Brief explanation of static image processing:

Loading the Image: The image PennAir 2024 App Static.png is loaded using OpenCV’s cv2.imread function. If the image isn’t found, an error is printed.

Grayscale Conversion: The loaded image is converted to grayscale using cv2.cvtColor to simplify processing, as color may not be necessary for contour detection.

Noise Reduction: Gaussian blur is applied with cv2.GaussianBlur to smooth the image and reduce noise, making it easier to detect contours.

Adaptive Thresholding: cv2.adaptiveThreshold is applied to convert the blurred image into a binary image. This step helps in segmenting the shapes from the background.

Morphological Operations: Morphological closing is performed using cv2.morphologyEx with a rectangular kernel. This step helps close small gaps in the shapes' contours, making them more continuous.

Edge Detection: cv2.Canny is used to detect edges in the morphologically processed image. It finds the gradients (edges) that define the shape boundaries.

Contour Detection: cv2.findContours detects the contours (boundaries of shapes) in the edge-detected image. cv2.RETR\_TREE retrieves all contours and builds a hierarchy to distinguish outer and inner contours.

Contour Filtering: The contours are filtered based on their area to remove small or irrelevant contours. For each contour larger than the set threshold, its approximate shape is calculated.

Drawing Contours: The filtered contours are drawn on the original image in black using cv2.drawContours.

Marking Centroids: The centroid (center) of each contour is calculated using image moments (cv2.moments). A white dot is drawn at the centroid of each detected shape.

Displaying the Result: The processed image with contours and centroids is displayed using cv2\_imshow.