COMPUTER VISION 24/25

Detection in underwater videos

Artur Almeida | Cecília Santos

Videos

- Underwater
- Camera motion
- Turbidity and sediments
- Low contrast
- Light scattering

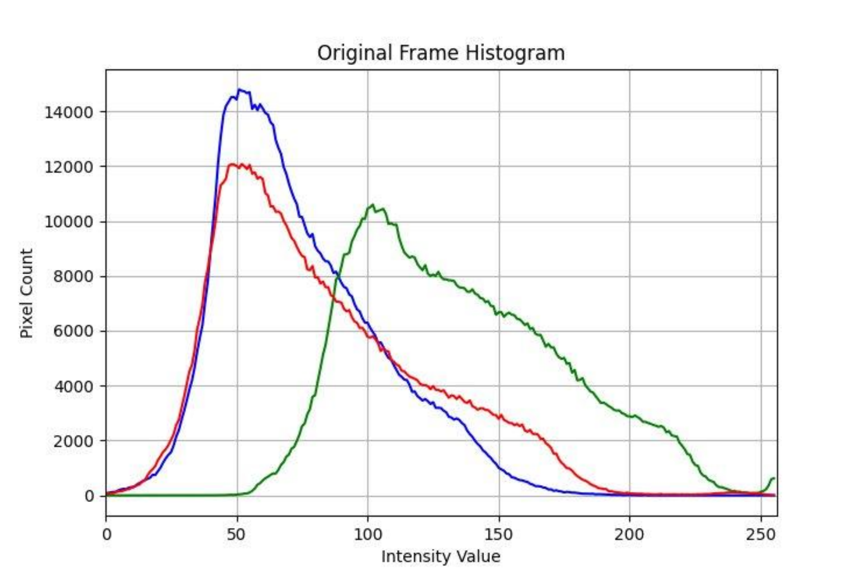
Overview

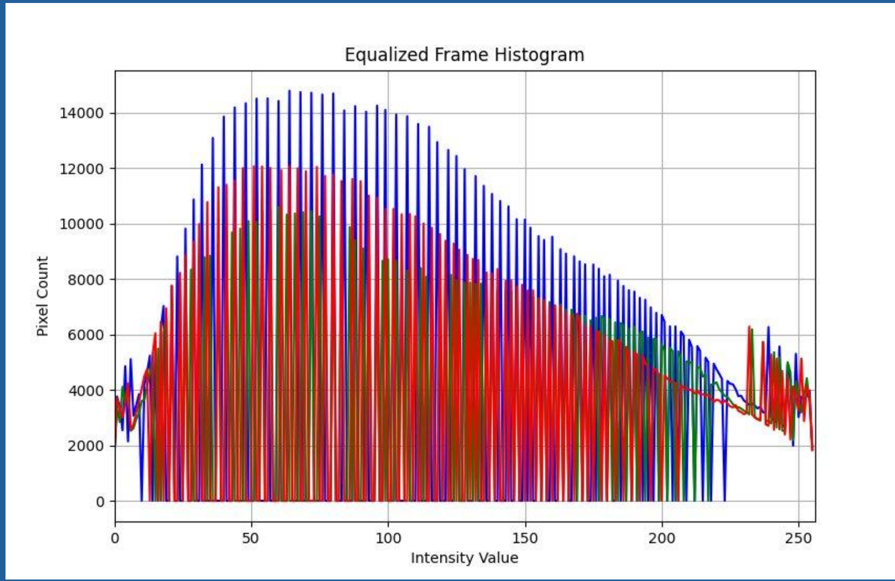
- **Detecting greenness**
  - Preprocessing- equalization
  - Mask- HSV thresholds
  - Morphological operations- refine mask
  - Contour detection
  - Results- green algae detection
- **Detecting motion**
  - Preprocessing- median blur
  - Motion compensation- align frames
  - Mask- difference between frames
  - Centroids
  - Tracking-
  - Results- fishes detection and tracking

