

Make It Aesthetic

Anna-Maria Hohmann
Konrad Kring
Meret Lindanis

5th presentation
17.12.2019





About Us

Meret

- computer science
- medical technologies

Anna

- business informatics
- project management

Konrad

- pedagogics
- music



Goals of Our Project

our motivation:

- interested in photography
- opening aesthetic photography to the public
- simplifying the aesthetic photography for the user
- being able to save every moment in beautiful photos
- bringing this knowledge into school

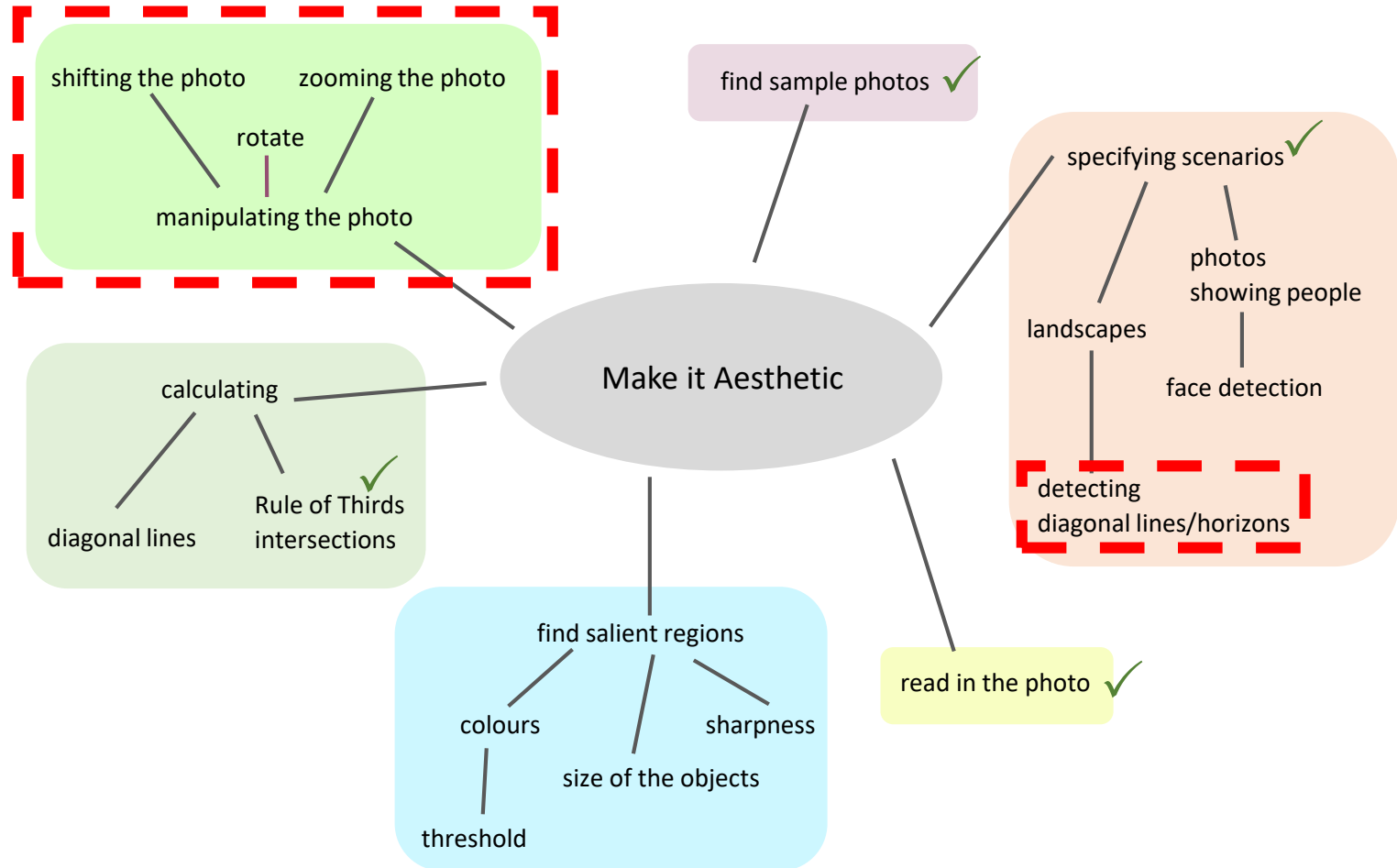


Goals of Our Project

- make given photos aesthetic
- by zooming, rotating or cropping the photo
- selecting the guideline the photo should follow



