Vision

By Kevin Jaget (900), Lyra Solomon (1678), Anthony Demetrescu (3476)

Vision

Anthony Demetrescu
Mentor, 3476

Bio

- Member of 3476 Code Orange for 4 years
- Lead programming mentor of 3476 for 4 years

Hardware

- Bright green 1 Watt LED(s), using constant current driver
- USB camera (PCB mounted, interchangeable lenses)
- Jetson TX1
  - Use orbitty carrier to shrink footprint
  - Nano is cheaper with comparable performance

Overview of Vision Pipeline

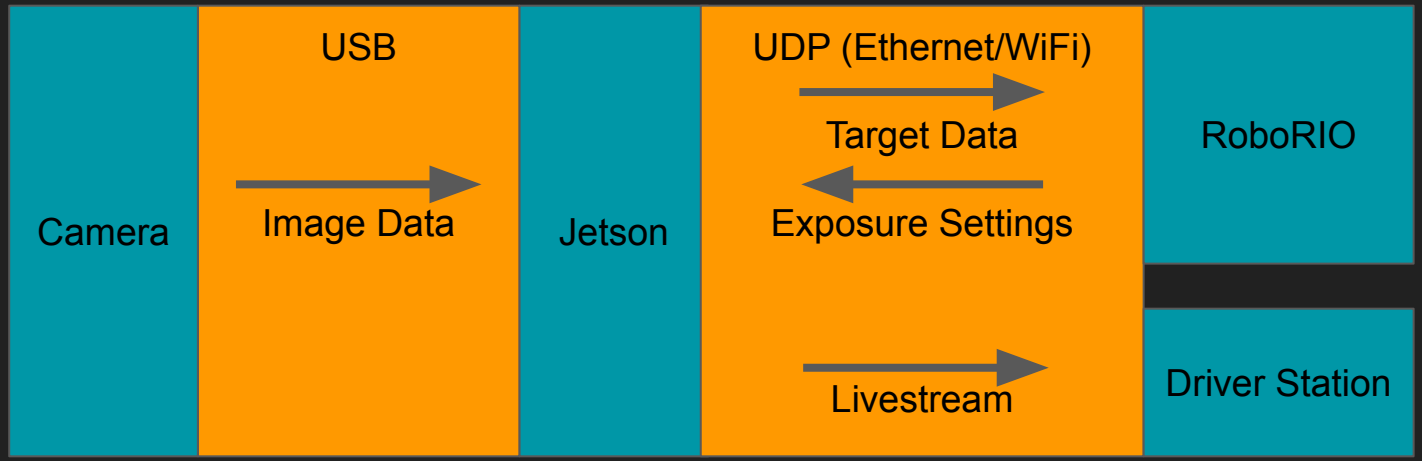
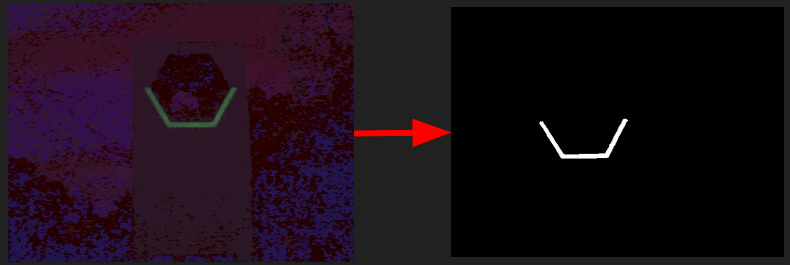


Image Processing - Gross Filtering

- HSV conversion and thresholding
  - Green hue
  - High saturation
  - Medium-high value
  - Output is a binary image
- Morphology
  - Works best with binary image
  - Gets rid of tiny granular noise
  - Faster than blur kernels
  - Works well unless parts of target are very thin



Opening

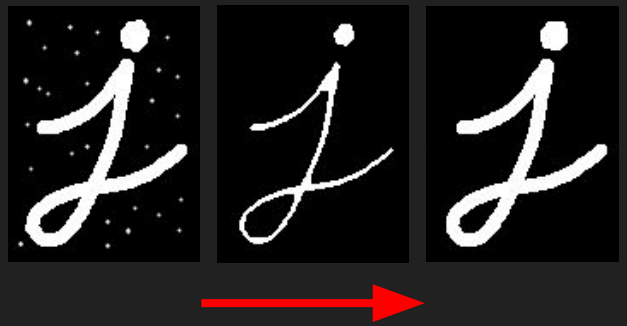
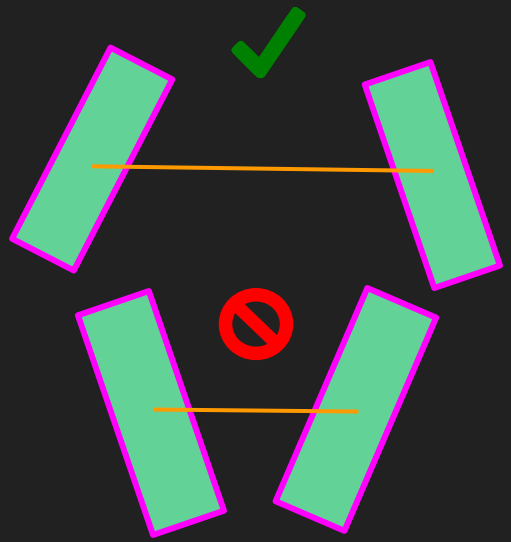