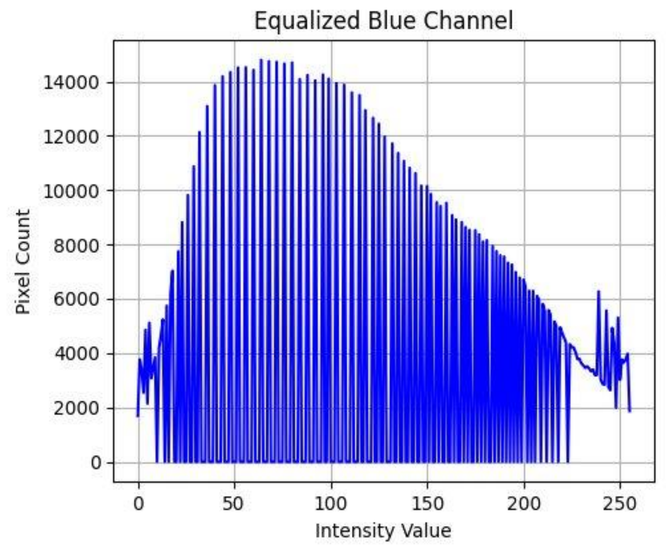
1. Detecting algae

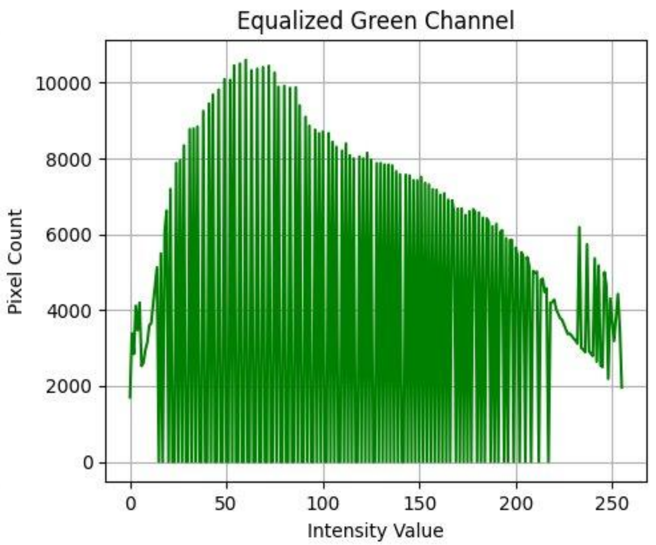
1.1. Preprocessing videos

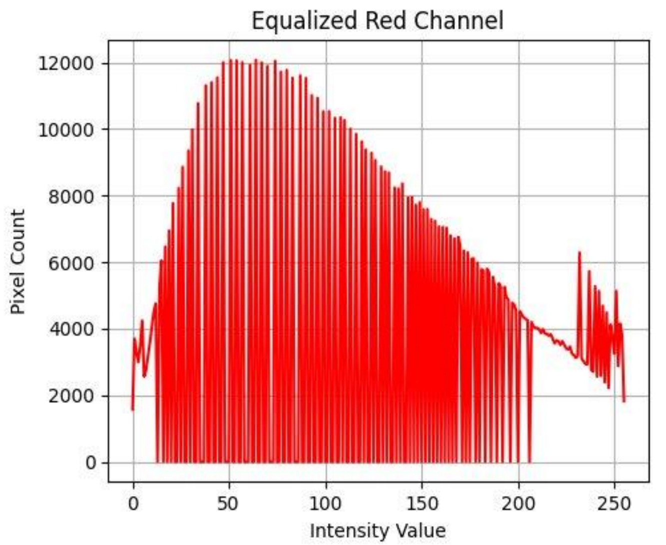
Color Channel Equalization: improve contrast (*cv2.equalizeHist*)





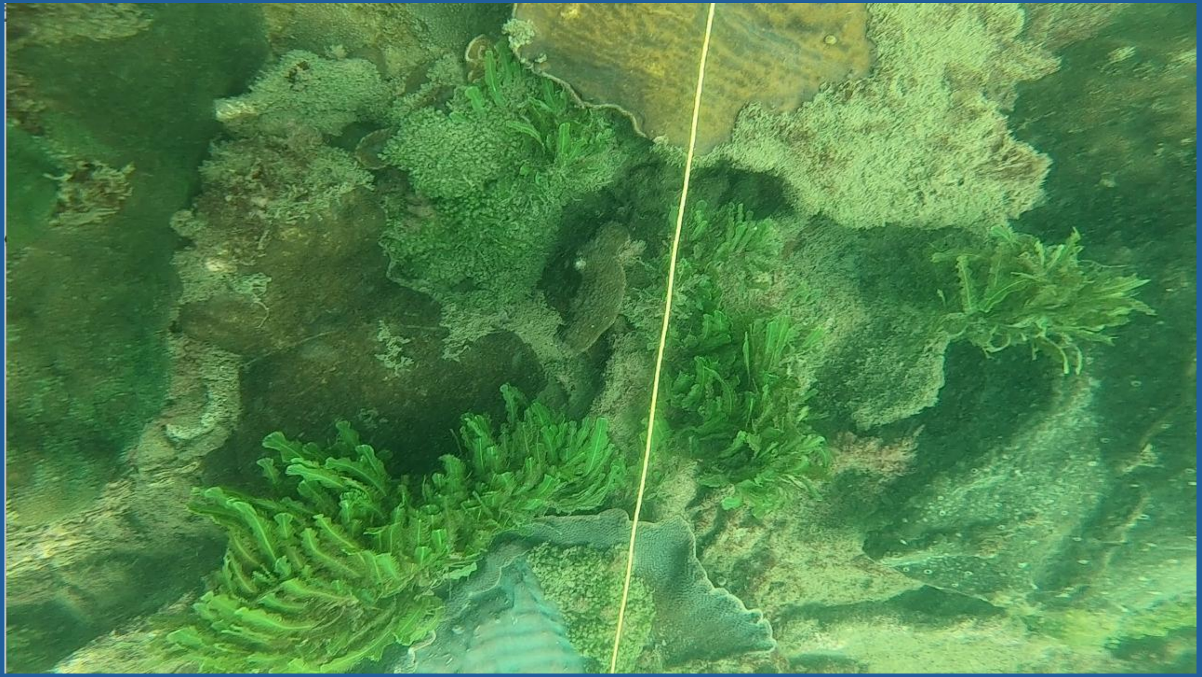






Results of equalization:

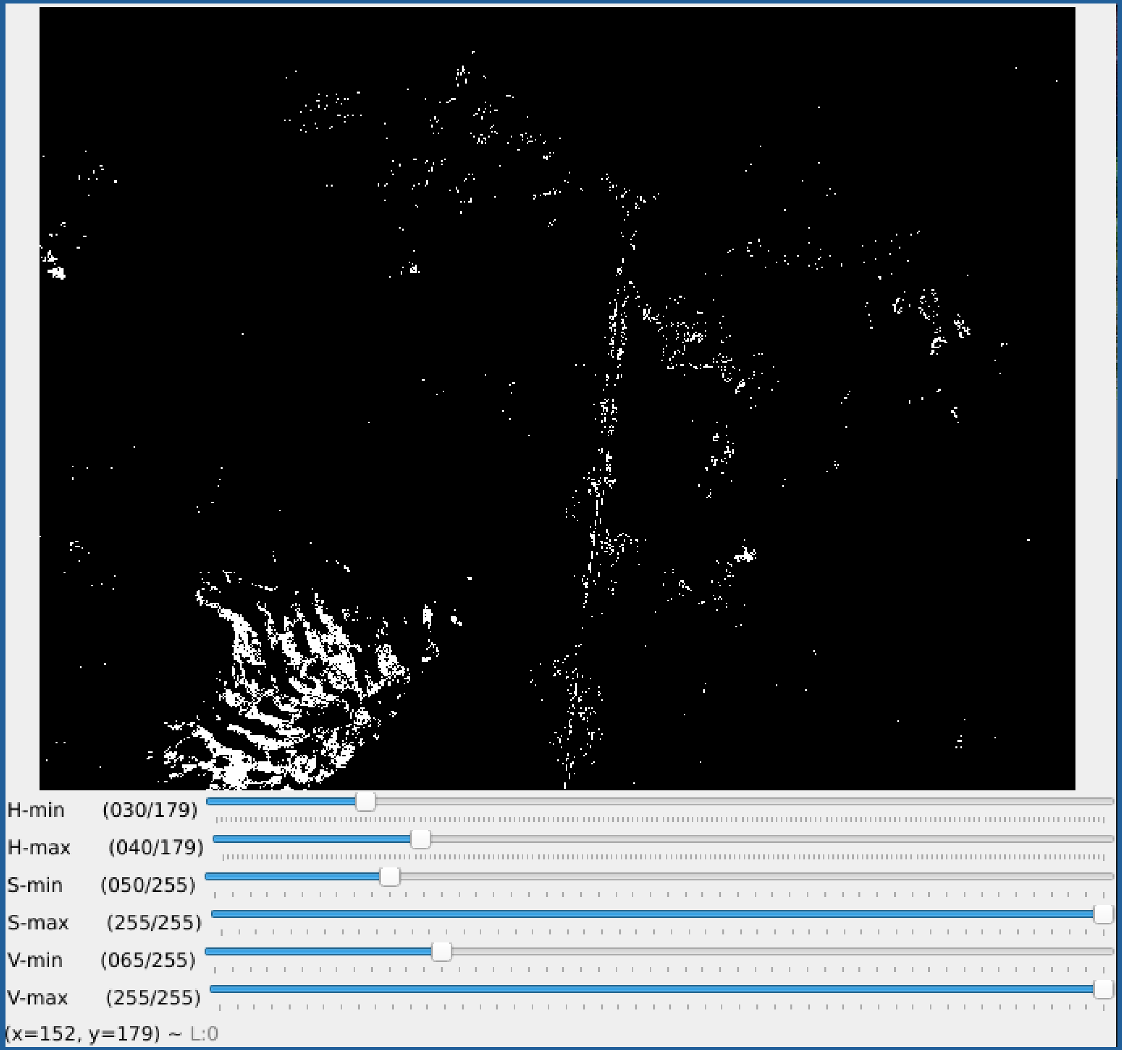
Before



After

HSV Conversion: Converts frames from BGR to HSV: more intuitive color isolation (*cv2.cvtColor*). Define HSV thresholds to detect specific colors with trackbars.

Comandos Possíveis	
n N	Avançar para o proximo frame
l L	Registar os intervalos atuais de HSV para gravar
w W	Gravar no ficheiro limits.json todos os intervalos HSV guardados
q Q	Fechar as janelas e fechar o programa



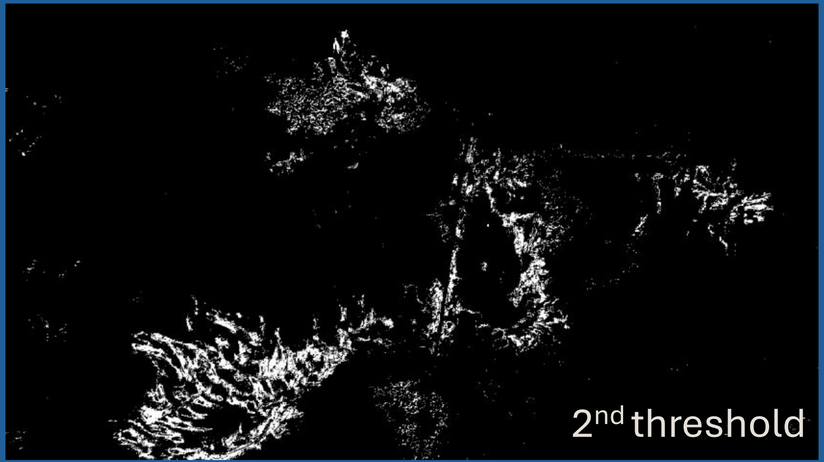
1. Detecting algae

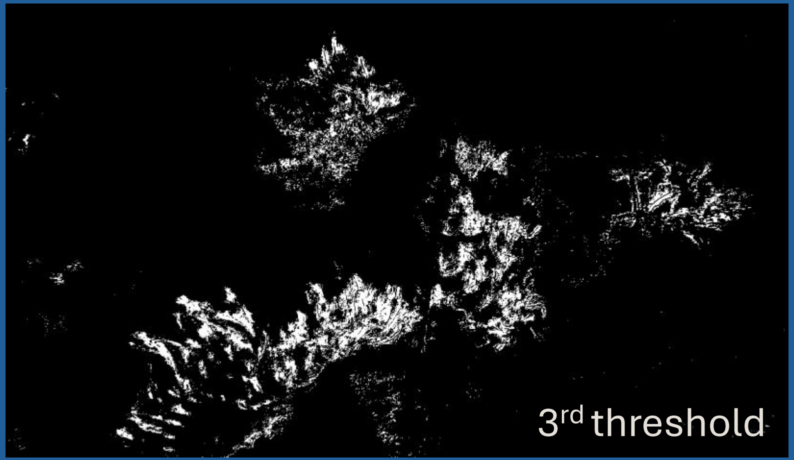
   1.2. Color Mask Creation

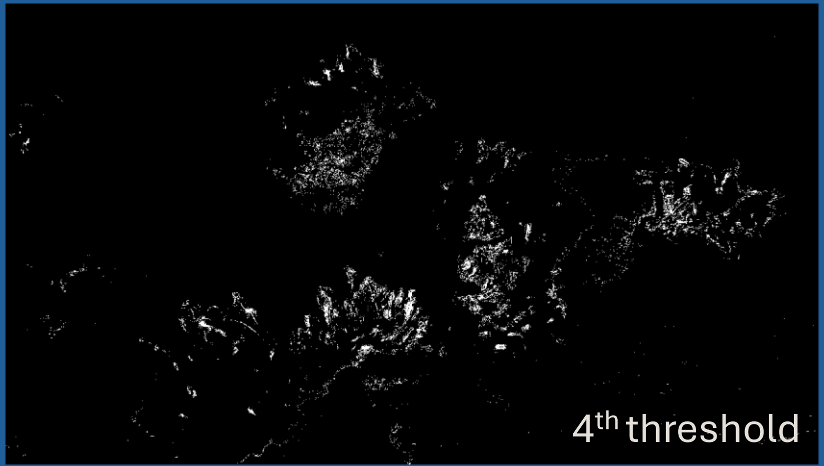
- Define a series of minimum and maximum HSV thresholds to detect greens
- For each HSV threshold generate a **mask** (*cv2.inRange)*
- Masks are **combined** to aggregate results from all thresholds (*cv2.bitwise_or*)