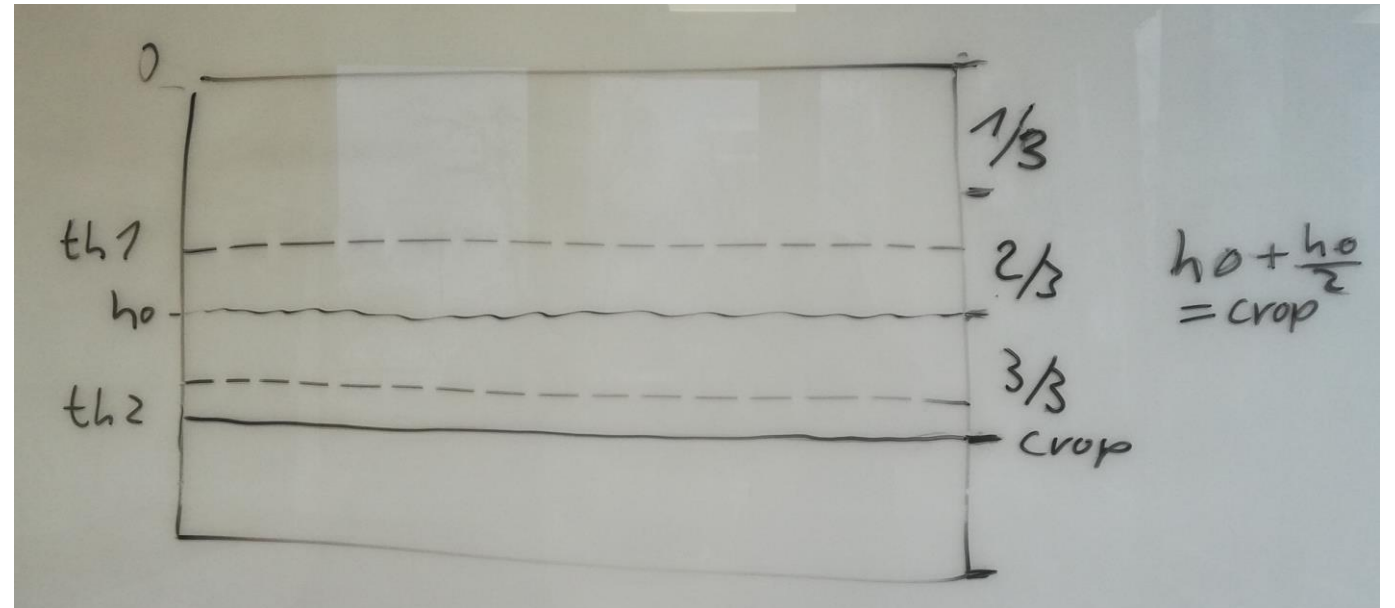
Milestones





Theory

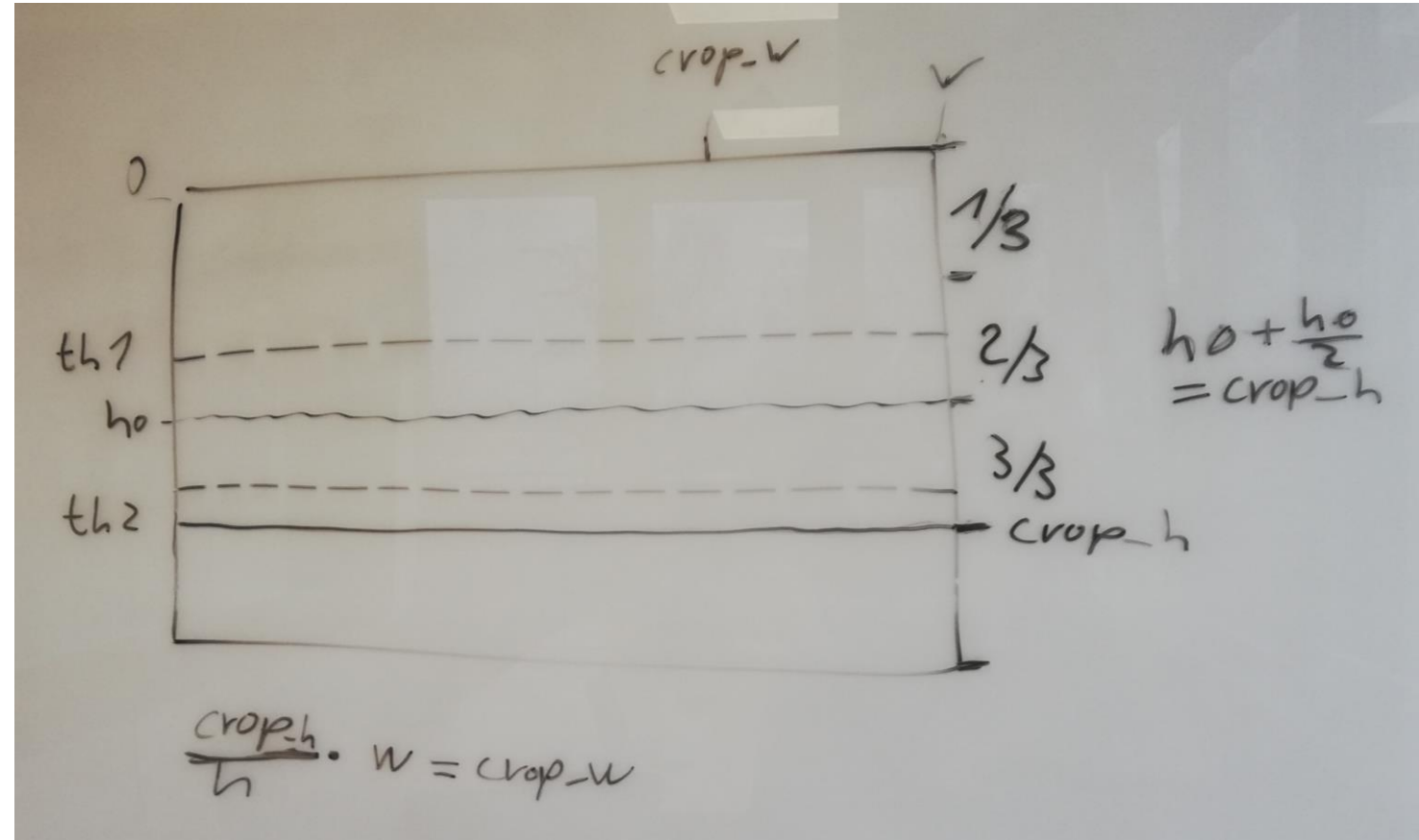
Cropping





Theory

Cropping





Implementation

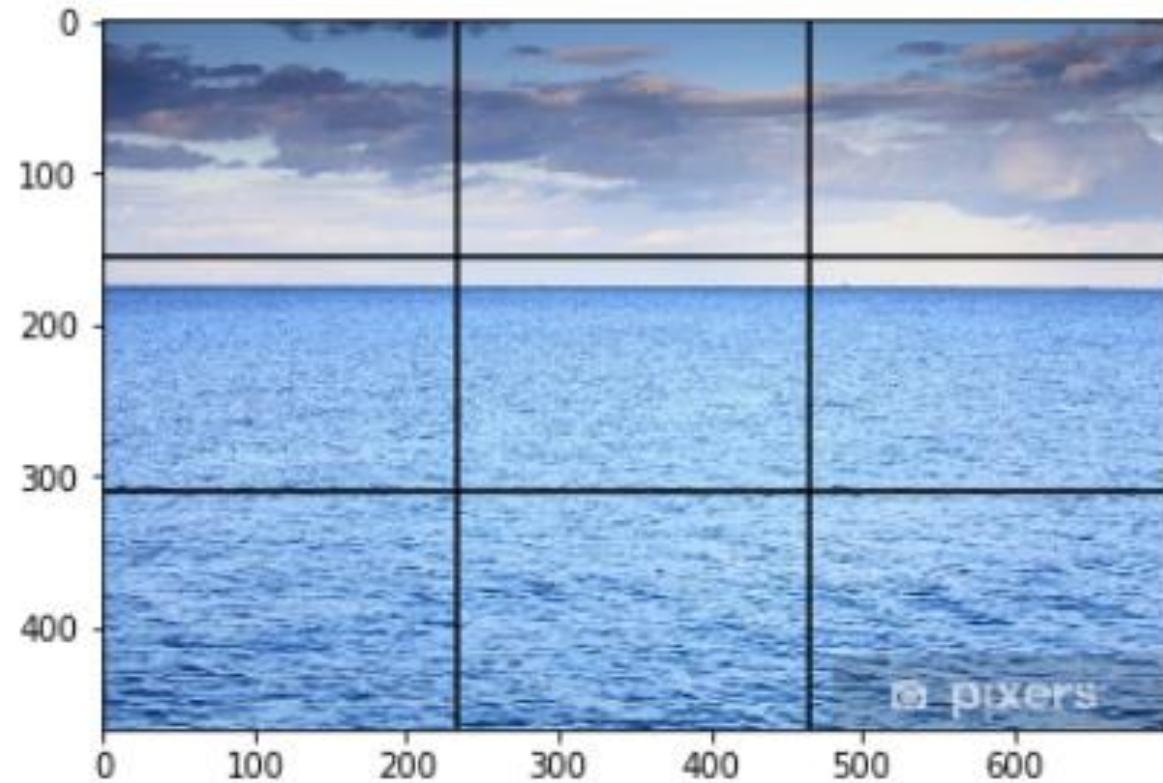
```
def crop_img(img):  
    h, w, third_of_height_1, third_of_height_2, third_of_width_1,  
    third_of_width_2 = generate_image_data(img)  
    img_line, edges, dilation, erosion, image_line, lines_edges,  
    lines = detect_horizon(img)  
    horizon_y1 = lines[0][0][1]  
    distance_horizon = third_of_height_2 - horizon_y1  
    x = 0  
    y = 0  
    cropping_point_y = horizon_y1 + (horizon_y1//2)  
    cropping_point_x = int((cropping_point_y/h)*w)  
    cropped_img = img[y:cropping_point_y, x:cropping_point_x]  
    return cropped_img
```




