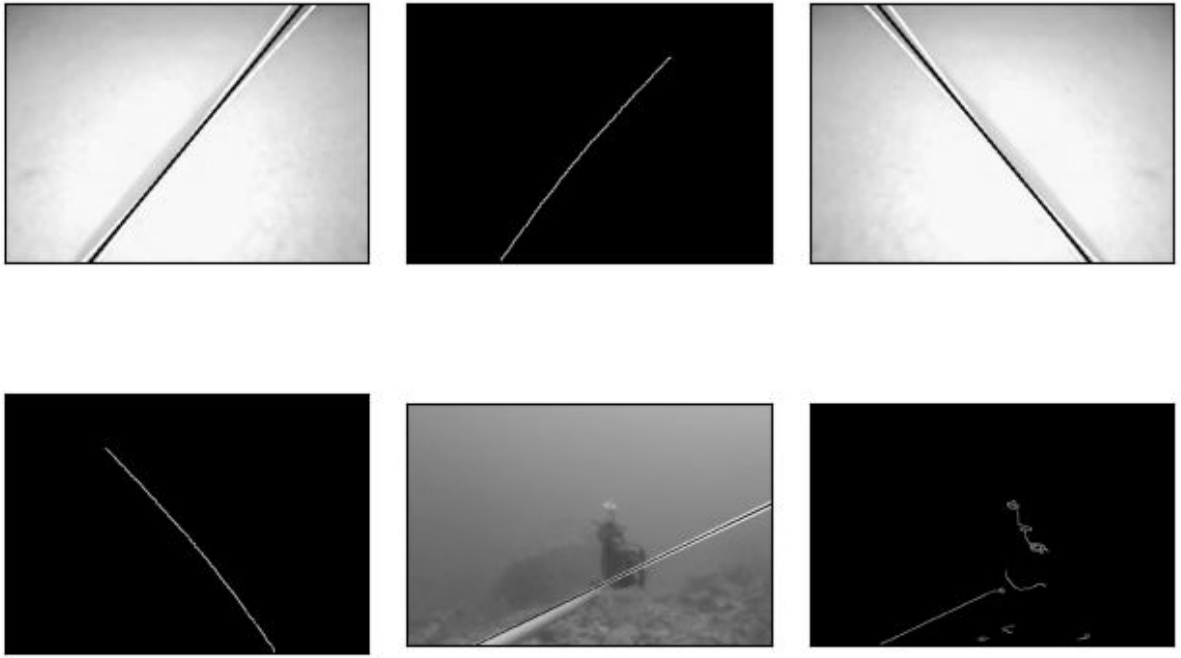
SD2711
Small Craft Design
KTH 2020

Students: Martti Yap, Michaela Söderström, Martin Schulthes, Pernilla Wikström
Supervisor: Aldo Teran Espinoza
Examiner: Jakob Kuttenkeuler

Pipe Tracking

With a Multibeam echosounder

Pre-study of pipe-following with great images (Hough transform, edge detection)

