Water level detections from ship hull

By Group 20

Problem Statement

Given a video (provided by the TA's) that shows the draft marks of a ship or large vessel, the objective is to accurately determine the water level in meters for every frame of the video. This involves reading the numerical markings on the ship's hull where the waterline intersects the draft marks.

Draft Mark

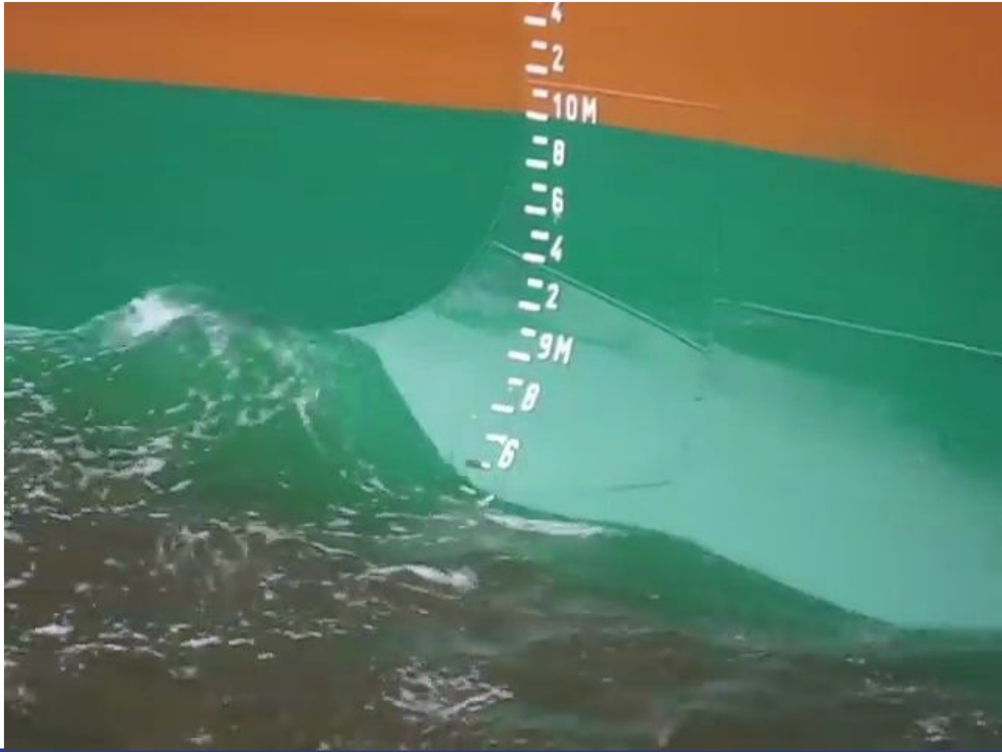
- Draft marks, or draught marks (British spelling), represent the distance between the ship's keel (the lowest longitudinal support member) and the waterline where the hull meets the water.

- The draft indicates how low the ship sits in the water. Overloading a vessel increases its draft, potentially making it unstable and prone to flooding, especially in rough seas.

- Draft marks are typically painted at multiple locations on a ship: near the bow (fore), midship, and stern (aft), on both port (left) and starboard (right) sides.

Video Frame of the Draft marks provided

Draft reading of current frame = 8.5m



Pipeline Overview

Video Initialization & Frame Acquisition: The script initializes video capture from a file and reads the frames sequentially.

Initialization: Set up the video capture from a file ("hull.mp4") and initialize a video writer to save processed frames.

Frame Acquisition: Frames are read one by one from the video.

Image Preprocessing: Each frame undergoes perspective transformation to align draft markings on the hull.

Perspective Correction: Applies a homography matrix to each frame to correct the perspective, making the draft markings easier to analyze.

Continued

Water Level Detection: The water level is detected using image thresholding, contour detection, and height estimation based on pixel measurements.

