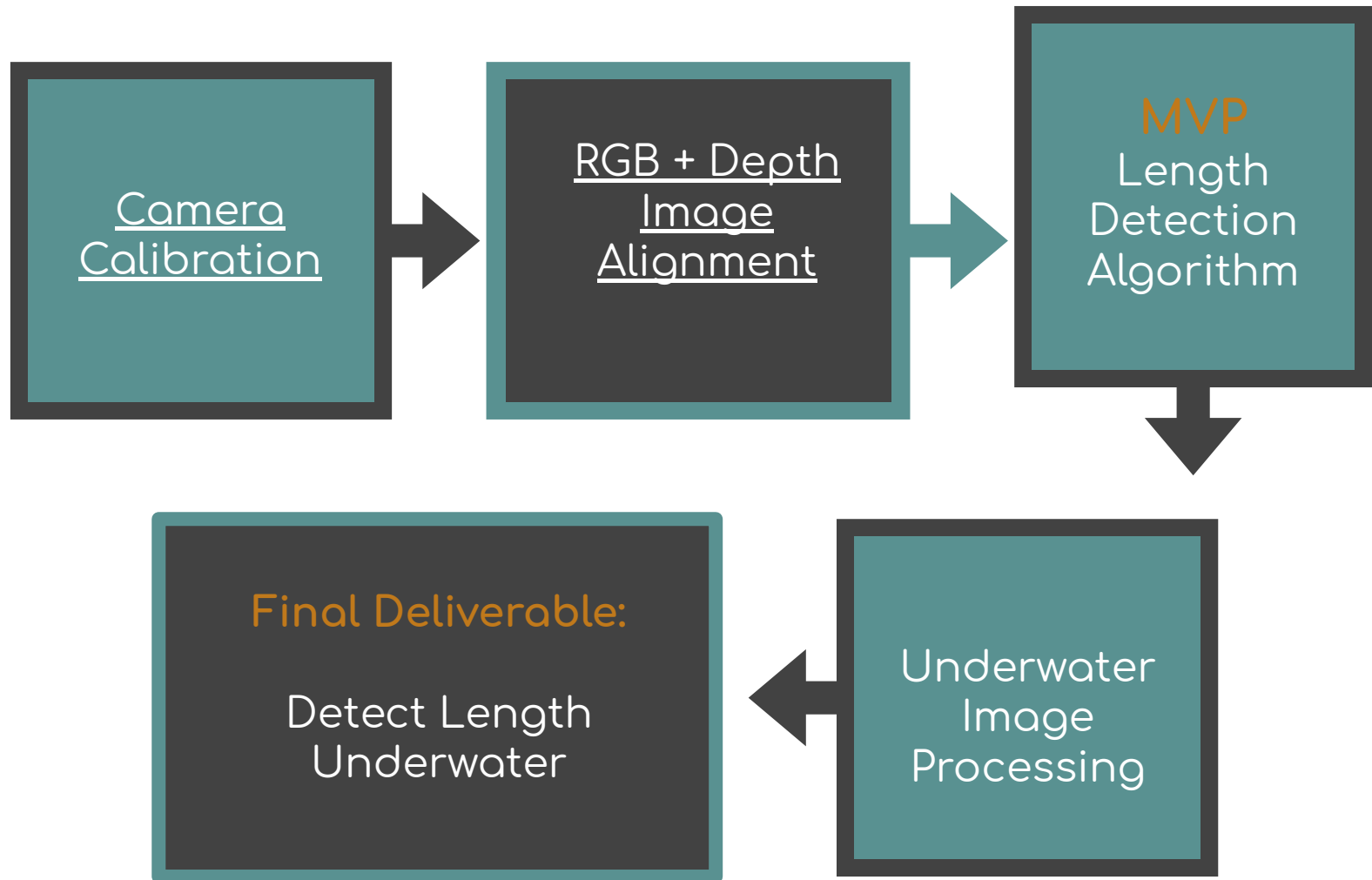


FishSense

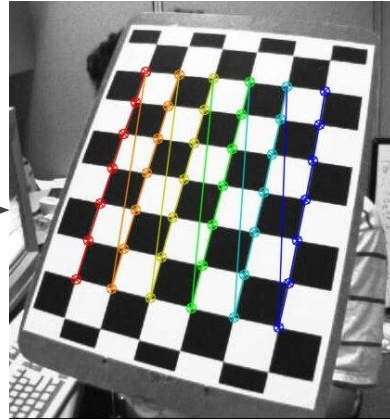
Presented by Emily Ferguson, Xilin Gao &
Zixiang Zhou