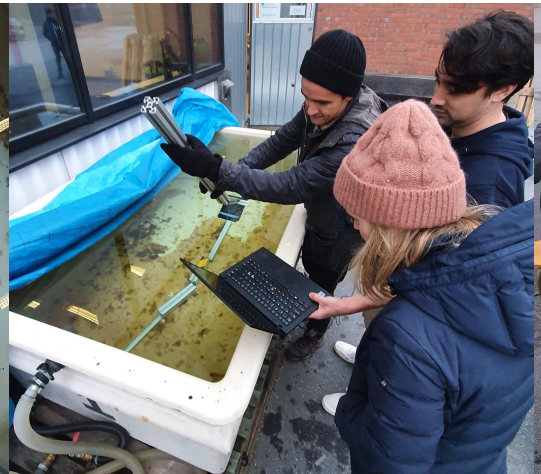


Data collection 30 October 2020

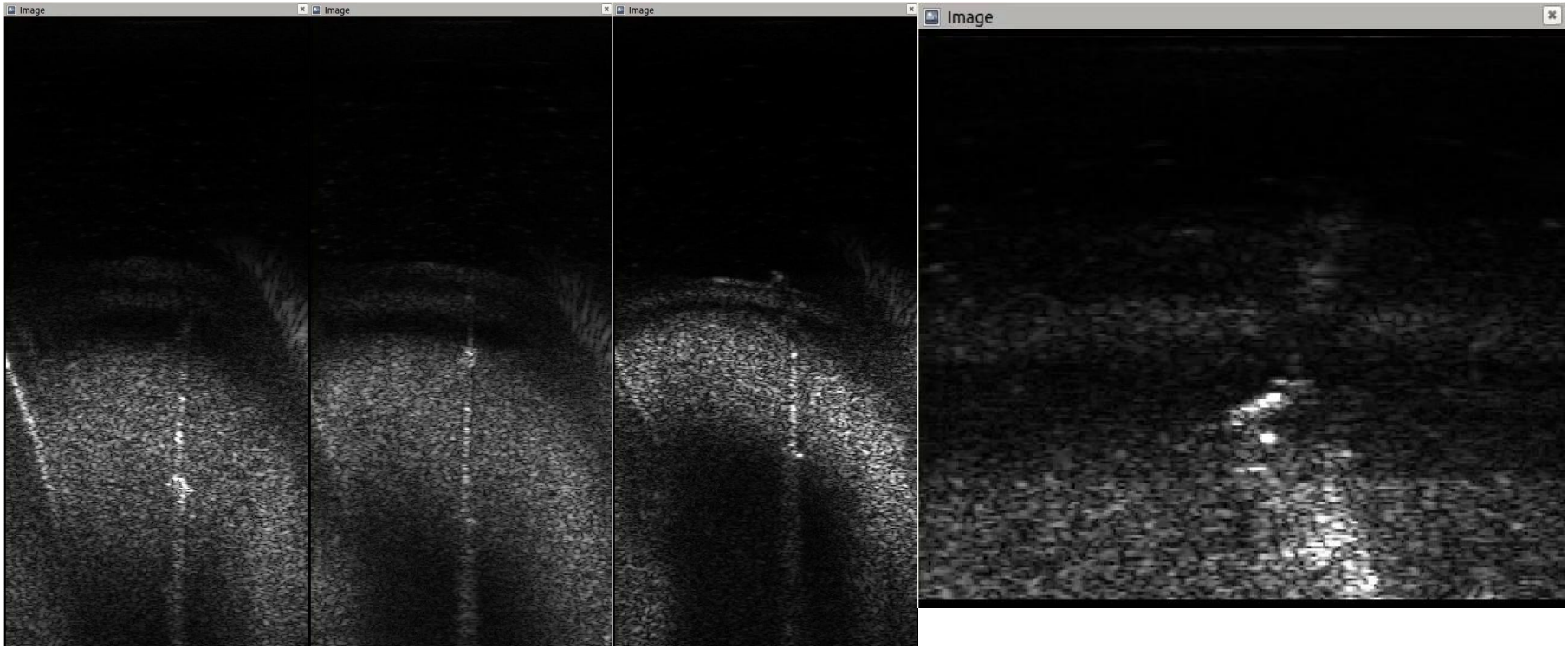
Location: KTH, with metal sticks in a “bathtub”.The data collection was changed by:

- Range of sonar, Gain, Gamma
- Pipe-geometry (straight, curved, T-shaped)

Only high frequency content



Raw data from water tank data collection



These are some of the original image samples from our data collection. As seen, quite noisy and need to be cropped.

Line detection study

- Purpose: investigate different methods to find a good way of finding pipes.
  - Canny Edge detection and Probabilistic Hough Transform
  - Filtered Hough Transform with binary thresholding
  - Binary thresholding and PCA
  - Probabilistic Hough Transform with Histogram Filter

Canny edge detection, probabilistic Hough transform

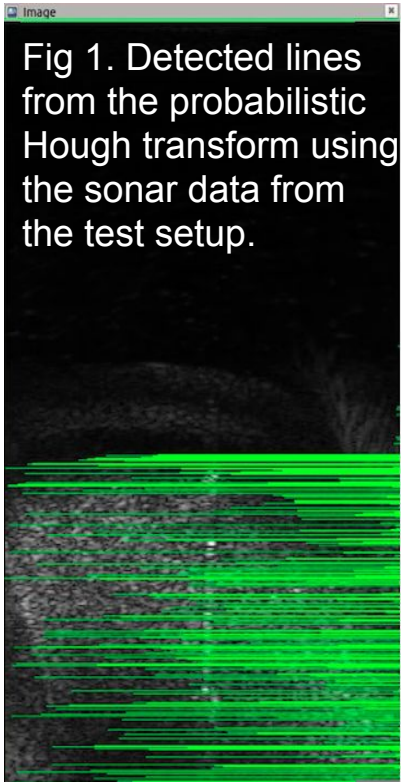


Fig 1. Detected lines from the probabilistic Hough transform using the sonar data from the test setup.