Threshold and Contour: Converts images to grayscale and binary, then detects contours that represent potential water levels.

Height Estimation: Calculates the height in pixels where the water level intersects with the hull.

Optical Character Recognition (OCR): Extracts textual data from the processed frames using Tesseract OCR to identify numeric markings indicating water levels.

Continued

OCR Processing: Applies image filters to enhance text visibility and then uses Tesseract OCR to extract characters and numbers that denote water levels.

Water Level Calculation: Calculates the water level in meters based on the pixel height and markings recognized in the OCR step.

Marking Analysis and Calculation: Identifies markings on the hull to translate pixel measurements into meters, adjusting for camera perspective and hull curvature.

Video Output and Display: Processed frames are displayed and saved to a new video file as output and deliverables.

Continued

Display and Save: Modifies the frames to include water level annotations and saves them to an output video file for review.

Data Analysis and Plotting: Analyzes the collected water level data to remove outliers and plots the average water level over time.

Statistical Analysis: Filters out outlier measurements and calculates the average water level.

Plotting: Visualizes the average water level against the frame number to show changes over time.

Purpose of Perspective Transformation

To correct perspective distortions in video frames to accurately analyze water levels on a ship’s hull.

Ensures that draft markings, which are essential for measuring water levels, are uniformly presented across all frames.

After applying the transformation, you typically would visually inspect the result to ensure that the draft markings are now properly aligned. This can also involve overlaying lines or grids on the image to check alignment.

Water Level detection

Grayscale Conversion: Simplifies the image for processing.

Binary Threshold: Highlights the waterline.

Background Subtraction: Differentiates moving water from the static background.

Morphological Operations: Cleans the mask by removing noise.

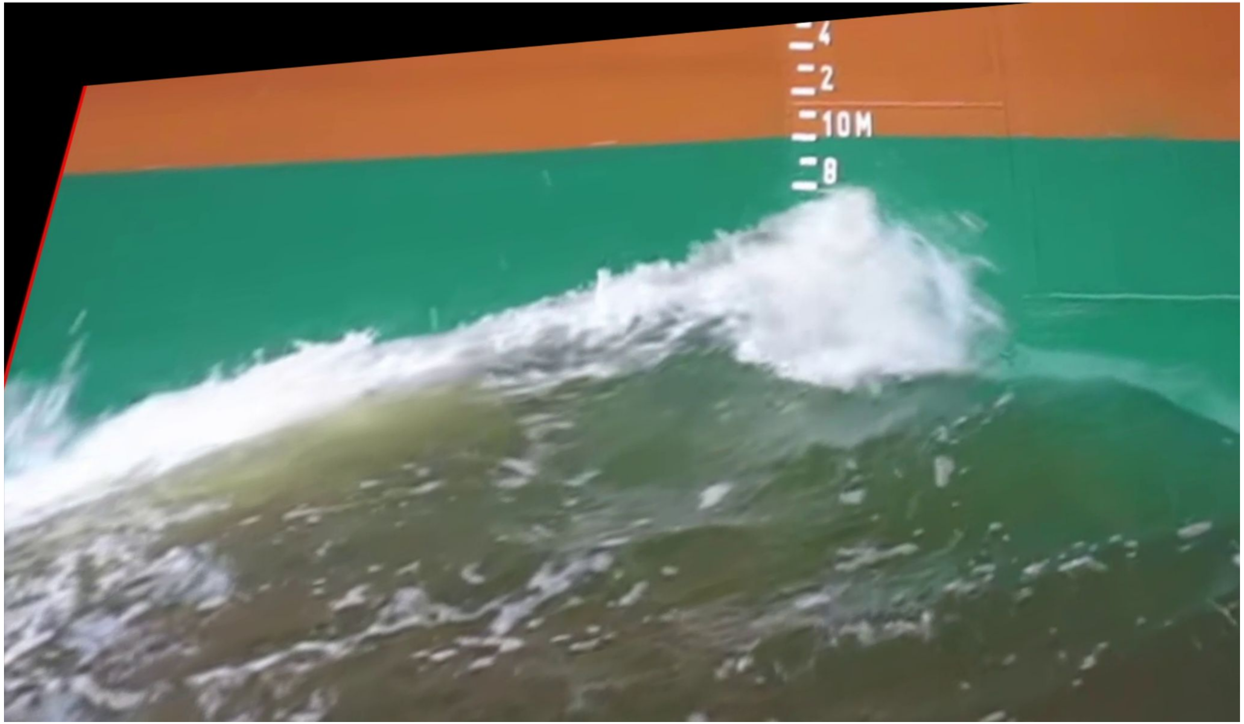
Contour Detection: Finds the outline of the waterline.

