

Computer Vision 1: Exercise Sheet 8

Summary:

1. Hough transform for circle detection.
2. ORB feature detectors.

1 Circle detection

Figure 1 shows a selection of the last Finnish pre-euro coins (10, 5, and 1 marks; and 50 and 10 pennis, respectively). We apply the Hough transformation to detect the coins.



Figure 1: A selection of Finnish coins.

- Download the image `coins.jpg` from Moodle. Read it and convert to grayscale.
- The mint specifies that the diameter of the 5 mark coin is 24.5 millimetres¹. The resolution of the image is approximately 0.12 mm/pixel. Calculate the radius r of the coin in pixels.
- Apply the Canny edge detector to find edges in the grayscale image. Use the built-in function `skimage.feature.canny`. Visualize the edges and check that the outlines of the coins are detected.
- Use `skimage.transform.hough_circle` to calculate the Hough transform of the edge detection result. Use the radius you calculated above. Draw the result. You should obtain something similar to Figure 2 that peaks strongly around the center of the 5 mark coin.
- Using `skimage.transform.hough_circle_peaks`, select the two highest peaks from the Hough transform. Get all outputs from the function, i.e., your call should look like: `accums, cx, cy, radii = hough_circle_peaks(...)`.
- Apply `matplotlib.patches.Circle` to draw the circles at the coordinates found superimposed on the original image. See https://matplotlib.org/api/_as_gen/matplotlib.patches.Circle.html for more help. Use `from matplotlib.patches import Circle` to import the circle tool to your code.

¹The diameters of the coins shown in Figure 1 from left to right are: 27.25mm, 24.5mm, 22.25mm, 19.7mm, and 16.3mm.

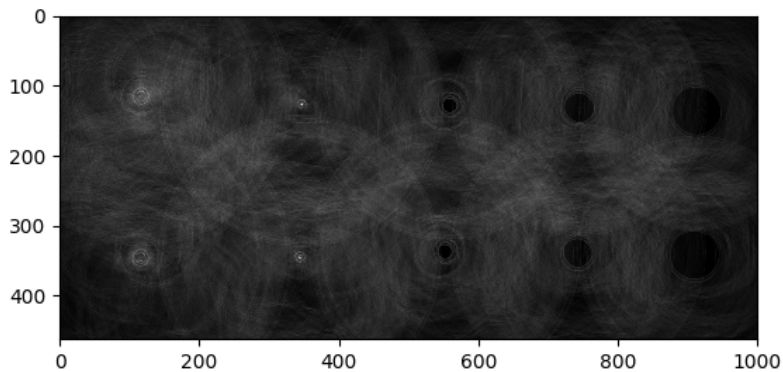


Figure 2: Circle Hough transform result. The strongest peaks highlight the 5 mark coin centers (compare to Figure 1).

2 ORB feature detectors

Download and read the image `Elbphilharmonie.jpg` on Moodle.

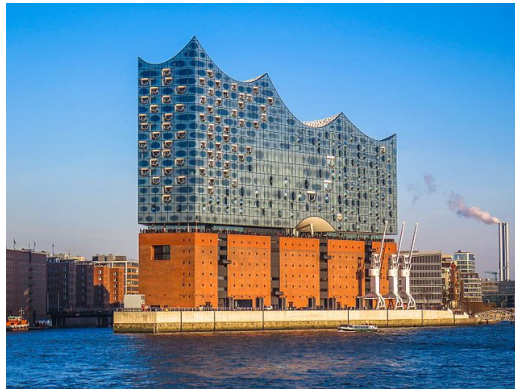


Figure 3: Hamburg Elbphilharmonie. Image source: Wikipedia

- Convert the image to grayscale image `im`.
- Using `skimage.transform.AffineTransform`, obtain a transformed image `im2` with the following parameters: shrink the dimensions by half, 20 degree counter-clockwise rotation, 300 pixels to the right and 300 pixels to the bottom translation.
- Visualize the images `im`, `im2`.
- Using `skimage.feature.ORB`, extract 100 ORB key points and descriptors of the three images above. Visualize the matching results.

Note: ORB is an efficient alternative for SIFT. Follow the example at http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_orb.html.