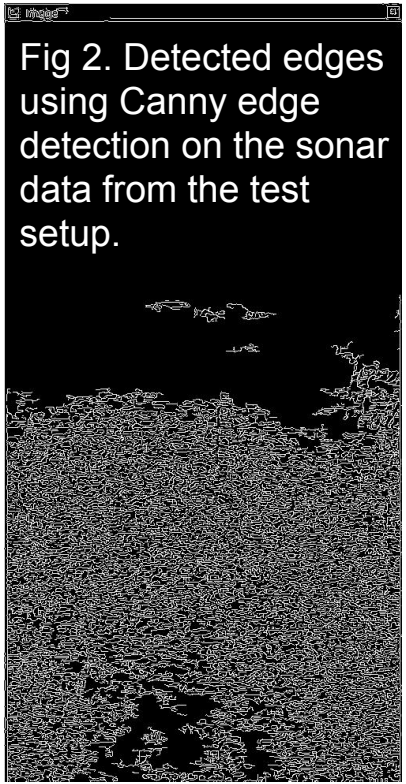
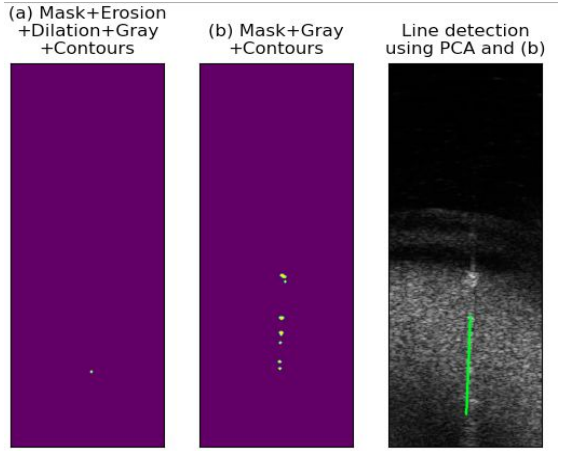


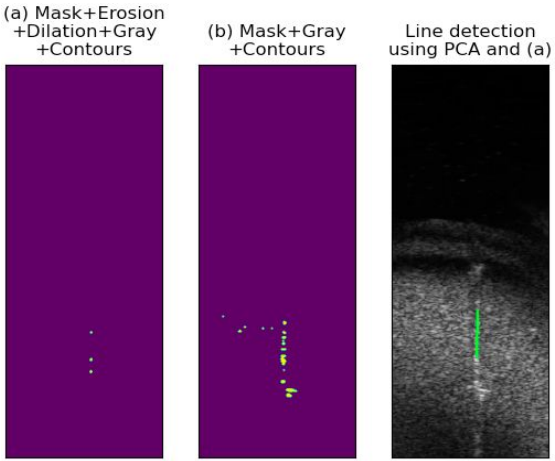
Fig 2. Detected edges using Canny edge detection on the sonar data from the test setup.

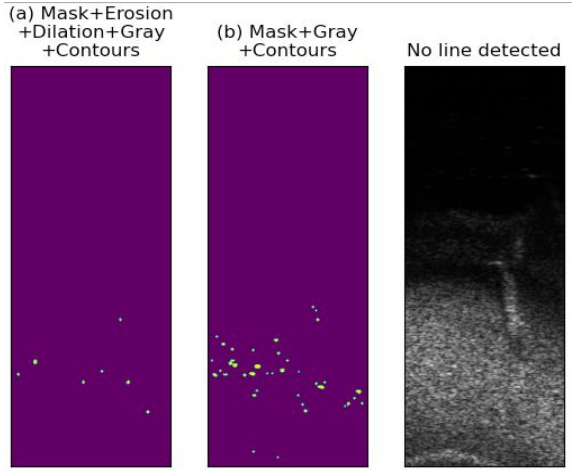
- Used data from the data collection setup
- The data is clearly too noisy and needs to be experimented on preprocessing to manage to achieve better results.