Largest Contour: Identifies the largest contour as the waterline and extracts its position.

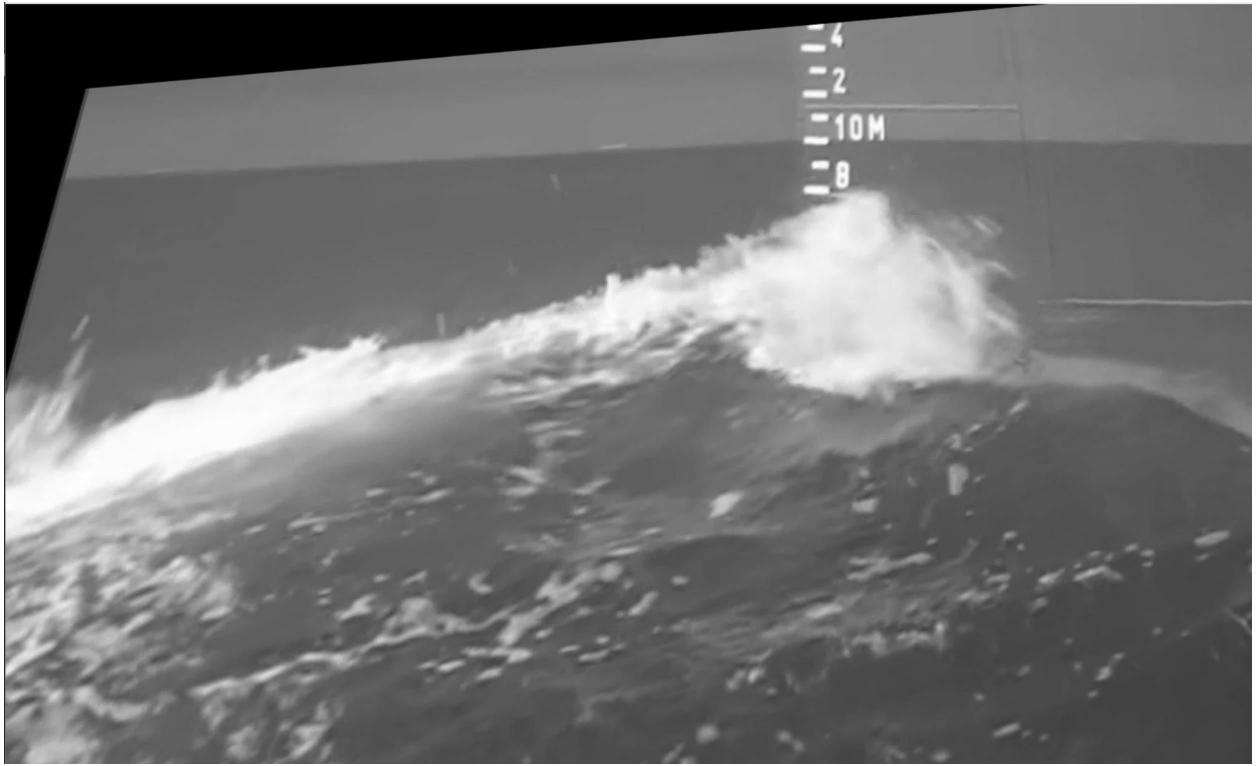
Return Values: Provides the Y-coordinate of the waterline and the processed frame.