----- Limites a usar -----

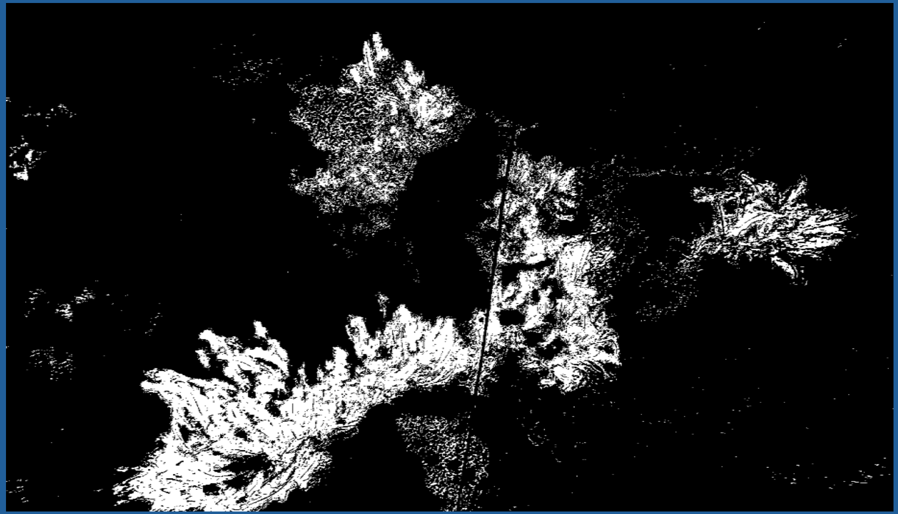
H	mín: 30	máx: 40
S	mín: 50	máx: 255
V	mín: 65	máx: 255
H	mín: 40	máx: 50
S	mín: 50	máx: 255
V	mín: 65	máx: 255
H	mín: 50	máx: 65
S	mín: 50	máx: 255
V	mín: 65	máx: 255
H	mín: 65	máx: 75
S	mín: 50	máx: 255
V	mín: 65	máx: 255







Combined



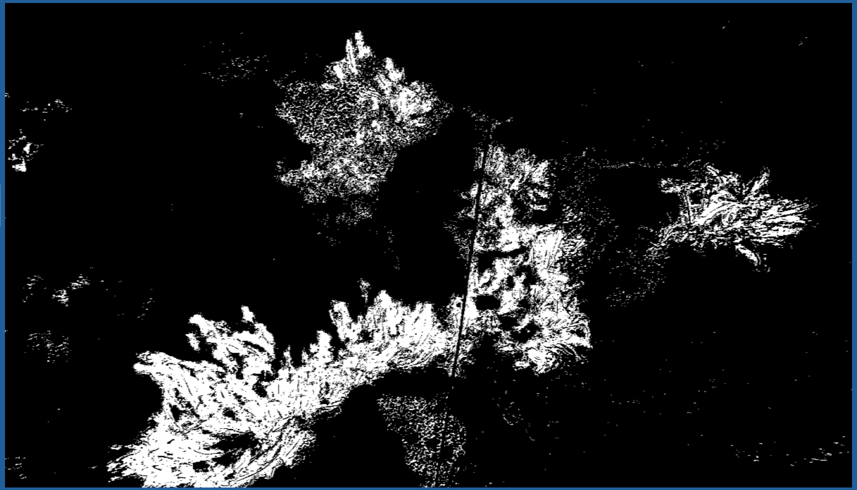
1. Detecting algae

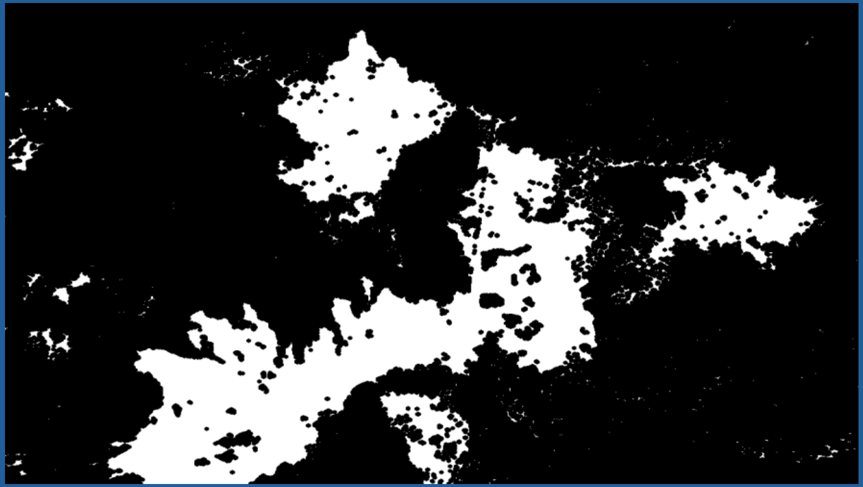
1.3. Morphological operations

To refine the combined mask:

- Close: Fills small gaps within detected regions.
- Open: Removes small noisy detections.
- Close: Merges nearby regions.

