiter are constructed
  - Necessary to fix the scaling and position of the two camera images for top-bottom and side-by-side
- Creation of several registers to modify variables for testing and to switch between the different 3D modes
  - Used with GPIO inputs to create mode-switching and camera-switching functionality

Breakdown – 4K to 1080p Downscaling

By default, ou



Video

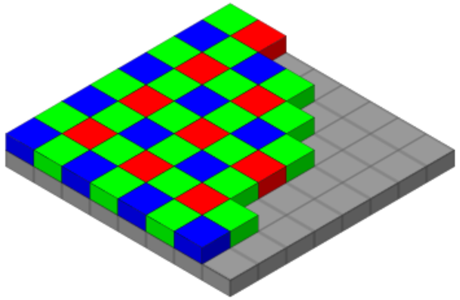
- This re
  1920x1

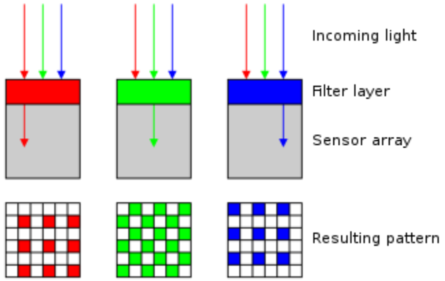
image into 1 pixel, then stich all of them together to downscale to 1080p

Image Processing Concepts

Bayer interpolation:

- Color filter array mosaic for arranging RGB color filters on a square grid of photosensors





Gamma correction:

- Used to encode and decode luminance values for video balance correction

Alpha compositing/blending:

- Used to combine two images (often used to create transparent effects)