Binary thresholding and PCA



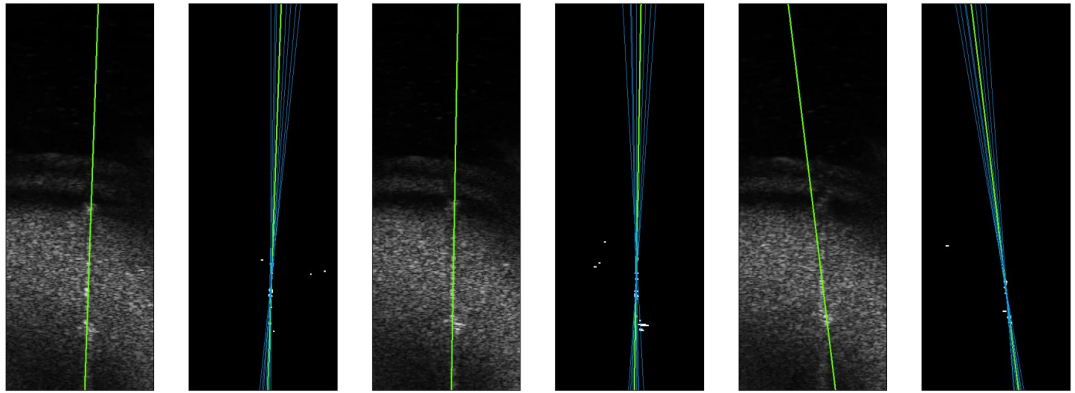
- 2 options of preprocessing:
  - (a) Mask+Erosion+Dilation+PCA
  - (b) Mask+PCA
- Check spread (σ) of white pixels in (a) and (b):
  - Choose the largest ratio of $\sigma(y)$ /$\sigma(x)$ from (a) and (b)
  - If $\sigma(y) > \sigma(x)$ draw line.
  - Filter out the lines not within ± 25° from vertical
  - Filter out lines shorter than 3 pixels long





Filtered Hough transform with binary thresholding

- Binary thresholding top 30% pixels
- Erode & dilate pixels
- Extract 8 best lines from Hough transform (most accumulator votes)
- Filter out the lines not within ± 15° from vertical
- Take the line with most points out of this subset (green)



Probabilistic Hough transform with Histogram filter

- Preprocessing:
  – Gaussian blur
  – Binary thresholding (Best choice for our data, see Fig 1.)
- Histogram filter, see Fig 2

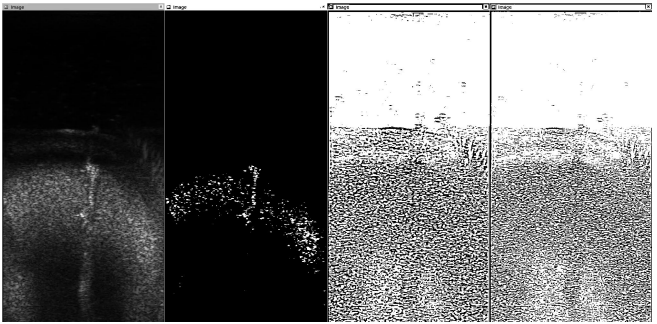


Fig 1. From the left: Original image, Binary threshold, Adaptive threshold of the mean, Adaptive threshold Gaussian