Implementation

Sample 1

Before cropping

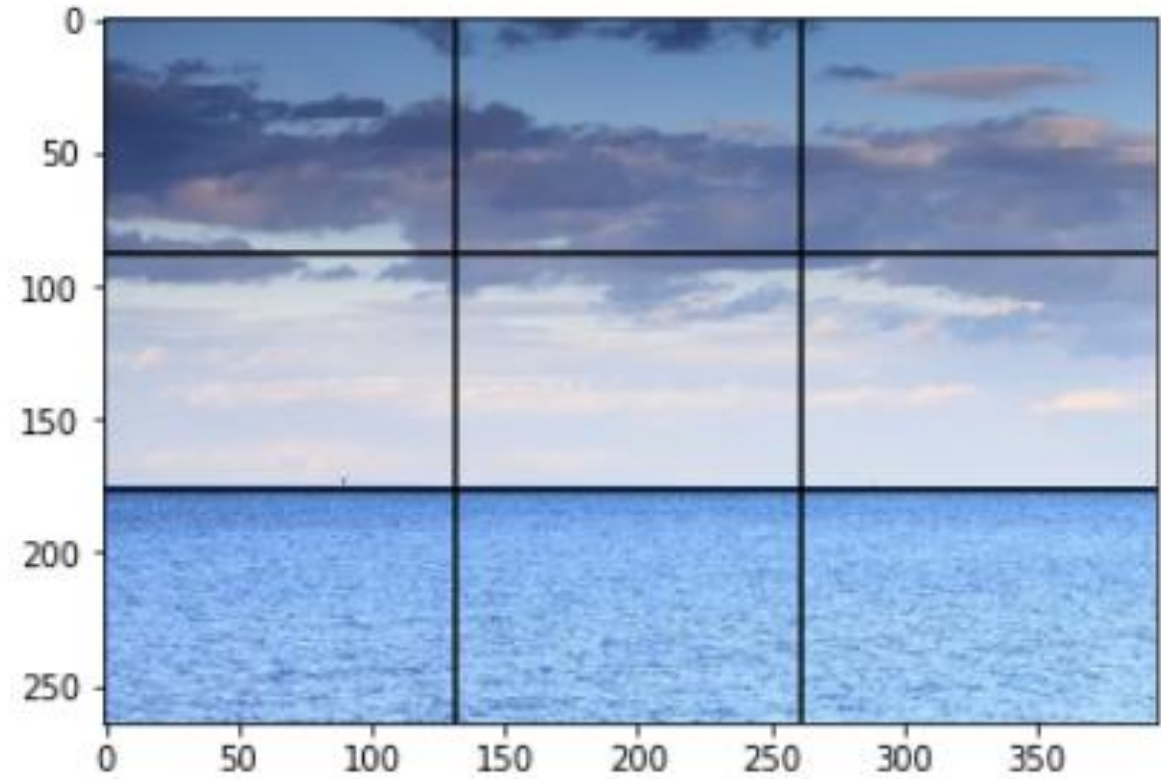




Implementation

Sample 1

After cropping





Implementation

Sample 2

Before cropping

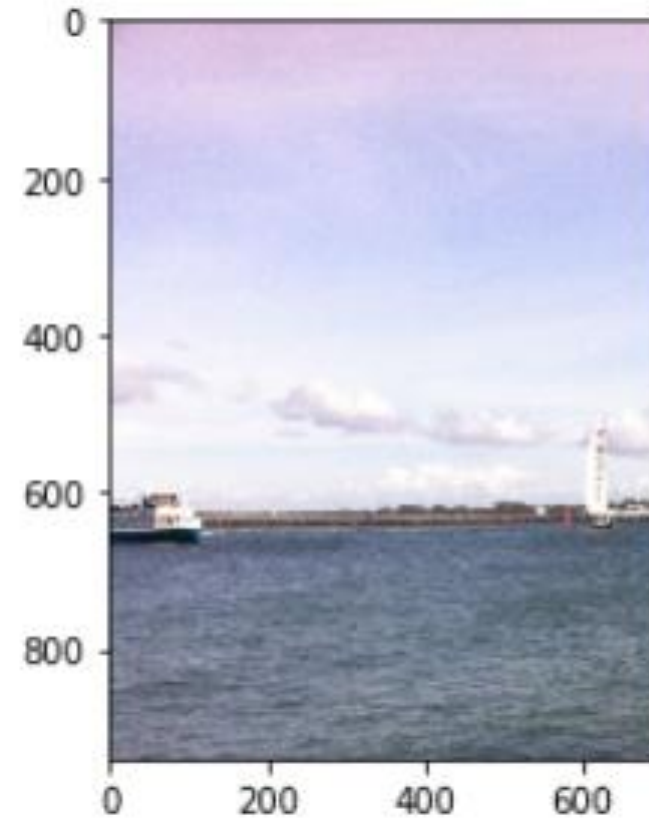




Implementation

Sample 2

After cropping





Implementation

Sample 3

Before cropping





Implementation

Sample 3

After cropping





Implementation

Sample 4

Before cropping





Implementation

Sample 4

After cropping





Implementation

Sample 4

Before cropping





Implementation

Sample 4

After cropping





Implementation

```
def roi(img, vertices):  
    #blank mask:  
    mask = np.zeros_like(img)  
    # fill the mask  
    cv2.fillPoly(mask, vertices, 255)  
    # now only show the area that is the mask  
    masked = cv2.bitwise_and(img, mask)  
    return masked
```