Image Processing - Target Extraction



- Finding contours
  - Creates a list of blobs
- Classification
  - Calculate metrics
    - Aspect ratio
    - Orientation (from moments or corners)
    - Area (from moments)
  - Get rid of blobs that are clearly not the target
    - Discriminate based on aspect ratio, orientation, very small, etc.
  - If target is multiple blobs (2019), figure out which blobs can feasibly make a target
    - Use distance from each other, relative size, orientation

Image Processing - Target Information

- Get position of center relative to camera
  - Helpful to convert to -1 to 1
- Estimate distance
  - Based on area or angle and robot height

Robot Processing - Using Target Information

- Turn camera position into angle using FOV
- Use distance to make a decision or compensate for camera not being at center of rotation
- Find location of target in space and move to point at it

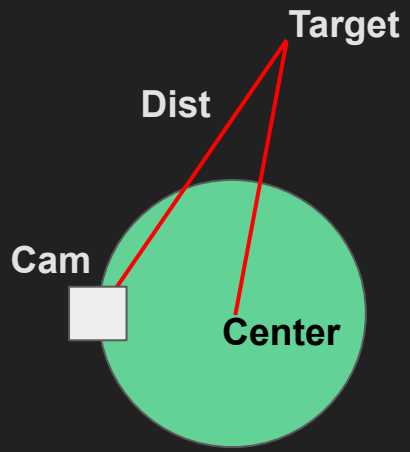
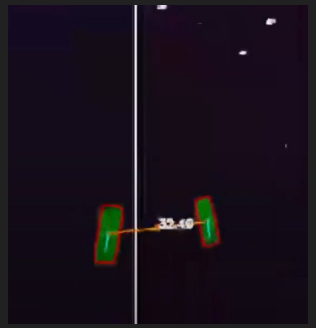


Image Streaming - Gstreamer

- Can use gstreamer to encode in H.265 for lower bandwidth requirements (especially with mostly black frames)
- Annotate before streaming to offer info to drivers
- One camera for both operator and vision usage by modifying exposure

Operator Exposure



Vision Exposure

Thanks!

Code: https://github.com/FRC3476/

Repos related to vision have vision in the name, e.g. FRC-2019-Vision,

FRC-2020-Vision

anthony@demetrescu.com

Vision

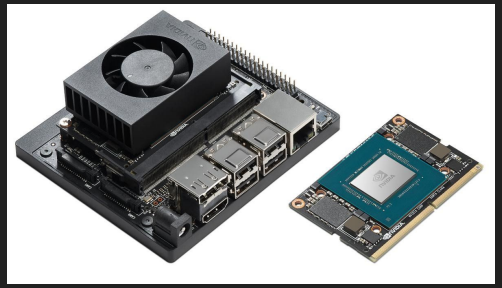
Kevin Jaget
Mentor, FRC900
support@team900.org

What the Zebracorns Have Been Up To

- Hardware updates
- Retroreflective tape targeting
- Machine learning for object detection
- Using vision results for automation

Zebracorns' Hardware Setup

- Jetson Xavier NX coprocessor
  - Next iteration of the Jetson platform
  - Offloads compute-intensive code from the roboRIO
- Stereolabs ZED2 RGB+depth camera
  - Gives a color image plus a distance in meters for each pixel
  - Lets us convert from image coords to real-world x,y,z displacements



Retroreflective Tape Targeting

- Detects both the high and low goal targets
- Uses our custom software, now 5+ years old
  - C++ with OpenCV
- Uses RGB+depth camera
  - Result is angle to target + range

Machine Learning Object Detection

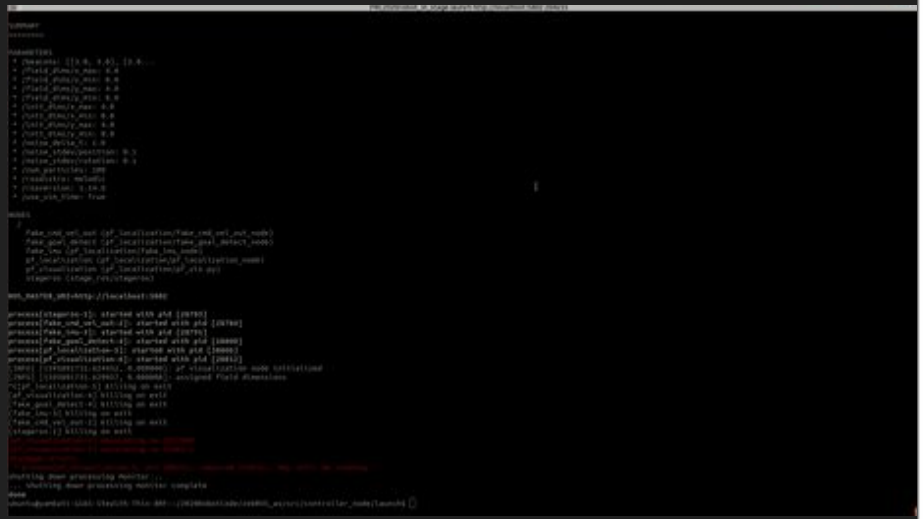
- Uses Tensorflow Object Detection toolkit
- 30+FPS detection of 37 classes of field objects
- Trained on dataset of 1500 images
  - ~14,000 labeled objects
  - Data set is open source -