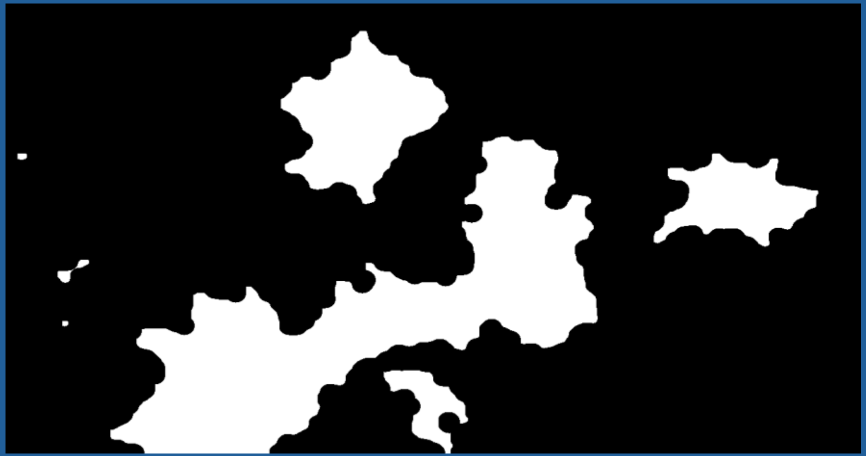
Combined Mask





1st Close

Open



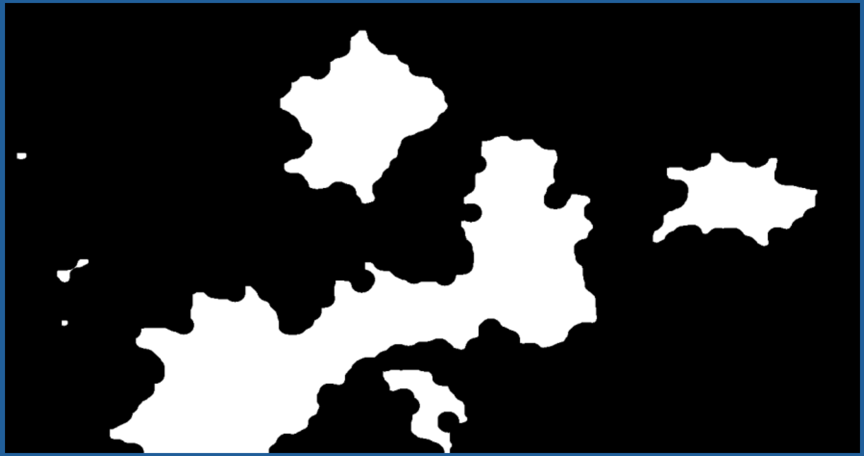
2nd Close

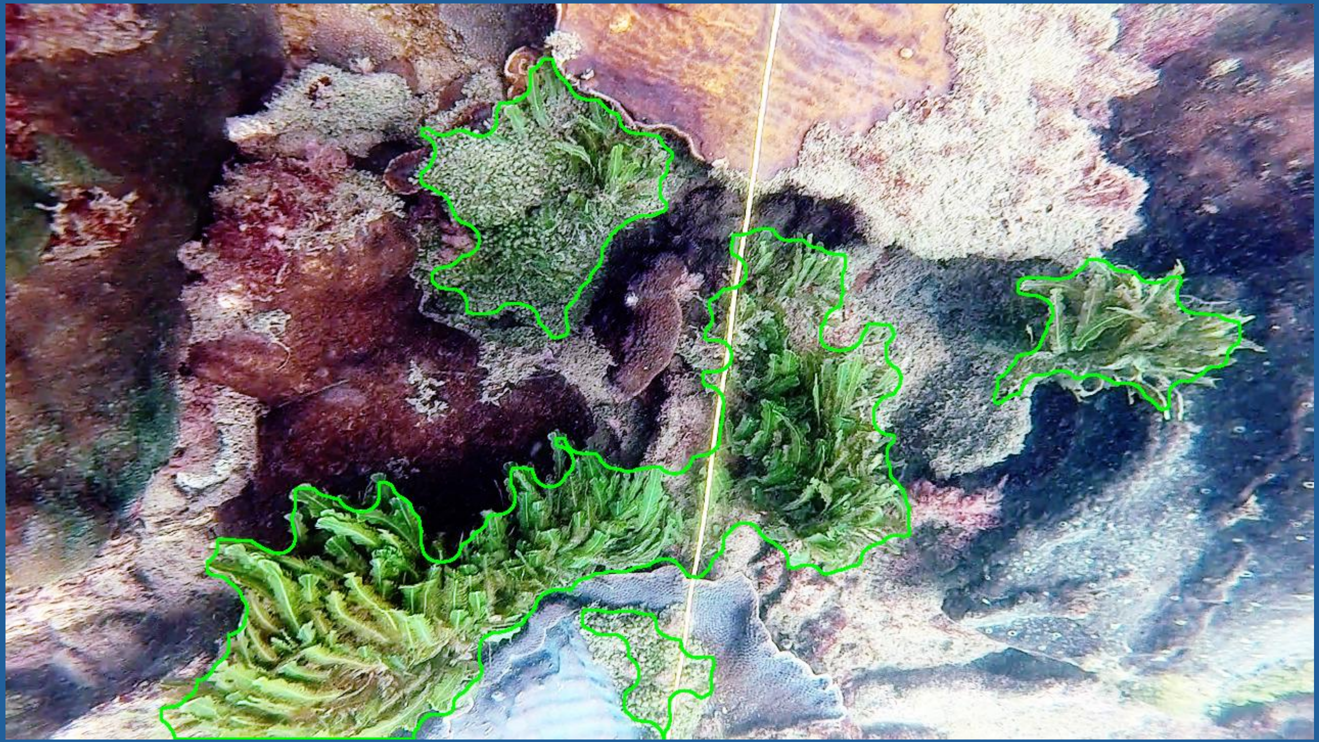
1. Detecting algae

1.4. Contour Detection

Contours are detected on the filtered mask (*cv2.findContours*)

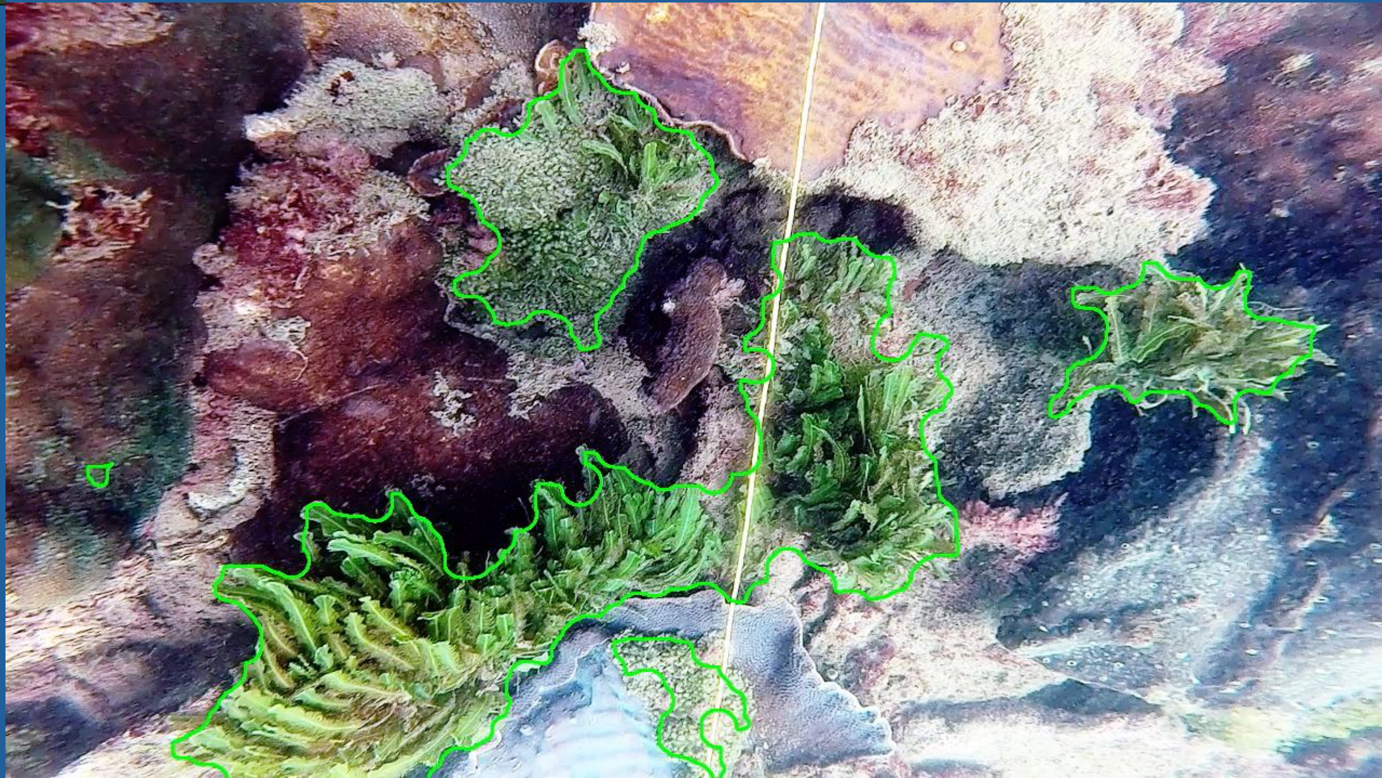
Only contours with a significant area (≥**500** pixels) are drawn on the original frame.