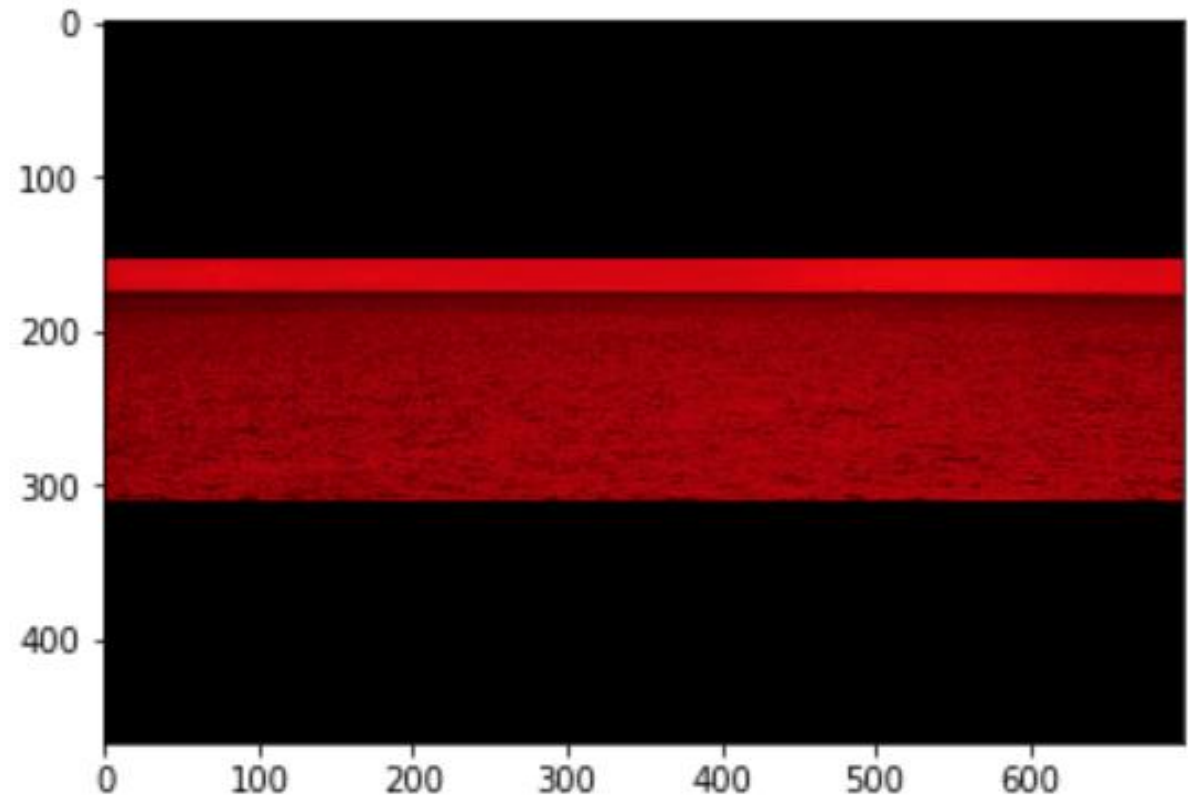
vgl. [\[1\]](#)

```
def process_img(img):  
    height, width, third_of_height_1, third_of_height_2, third_of_width_1,  
    third_of_width_2 = generate_image_data(img)  
  
    vertices = np.array([[0,third_of_height_1],[width,third_of_height_1],  
                        [width,third_of_height_2],[0,third_of_height_2]],  
                        np.int32)  
    processed_img = roi(img, [vertices])  
  
    return processed_img
```

vgl. [\[1\]](#)