Sample Frame

Warping Perspective



Convert the image to Gray



Threshold the frame to get water and draft levels

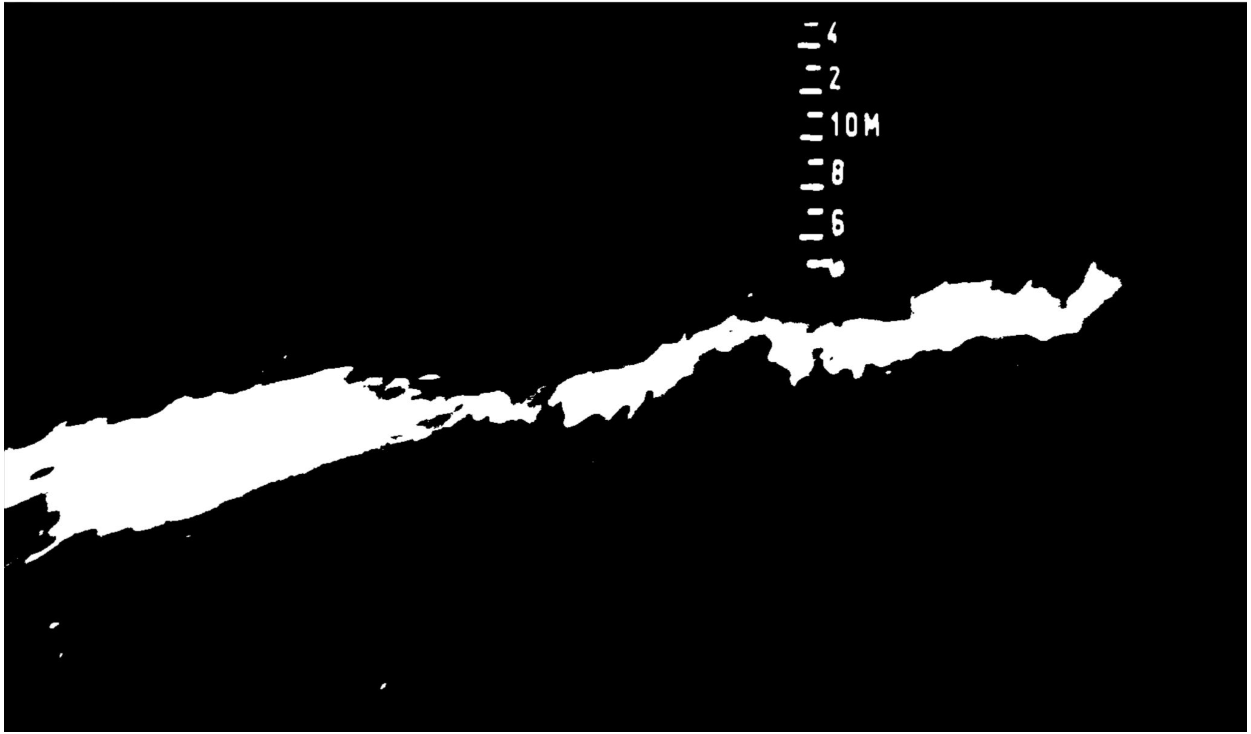
Subtract the background from the foreground