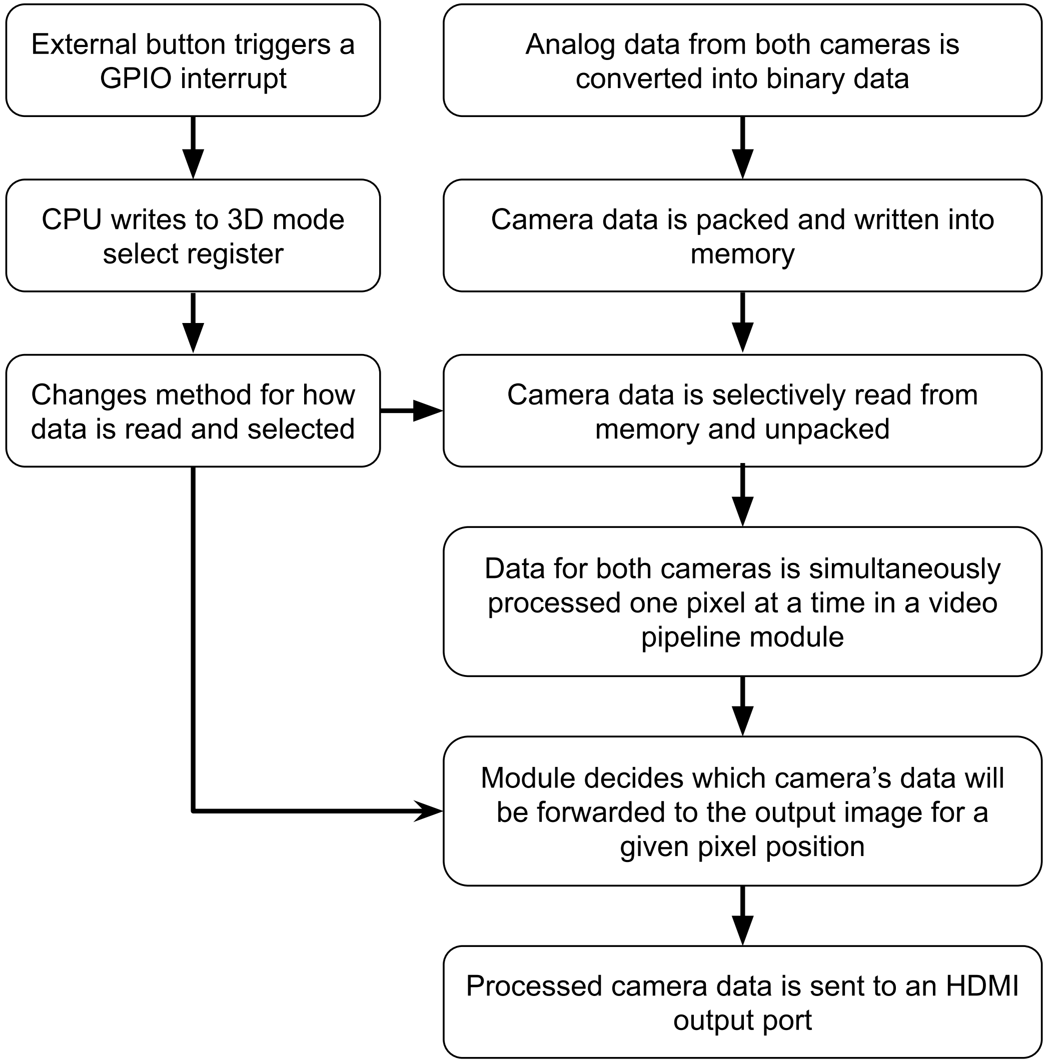
Mode Switching

Two buttons: one to toggle L/R polarity; one to cycle between 3D display modes

Each GPIO button is wired to one of the RISC-V's IRQ lines

ISR writes to GPIO block that's internally wired to the HDL, triggering a state machine update that alters the formatting of the video data

Design Flow



External button triggers a GPIO interrupt

CPU writes to 3D mode select register

Changes method for how data is read and selected

Analog data from both cameras is converted into binary data

Camera data is packed and written into memory

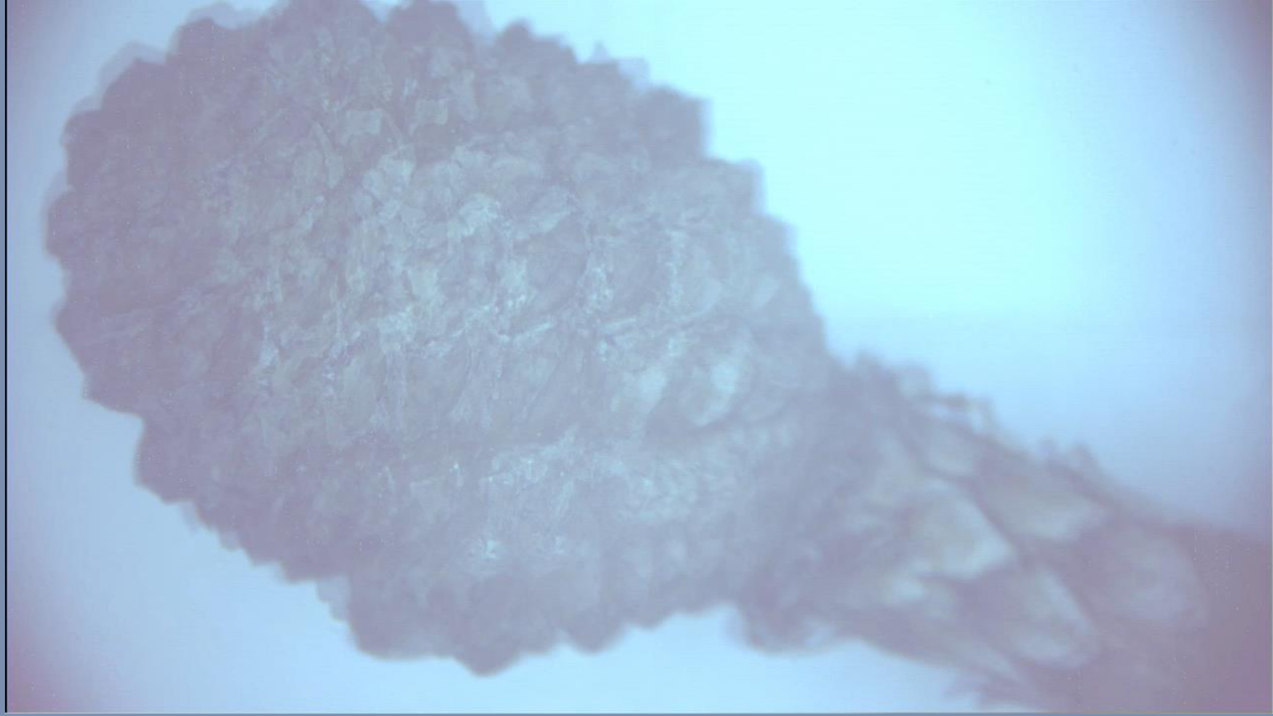
Camera data is selectively read from memory and unpacked