1. Detecting algae

1.5. Results



2. Detecting fishes

2.1. Video Preprocessing

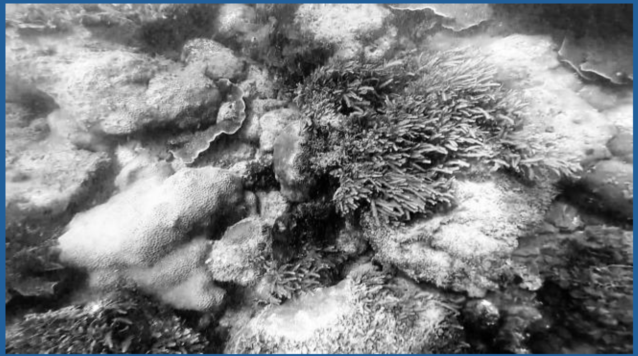
To improve motion and contour analysis:

- Equalize histogram for lighting normalization (*cv2.equalizeHist*) and convert to grayscale (*cv2.cvtColor*).
- Apply median blur to reduce noise (*cv2.medianBlur*).

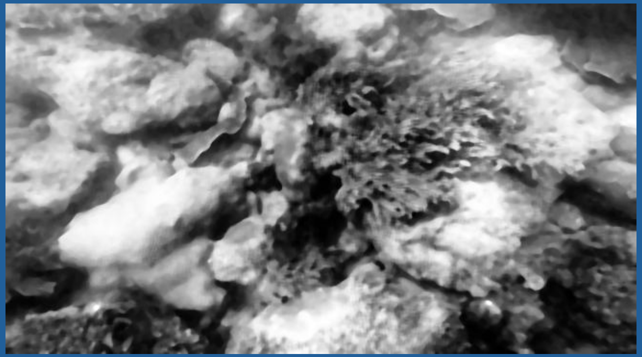
Original

Equalized

Grayscale



Blurred

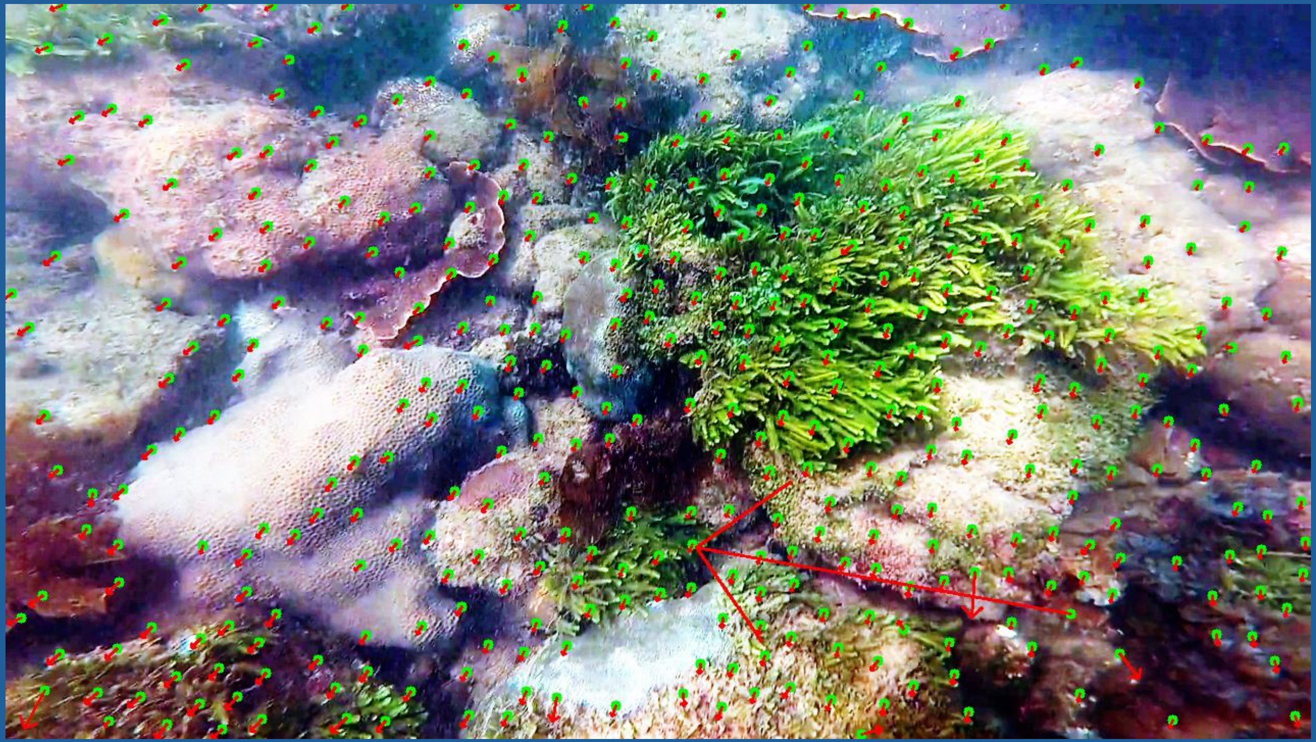


2. Detecting fishes

2.2. Motion Compensation

To reduce false positives in motion detection caused by camera movement and stabilize the video by aligning consecutive frames using feature matching and affine transformation.