Camera Calibration

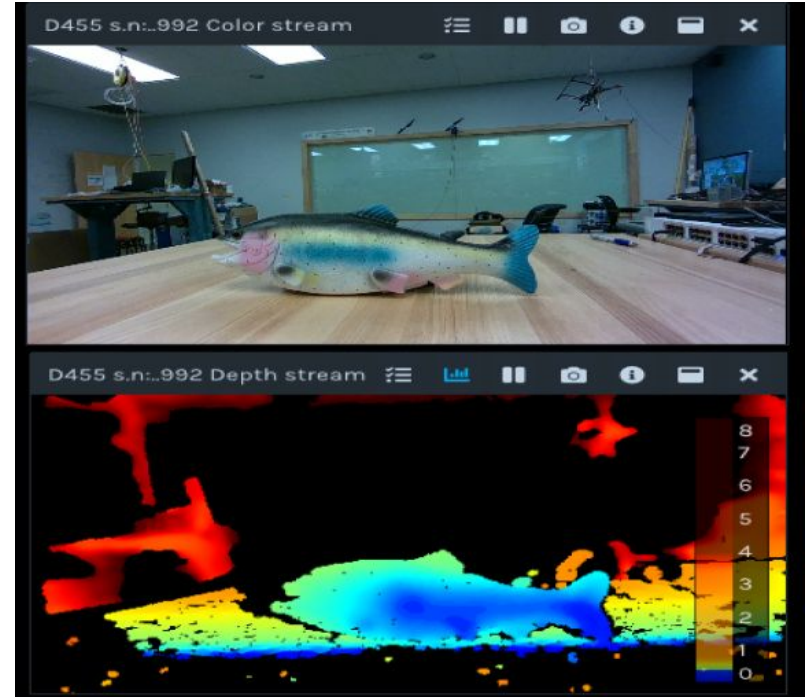


Site from openCV Camera Calibration

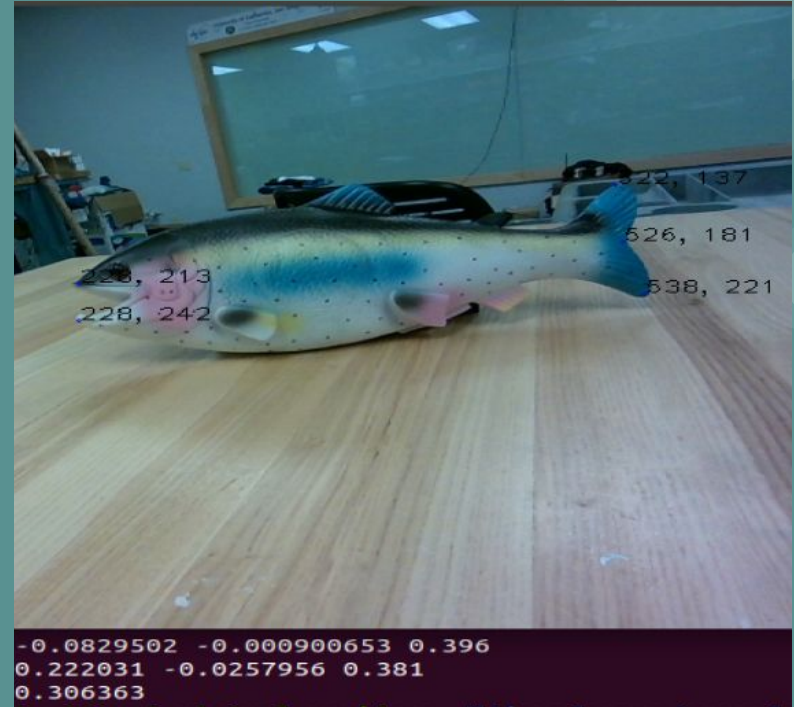
Align RGB & Depth Images

(1) Point-to-point mapping

