AIM 825 - Visual Recognition: Assignment 1

Pranav Kulkarni - IMT2022053

March 2025

1 Problem Statement

The objective of this project is to develop an image processing pipeline for edge detection, image segmentation, coin counting, and image stitching. The aim is to extract meaningful information by applying various techniques, optimizing parameters, and selecting the most effective methods. The pipeline enhances image quality, detects objects (coins), and seamlessly combines images to generate a panoramic view. The report discusses the attempted methodologies, their parameters, and the final approaches for each problem.

Figure 1: The input image containing the front and back sides of Rs. 0.50, Rs. 1, Rs. 2, Rs. 5 and Rs. 10 Indian coins. The resolution of the input image is 1742 × 854 pixels.

Figure 2: Marr - Hildreth

Figure 3: Sobel

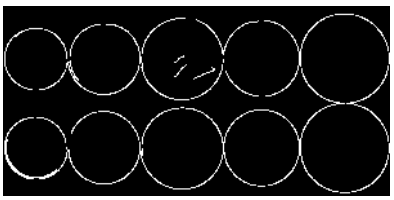

Figure 4: Canny

2 Edge Detection

Several edge detection methods were explored:

- **Marr-Hildreth Edge Detection:** Gaussian blur (kernel size = 5, $\sigma = 1.4$) was applied, followed by the Laplacian operator to detect intensity changes.
- **Sobel Edge Detection:** Computed horizontal and vertical gradients using kernel size 3, combined into an edge image.

Marr - Hildreth and Sobel were not as good as canny edge detectors because they gave incomplete edges along with finer details in the coins.
The final approach was **Canny Edge Detection**. CLAHE-enhanced images were smoothed with Gaussian blur (kernel size = 17) before applying Canny edge detection (low threshold = 0, high threshold = 216). Detected edges were dilated using a 5×5 kernel and superimposed on the original image.

3 Image Segmentation

Various segmentation techniques were attempted:

- **Manual Thresholding:** A fixed threshold (250) decided based on the histogram of the original image converted grayscale images into binary images but was sensitive to illumination changes.
- **Adaptive Thresholding:** Computed region-specific thresholds using a Gaussian-weighted sum for robustness.
- **Otsu's Thresholding:** Determined the optimal threshold by maximizing inter-class variance. Otsu's threshold came out to be 202.0.

Manual thresholding provided a lot of rough edges along the coins, adaptive thresholding gave a grainy segmentation. Otsu's was the best among the thresholding methods giving smooth thresholding.
The final approach, **Watershed Segmentation**, involved CLAHE contrast enhancement (clip limit = 2.50, tile grid = 6), Gaussian blur (kernel size = 55), morphological operations (63 iterations of opening, 147 iterations of dilation), and watershed markers for final segmentation.

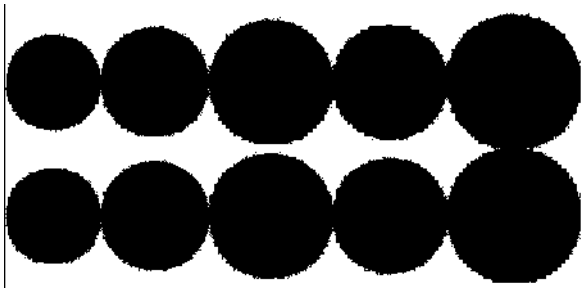

Figure 5: Manual Thresholding

Figure 6: Adaptive Thresholding

Figure 7: Otsu Thresholding

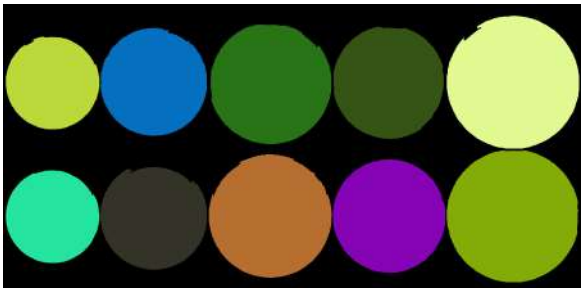

Figure 8: Watershed Segmentation

4 Counting Coins

Figure 9: Hough Circle Transform

The techniques were explored:

- **Hough Circle Transform:** Detected circular shapes (dp = 1.2, minDist = 100, param1 = 105, param2 = 30, min radius = 69, max radius = 200).

The final approach was the **Hough Circle Transform**, which demonstrated superior accuracy in detecting coins under varying conditions.

Figure 10: Input Image 1

Figure 11: Input Image 2

Figure 12: Input Image 3

5 Image Stitching

The pipeline followed these steps:

1. **Keypoint Extraction:** Used SIFT to extract invariant keypoints.
2. **Keypoint Matching:** Matched keypoints using BFMatcher with Lowe's ratio test (threshold = 0.75).
3. **Homography Estimation:** Computed the homography matrix using RANSAC (reprojection threshold = 4.0).
4. **Warping and Blending:** Warped image using the homography, applied an alpha blend (alpha = 0.5), and cropped the final panorama.

The choice of SIFT and robust keypoint matching resulted in a smooth and well-aligned panorama.

Figure 13: Stitched Image

6 Conclusion

This report explored edge detection, segmentation, coin counting, and image stitching. The final approaches—Canny edge detection, watershed segmentation, Hough circle transform, and SIFT-based stitching—were selected based on accuracy and robustness. Parameter tuning and

preprocessing ensured reliable results, leading to an integrated pipeline for advanced image processing.

7 References

1. OpenCV Documentation: `https://docs.opencv.org/4.x/index.html`
2. First Principles of Computer Vision - YouTube: `https://www.youtube.com/@firstprinciplesofcomput`