Data for both cameras is simultaneously processed one pixel at a time in a video pipeline module

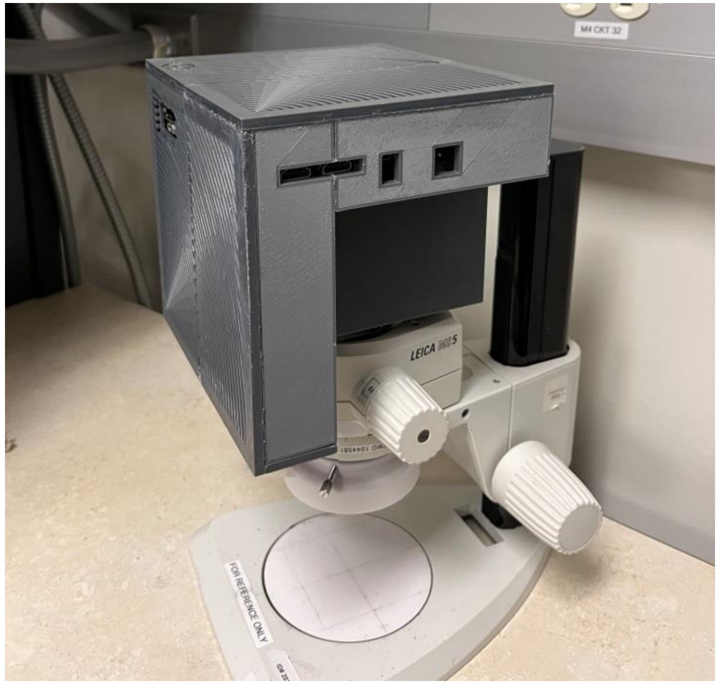
Module decides which camera’s data will be forwarded to the output image for a given pixel position

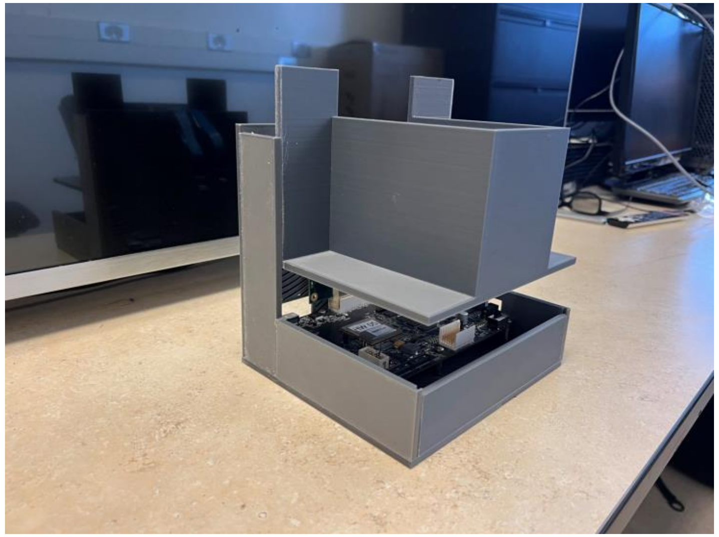
Processed camera data is sent to an HDMI output port