    https://github.com/FRC900/tensorflow_workspace

- Read our vision white paper @ https://team900.org/labs!

Uses for Vision Data

- Automated shooter
  - Automatic selection of speed and angle
- Localizing robot on the field
  - Results from object detection
  - Plus map of object locations
  - Plus location history
- More to come?

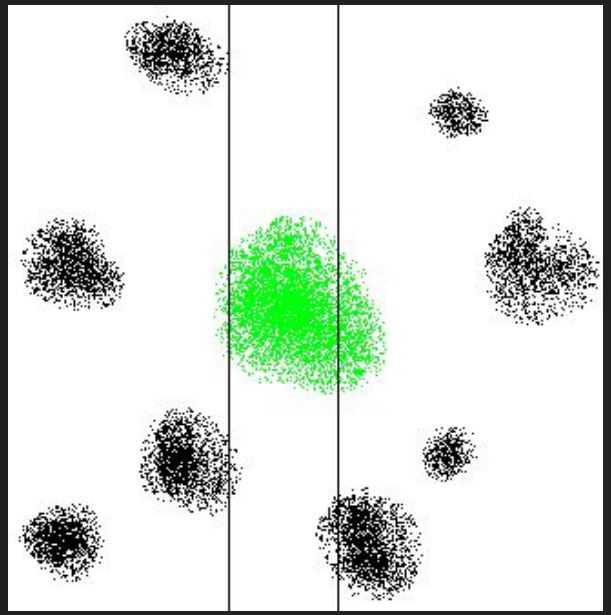


Vision

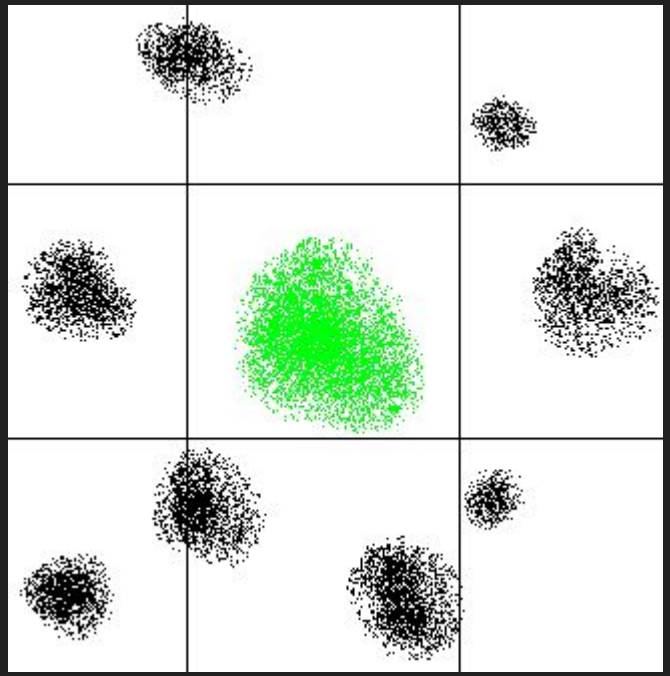
Lyra Solomon
Alumna, 1678

Detecting Targets

Potentially Buggy

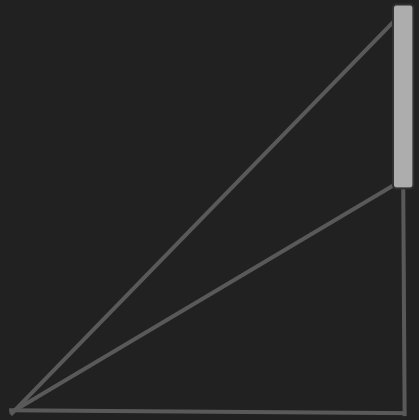


Stable



Multiple Filters

- Color—good first pass
- Size
- Location
- Shape



Use precise calculations rather than tuning cutoffs

Control Loops: Challenges

- Low sample rate
- High noise
- Lag is a problem
- Resonance is very bad

Control loops: Implementation

- Don’t use camera for feedback
- Record position over time
- Timestamp image data

Links

- https://github.com/frc1678/robot-code-public/tree/master/c2017/vision
- mailto:mailtolyra@gmail.com