Implementation





Implementation

Result





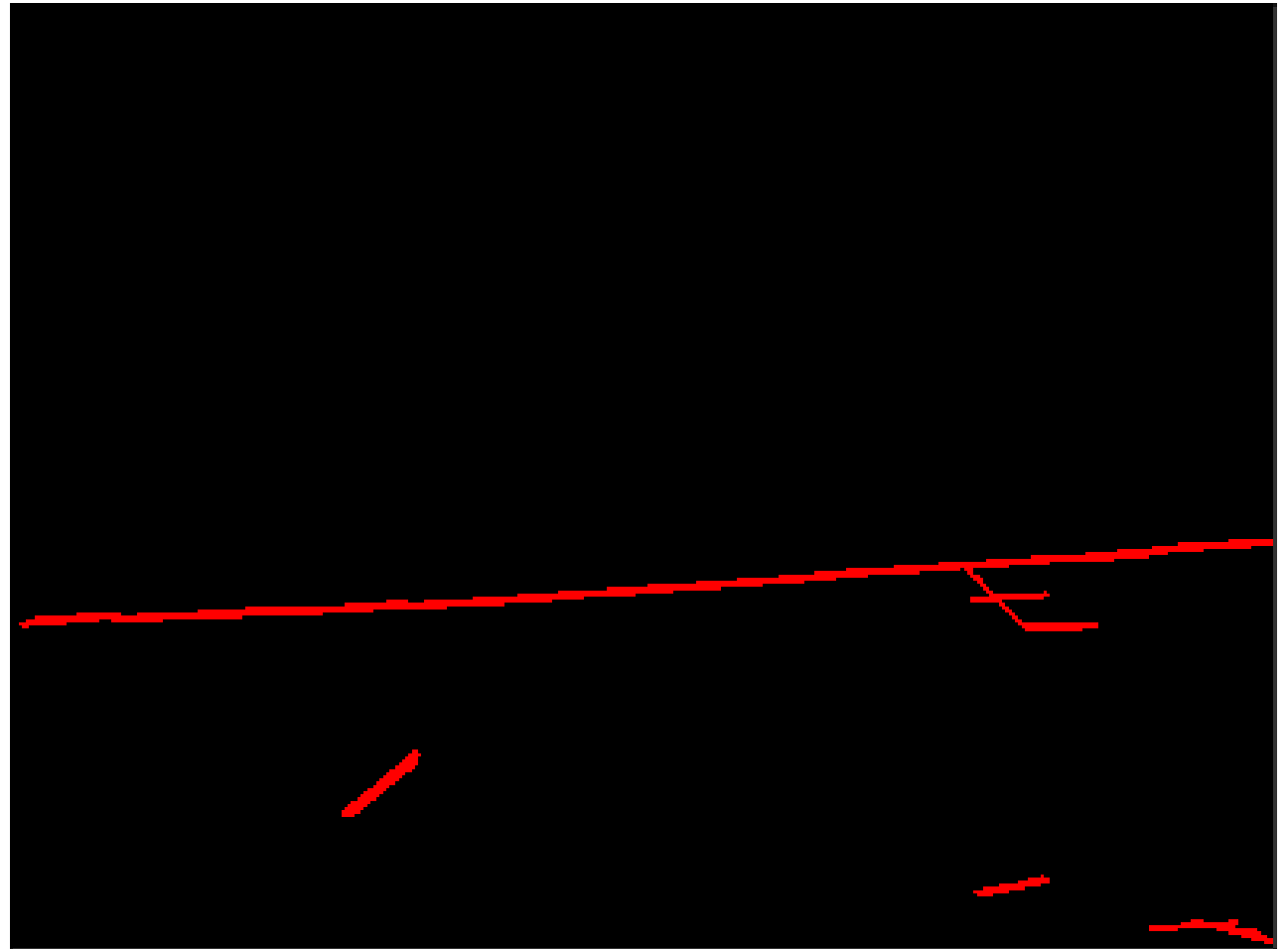
Implementation

Result



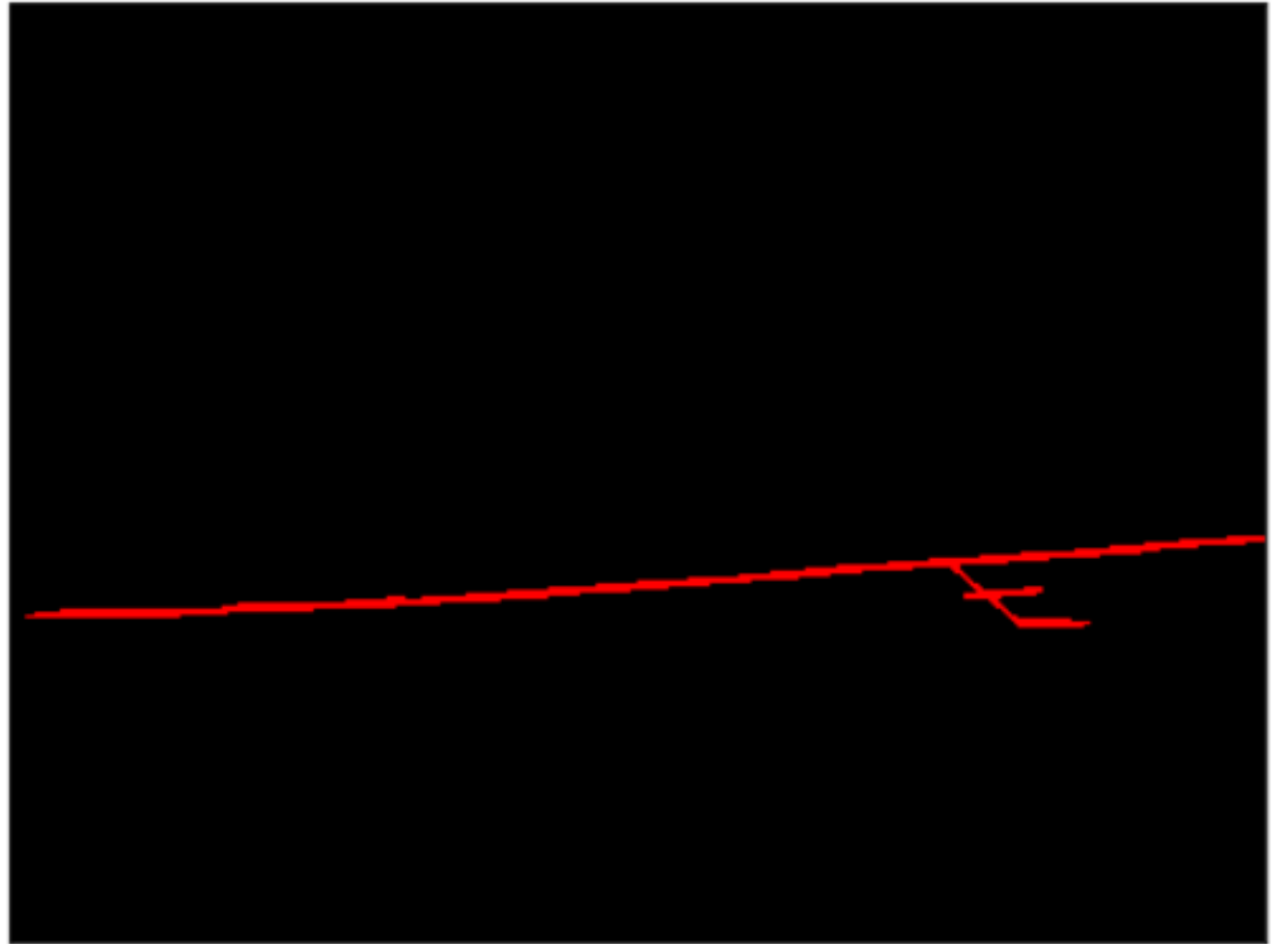


Implementation





Implementation





Challenges

- Concepts are not working properly (Accumulator array)
- Horizon is still not detected in just one line
- Unclear whether the first left point is detected or the first point from the top



Next Steps

- Decision unclear whether cropping should be on top or bottom
- Rotation of the image if the horizon is not straight
- Object detection



Sources

- [1] <https://pythonprogramming.net/lane-region-of-interest-python-plays-gta-v/>
- [2] <https://stackoverflow.com/questions/51009135/how-do-i-transform-the-values-of-an-accumulator-hough-transformation-back-to-a>

Make It Aesthetic

Anna-Maria Hohmann
Konrad Kring
Meret Lindanis

5th presentation
17.12.2019