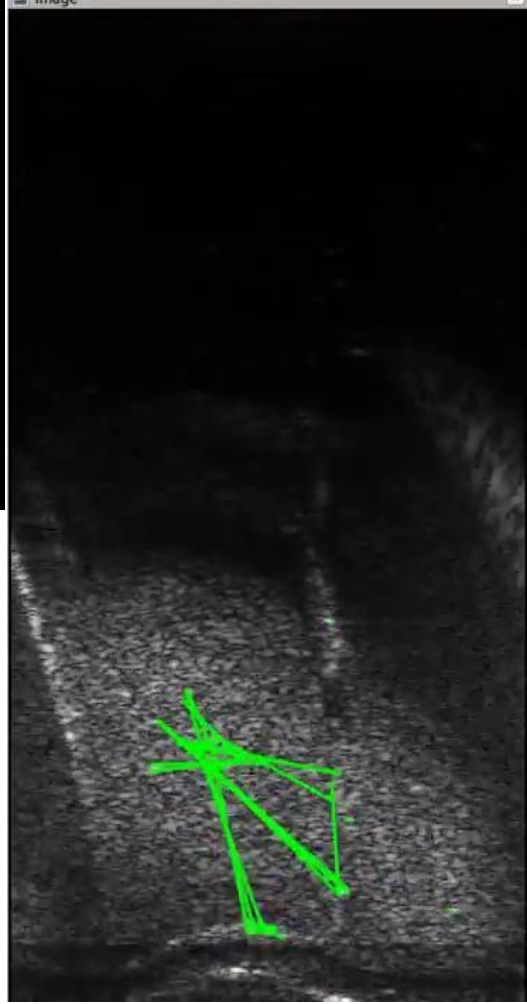


Fig 2. From the left: Preprocessed image, Applied histogram filter

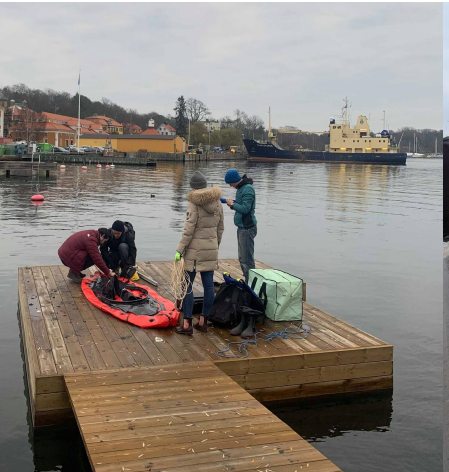
- Spread results depending on data file. The one presented here is easiest to perform on
- Ok performance on this data, but not robust
- ToDo: Fine tune region of interest + other parameters, draw average line

Data collection 2 December 2020

Location: Beckholmen, recording pipes underwater
The data collection was changed by:

- Range of sonar, Gain, Gamma, pipes/targets

Only high frequency content



Line detection investigation - continuing

- Purpose: see if our chosen techniques performs well on real pipes underwater and to improve them
- Chosen methods from the study
  - Filtered Hough Transform with binary thresholding
  - Binary thresholding and PCA
- Added buffer to stabilize line detection, see next slide.

Buffer

- Purpose: make the line detection more stable and avoid rapid changes of rotations
- Steps:
  - Adding all results of line detection of N frames to the buffer
  - Updating the buffer to only include valid lines and remove the rest.
  - Computes and return average line out of N frames.
    - Uniformly weights on each line
    - Linear weights on lines, such that the most recent frame has the largest weight

ROS implementation

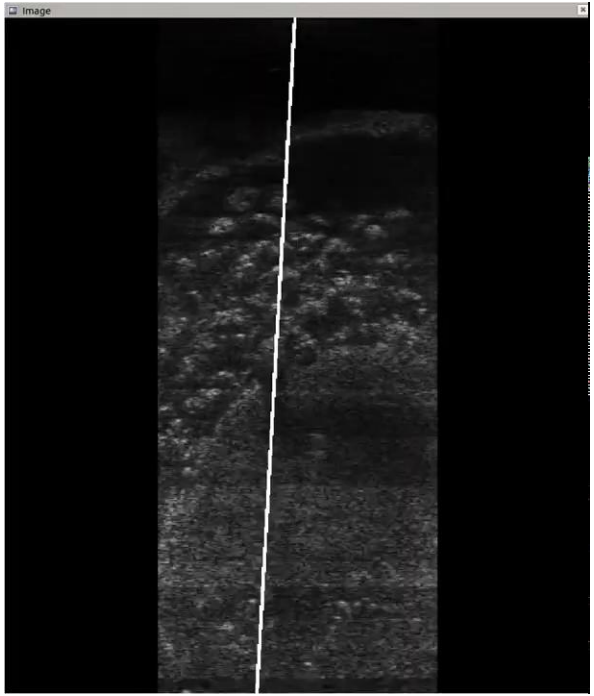
The main file

- read bags
- process the bag with either hough transform (line_detect1.py) or PCA (line_detect2.py)
- publish results to node in terms of images and points (p1,p2) = ((p1x, p1y),(p2x, p2y))