(2) Create align filter



MVP: Fish Length Detection Algorithm



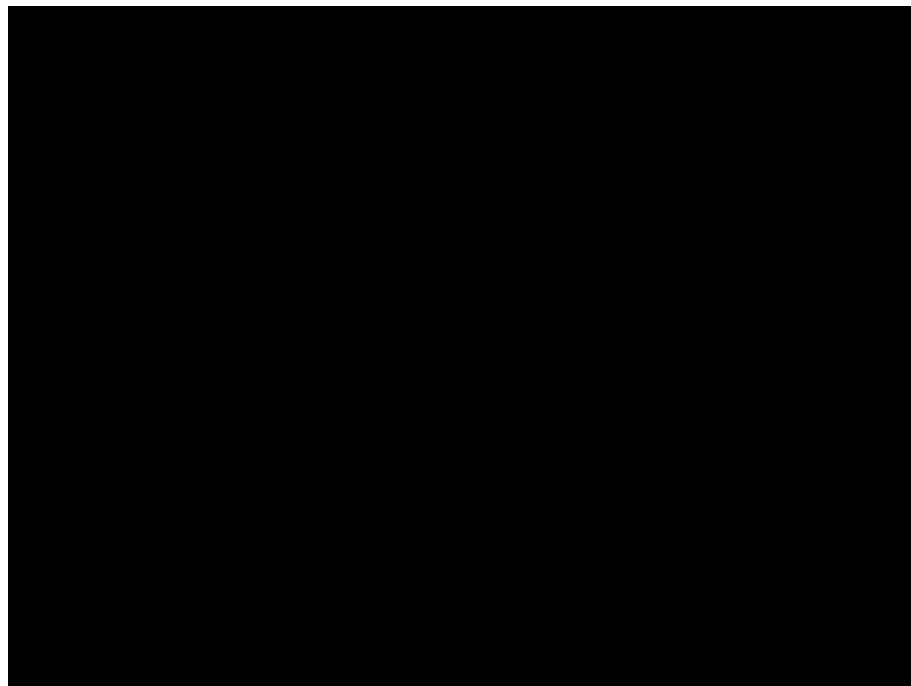
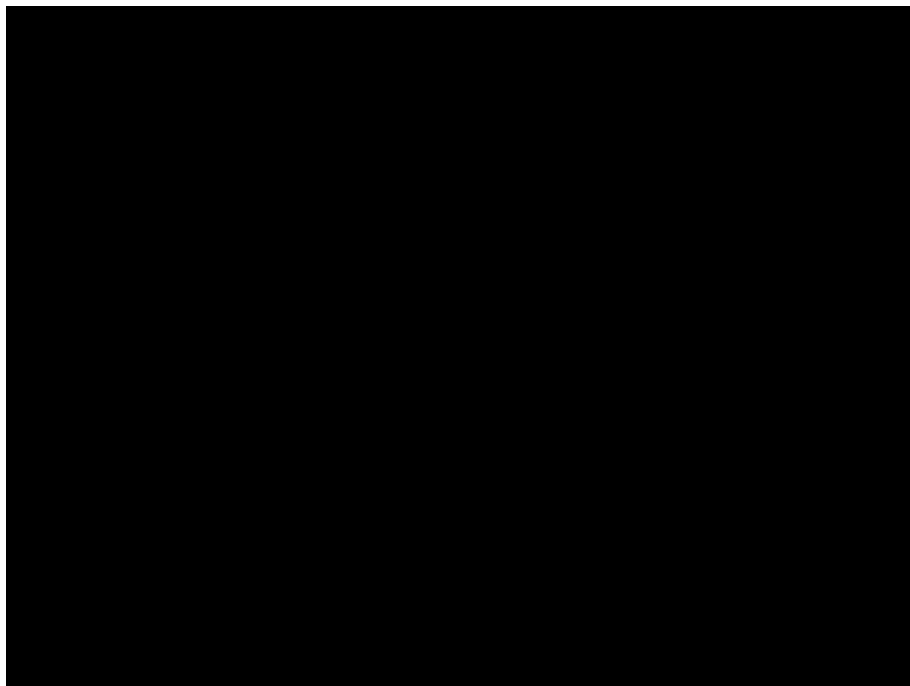


Length Detection Demos:

Manual

vs.