Perform Morphological Operations to clean frame

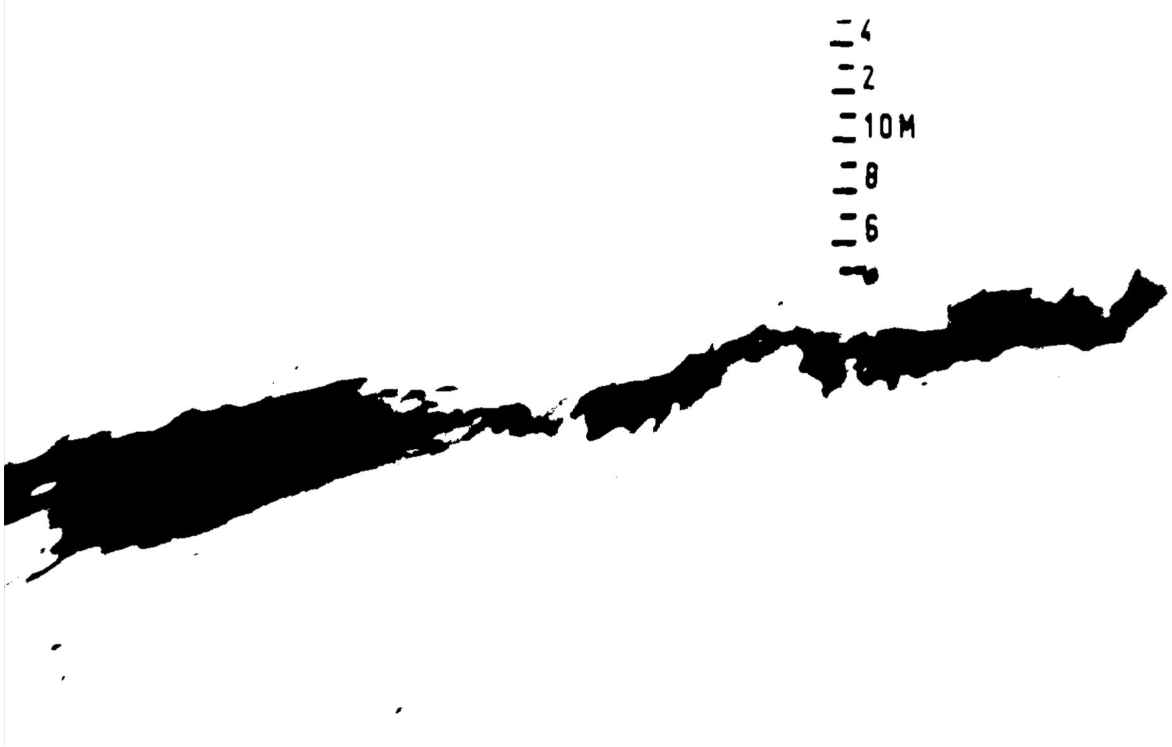
Extract the largest contour

Here we extract the largest contour which would correspond to the water, and then we draw a bounding rectangle to get the position of the contour and its height and width.