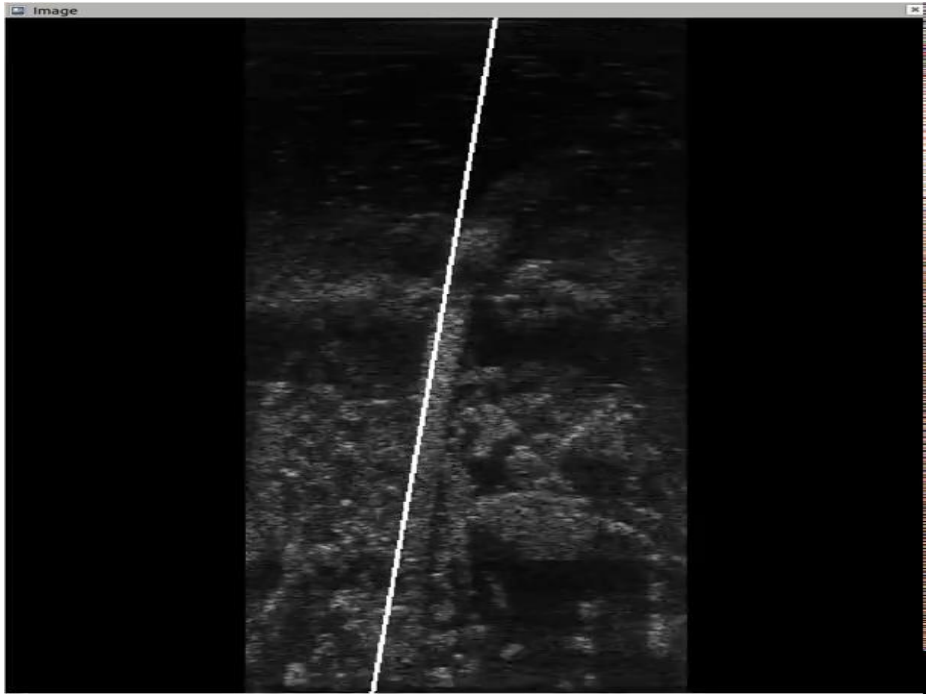
Results

- The results shows a successful project where it is possible to find pipes underwater using two different computer vision methods.

PCA



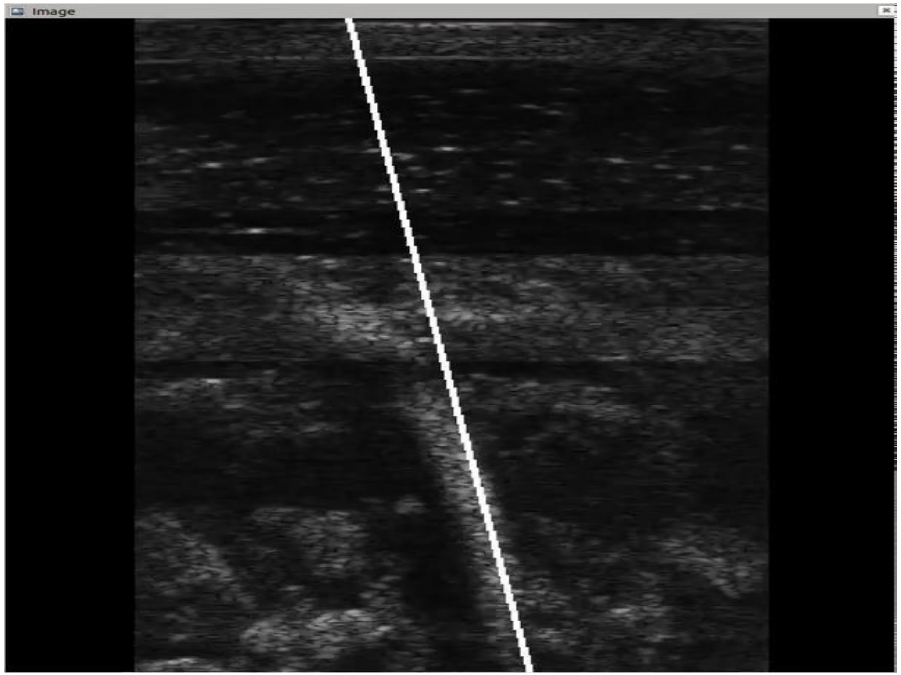
Hough



Results

- The results shows a successful project where it is possible to find pipes underwater using two different computer vision methods.

Hough



PCA