Automatic



Underwater Image Processing

HE / RGHS

1

CLAHE/
GC/ ICM

2

Dehazing

3

Gaussian +
Canny

4

Noise Filtration on underwater images



HE Image enhancement by histogram transformation (2011)

Noise Filtration on underwater images



CLAHE Contrast limited adaptive histogram equalization (1994)

Noise Filtration on underwater images



Relative Global Histogram Stretching Based on Adaptive Parameter Acquisition

Noise Filtration on underwater images



GC Gamma Correction

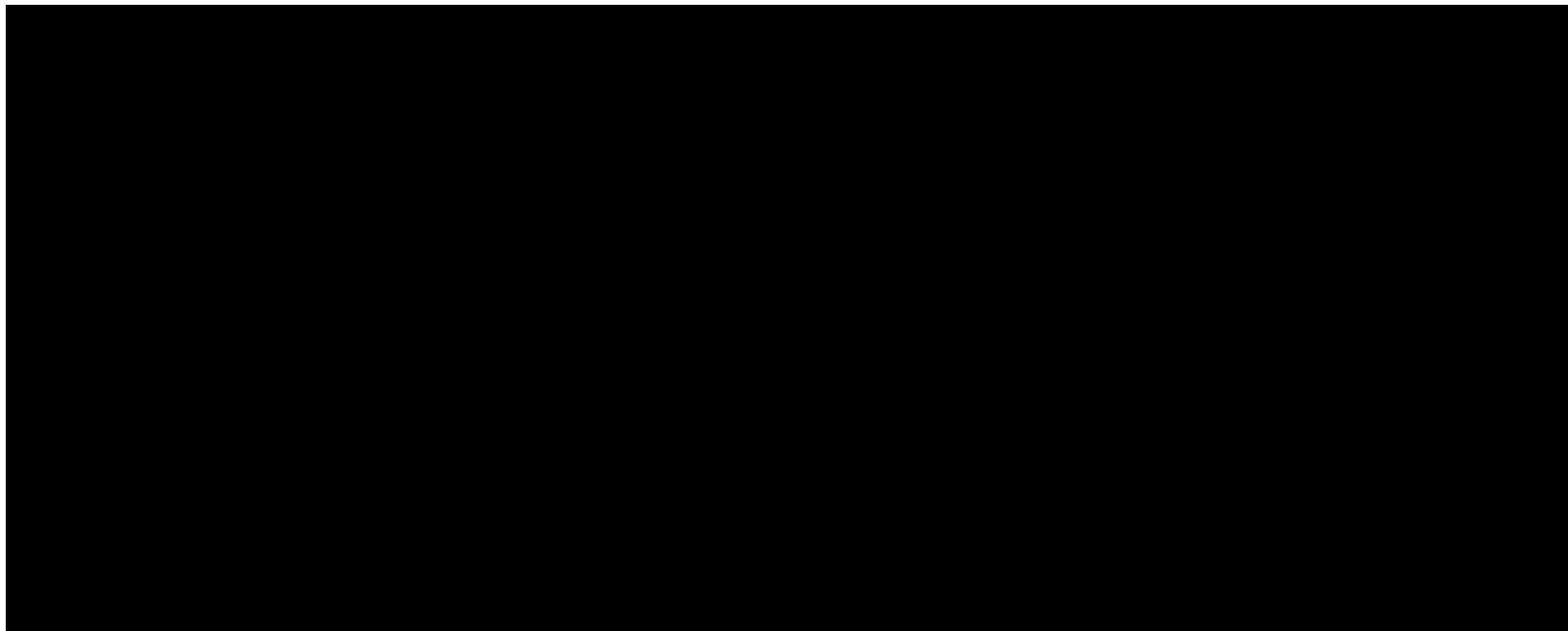
Noise Filtration on underwater images



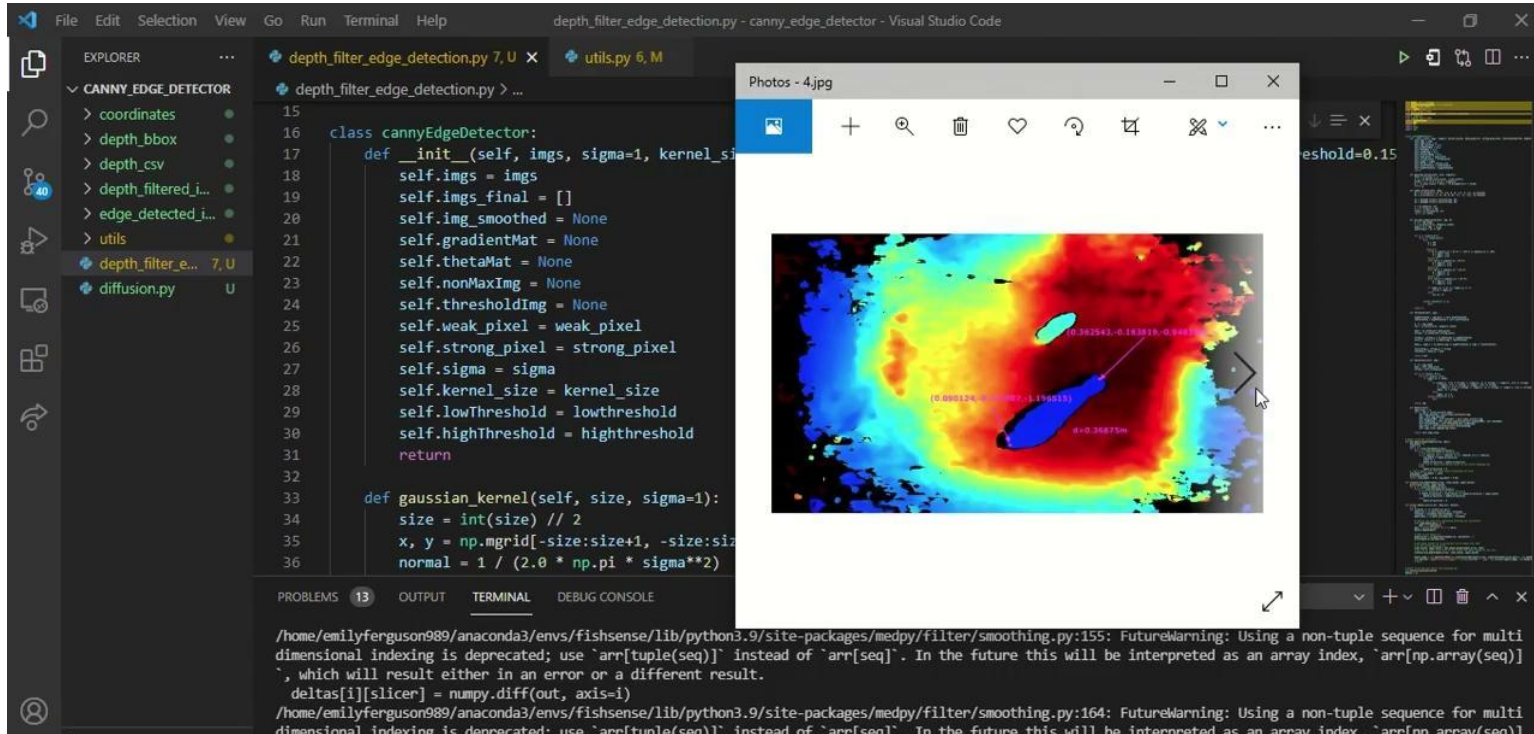
ICM Integrated Colour Model (2007)



Underwater Image Processing Demo - Underwater Dehazing:



Noise Filtration Demo - Gaussian Blur + Canny Edge Detector



The screenshot displays a Visual Studio Code editor window with the file `depth_filter_edge_detection.py` open. The code defines a `cannyEdgeDetector` class and a `gaussian_kernel` function. The class `cannyEdgeDetector` has an `__init__` method that initializes various attributes like `imgs`, `imgs_final`, `img_smoothed`, `gradientMat`, `thetaMat`, `nonMaxImg`, `thresholdImg`, `weak_pixel`, `strong_pixel`, `sigma`, `kernel_size`, `lowThreshold`, and `highThreshold`. It also includes a `gaussian_kernel` method that calculates a Gaussian kernel for a given size and sigma.

Overlaid on the editor is a window titled "Photos - 4.jpg" showing the result of the edge detection. The image is a heatmap where the edges of a fish are highlighted in blue and red, while the background is a mix of yellow and green. The fish is oriented diagonally, with its head towards the bottom left and its tail towards the top right. The edges are clearly defined, showing the outline of the fish and its internal structures like the eye and fins.