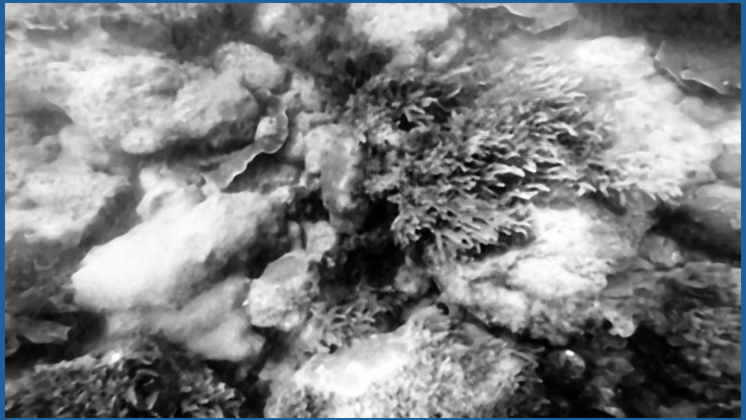
- **Feature Detection**: detects points of interest in the `previous frame` using the Shi-Tomasi corner detection method (*cv2.goodFeaturesToTrack*).
- **Optical Flow**: Lucas-Kanade method calculates the motion of each feature point from the previous frame to its corresponding location in the current frame (*cv2.calcOpticalFlowPyrLK*).

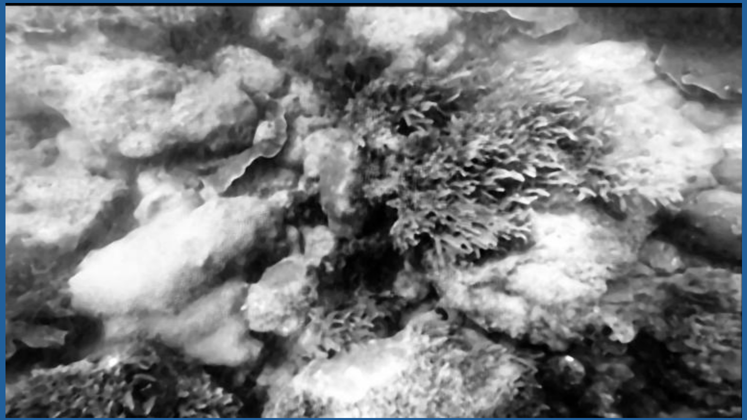


Points of interest in green and motion vectors in red.

- **Affine Transformation Estimation**: With matched points, an affine transformation matrix is computed (*cv2.estimateAffinePartial2D*). Then applied to align the `current frame` with the `previous frame` (*cv2.warpAffine*).

Original frame





Aligned frame

Original frame

Difference

2. Detecting fishes

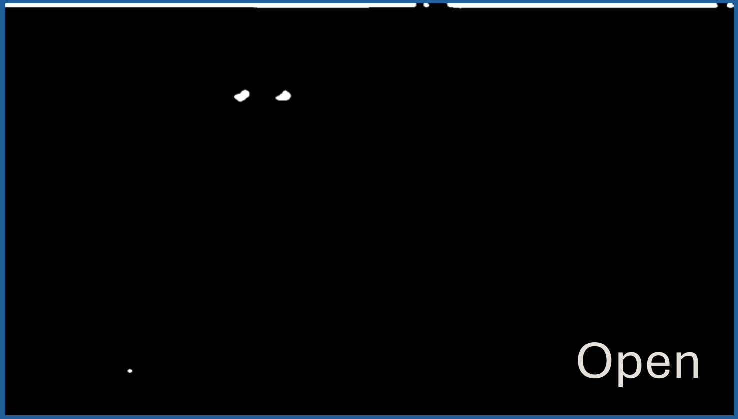
2.3. Filtering and Noise Removal

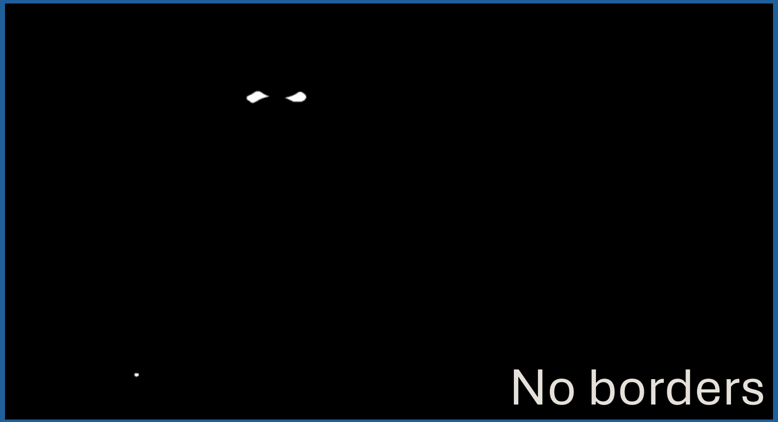
To highlight areas that do not align due to motion, we computed the absolute difference between the motion-compensated frame and the current frame (cv2.absdiff). Then convert the difference image into a binary mask, where pixels with a value greater than 70 are set to 255 (cv2.threshold), filtering smaller variations due to noise and illumination, and isolating motion.

To remove noise using morphological operations:

- Open: Removes small noisy detections.
- Close: Merges nearby regions.

Borders excluded to prevent false detections from edges.



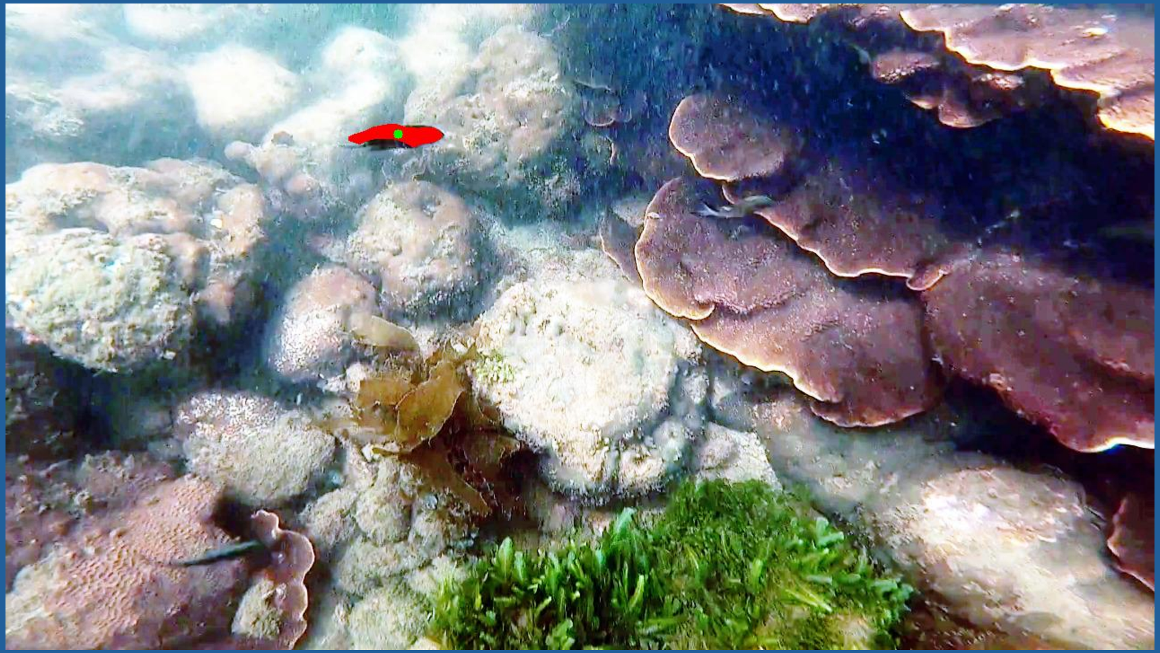


2. Detecting fishes

2.4. Detection and Tracking

Detect and track moving objects across video frames by:

- Identifying contours (*cv2.findContours*) in the binary filtered frame (motion differences). Discard small contours.
- Computing centroids for each detected contour (*cv2.moments*).
- Matching new contours with previously tracked objects based on proximity, with the Euclidean distance.
- Updating (matched) or initializing (not matched) objects. If they have been visible for more than 3 frames than the contour is drawn.



Mask in red and centroid in green.

2. Detecting fishes
2.5. Results

Processed

(x=989, y=719) ~ R:5 G:3 B:9

2. Detecting fishes
2.6. Approach with tracking

Previous approach:

- Detects fishes based on motion detection
- No explicit tracking, analyses contours based on their spatial proximity

Process:

1. Filters contours below a minimum area.
2. Calculates **centroids** of contours.
3. Matches new contours to existing IDs using distance (based on proximity to previous centroids).
4. Creates new IDs for unmatched contours.

Tracking approach:

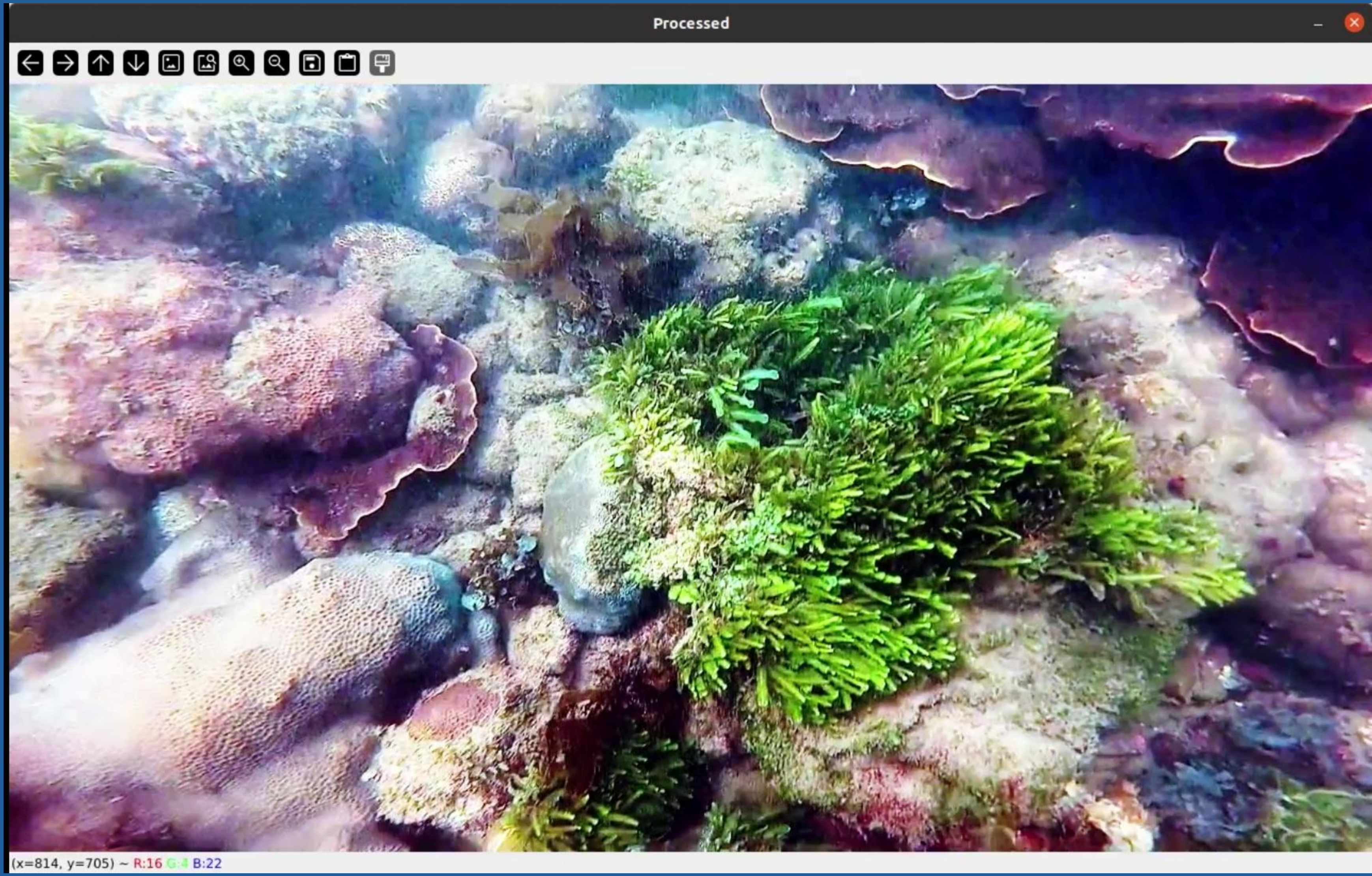
- Detects fishes using both contour-based detection and optical flow tracking.

Process:

1. Filters contours below a minimum area.
2. Calculates **centroids** of contours.
3. Optical flow for tracking, to estimate the motion between frames.
4. Dynamically updates the object's position based on tracked motion.

Problems and challenges with tracking approach:

- After identifying the centroid of each fish, **Optical Flow** was used to track it across frames.
- The idea is it would be better for objects that move quickly or are momentarily occluded.
- However, **noise** within the frames caused significant difficulties, making the algorithm “lose” the fish.



3. Challenges

- Sediments in the water caused a lot of noise.
- Between frames, it was challenging to determine whether the identified **contours** belong to the fish or to other elements, such as algae, hand movements, or even shifts caused by the camera itself.
- In the end, what we are detecting is motion, not necessarily fishes!

Thank you!

Artur Almeida 123196 (50%)
Cecília Santos 124056 (50%)