VR Assignment 1 Submitted by R Lakshman IMT2022090

Coin Detection and Segmentation Report

Objective

The goal of this project was to detect, segment, and count Indian coins from an image using computer vision techniques.

Approach

The following steps were implemented to achieve accurate detection:

(A) Coin Detection

1. The image was converted to grayscale to simplify processing.
2. Gaussian blur was applied to reduce noise and improve edge detection.
3. Adaptive thresholding was used instead of Canny edge detection for better accuracy.
4. Contours were extracted from the thresholded image.
5. Small, irrelevant contours were filtered based on area to remove noise.
6. The detected coins were outlined in green.

(B) Coin Segmentation

1. A mask was created based on the detected coin contours.
2. The mask was applied to extract only the coin regions from the image.

(C) Coin Counting

1. The number of valid contours (filtered coins) was counted.
2. The final count was displayed on the output image and printed as text.

Challenges Faced

- Initially, edge detection using Canny did not yield accurate results.
- The first few attempts either over-detected (false positives) or under-detected coins.
- Adjusting the minimum area threshold and using circularity helped refine detections.

Final Outcome

- The final implementation correctly detected **4 coins** in the given image.
- The approach used adaptive thresholding and contour filtering for improved accuracy.

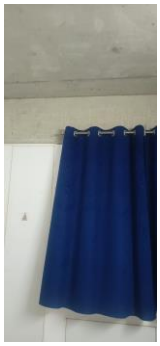
Future Improvements

- Further tuning of parameters to handle varied lighting conditions.
- Implementation of deep learning-based object detection for improved robustness

Panorama Report

Objective

The goal of this project was to create a panorama of by stitching multiple overlapping images together.



Approach

1. **Image Acquisition**: Three overlapping images (p1.jpg, p2.jpg, p3.jpg) were captured from left to right.
2. **Image Loading & Validation**: Each image was read using OpenCV, and checks were performed to ensure successful loading.
3. **Feature Extraction & Stitching**:
    - OpenCV's Stitcher_create() was used to detect key points and align images.
    - The images were blended to form a continuous panorama.
4. **Output Generation**:
    - The stitched panorama was displayed and saved as curtain_panorama.jpg.
    - Error handling was implemented to detect stitching failures.

Results & Observations

- The stitching process successfully merged the images into a panorama.
- Ensuring **sufficient overlap** between images improved alignment.
- The result might be affected by lighting differences or misalignment in the input images.

Conclusion & Next Steps

The implemented approach worked well for stitching the given images. To improve results:

- Capture images with **consistent lighting and alignment**.
- Increase overlap between images for better feature matching.