At the bottom of the editor, the TERMINAL pane shows the following output:

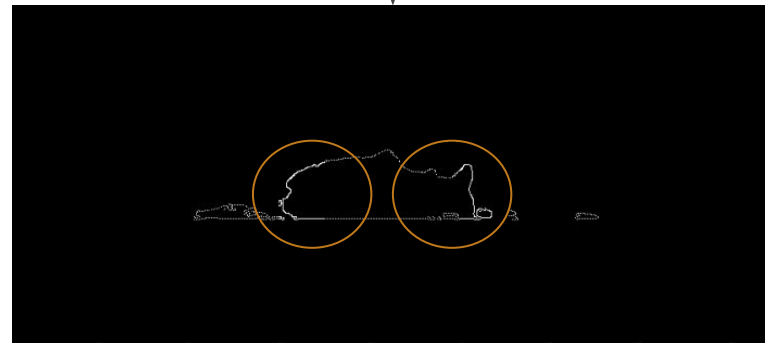
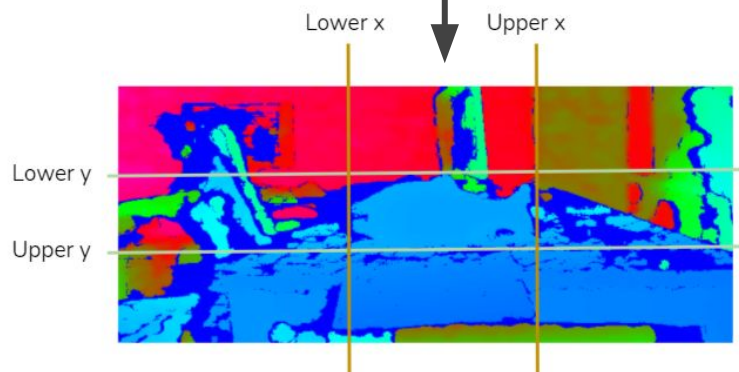
```
/home/emilyferguson989/anaconda3/envs/fishsense/lib/python3.9/site-packages/medpy/filter/smoothing.py:155: FutureWarning: Using a non-tuple sequence for multi-dimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.  
deltas[i][slicer] = numpy.diff(out, axis=i)  
/home/emilyferguson989/anaconda3/envs/fishsense/lib/python3.9/site-packages/medpy/filter/smoothing.py:164: FutureWarning: Using a non-tuple sequence for multi-dimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]'
```

Current Framework



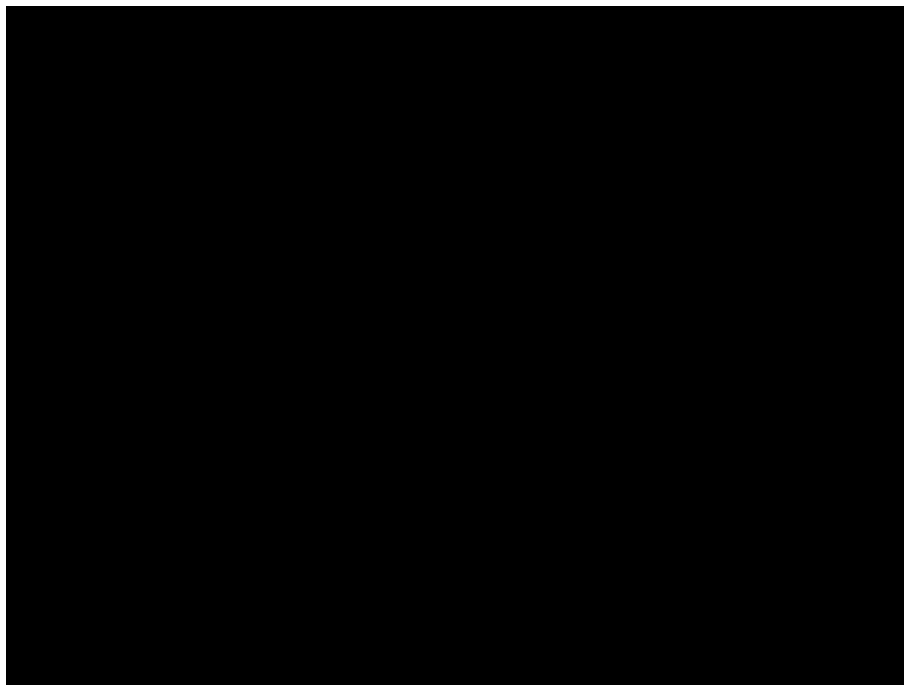
Current Length Detection Framework

0	0	0	4.028
0	4.085	4.07	4.056
0	4.114	4.085	4.07
4.128	4.114	4.114	4.085
4.173	4.158	4.128	4.128
0	4.173	4.158	4.158
0	0	4.188	4.173





Length Detection Framework Demo



Challenges

- (1) Setup Difficulty (solved)
- (2) Lack of Accurate Data
- (3) Underwater calibration

Questions?