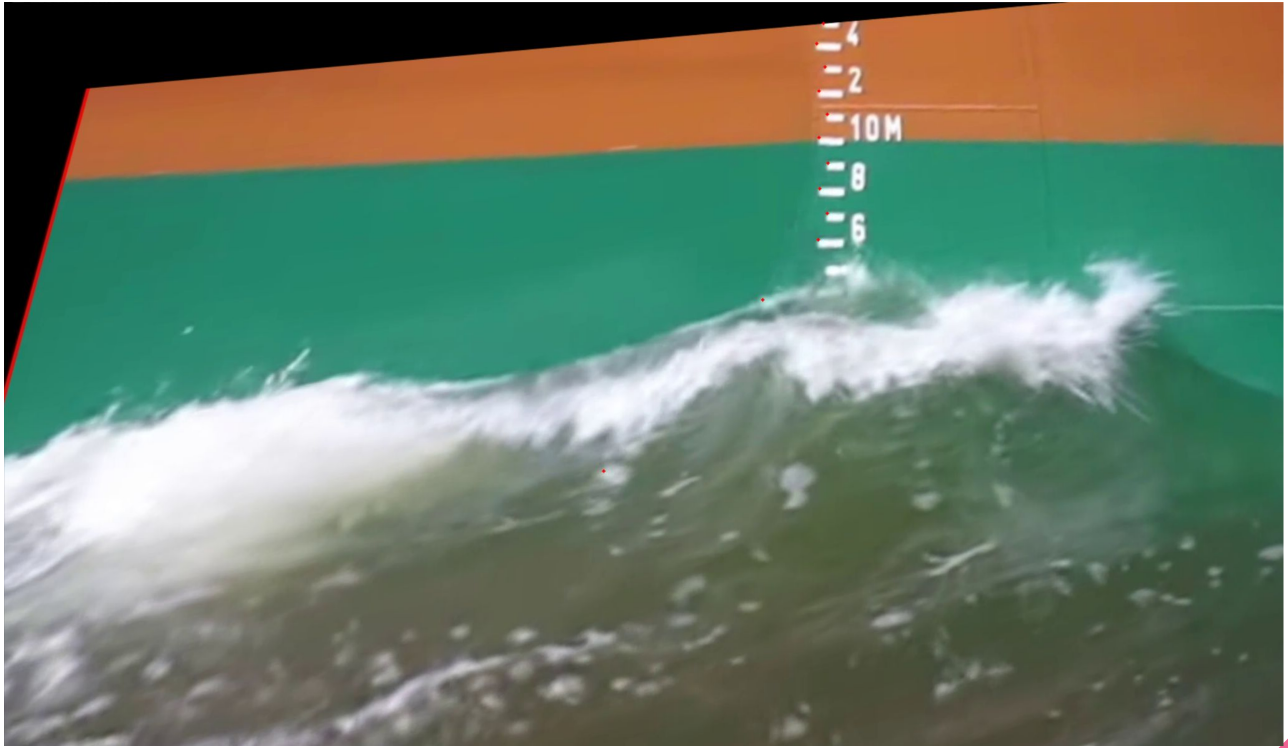
Sharpening the image for OCR



Inverting the image for OCR reading



Markings of the draft numbers



Logic for measurement of waterline

- Let's say we have the following markings on the ship's hull: 10.5, 10.4, 10.3, 10.2, 10.1, 10.0
- Find the pixel gap between each marking and map the values. How?
- Seems roughly the difference between consecutive markings was around 50 pixels.
- Previously we know the waterline y-coordinate.Now we look for the mapped pixel value that is closest to the waterline y-coordinate. Keep in mind that the mapped pixel value should just greater than the y-coordinate.
- Calculate the remaining distance, use the pixel gap to convert that distance to meters and add it to the mapped distance.

Conclusion

For our report we plan to do the following:

- Improved Water Level Detection Accuracy: By integrating optical flow techniques, specifically the Farneback method, we will enhance our system's ability to accurately track waterline movements, even in varying conditions.
- Consistent OCR Alignment: The use of homography transformation ensures that the draft marks are consistently aligned for OCR, resulting in more reliable readings from the detected characters on the ship hull to get a more accurate output.
- Robust Performance in Variable Conditions: The combination of optical flow and OCR, along with post-processing to eliminate outliers, ensures accurate water level detection despite potential issues like motion blur or low-quality video, reducing errors to approximately 0.1 meters.

Future Work

Several promising directions for future work can enhance the system's capabilities, accuracy, and applicability.

- Integration of Machine Learning Models: Implement a CNN-based model trained on diverse datasets encompassing various lighting conditions, water surfaces, and ship hull structures to ensure robust performance across different scenarios.
- Real-time Processing Enhancements : Utilize optimized image processing techniques, hardware acceleration (e.g., GPUs), and parallel computing paradigms to minimize processing latency and enable seamless integration with onboard monitoring systems.
- Hardware Integration for Onboard Monitoring: Develop compact and ruggedized sensor arrays equipped with onboard processing units capable of real-time data collection, analysis, and transmission. Integrate these systems seamlessly with existing onboard infrastructure for comprehensive monitoring and control.

Thank you!