Demonstration



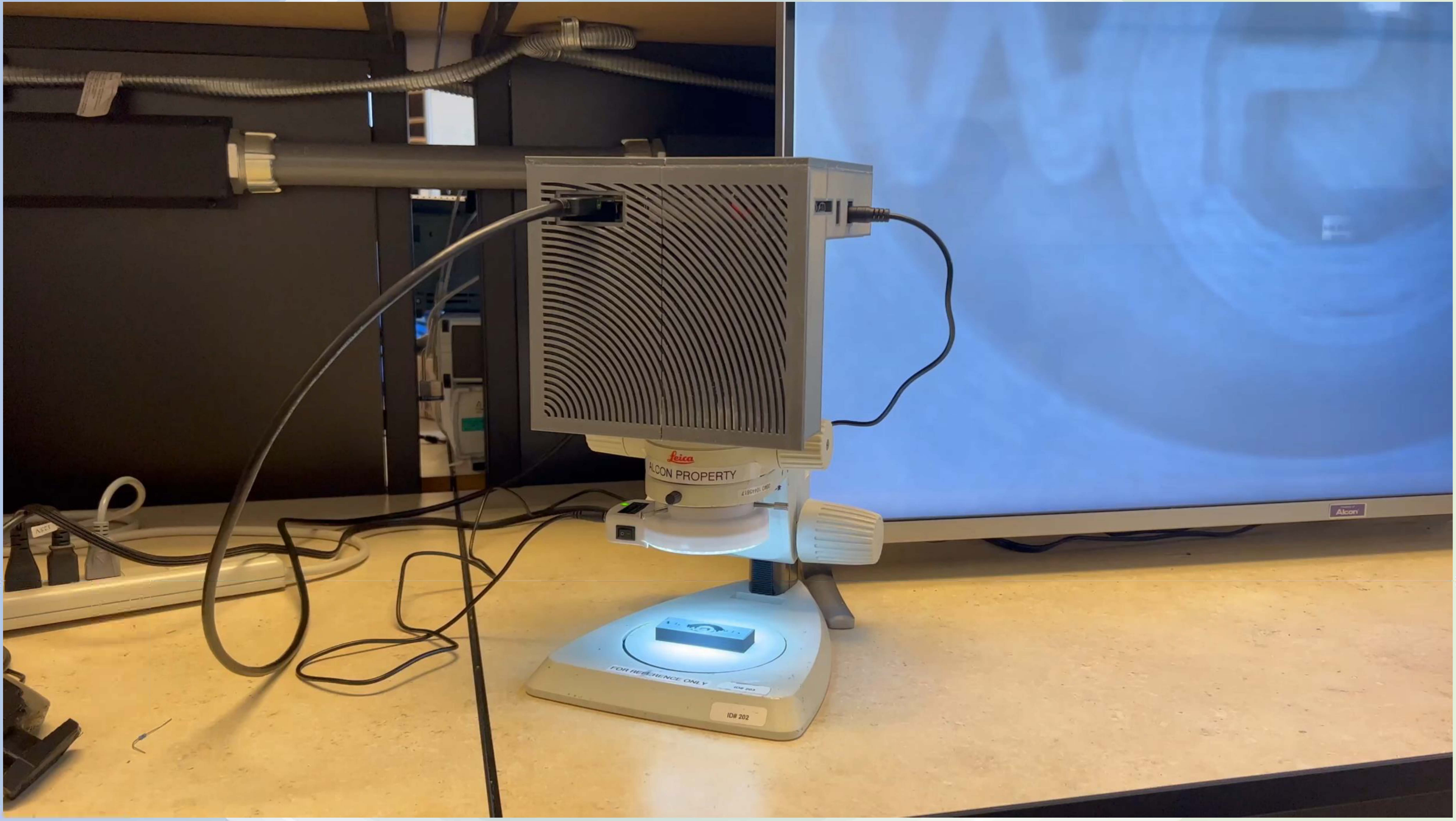
Final Assembly





The Enclosure

- 2-part 3D printed casing
  - Proper depth-of-field alignment for image capture sensors achieved using lens adapter
  - Can mount to most microscopes by changing metal ring at bottom of adapter
    - For demo purposes, we use the Leica MS5
  - Access to mode selection and camera switching buttons
  - Minimize light bleed and allow ventilation


Leica
ALCON PROPERTY
FOR REFERENCE ONLY
ID# 202
Alcon

Special Thanks To

Professor Yogananda Isukapalli

TA's Alex Lai, Jimmy Kraemer, and Venkat Krishnan

Alcon

Yuepei Hu

Jairo Hernandez