VR Assignment 1 Report

1. Coin Detection

Objective: To detect and segment individual coins in an image.

Approach:

The provided code utilizes edge detection and contour finding techniques to identify coins. Here's a breakdown of the process:

1. Image Loading and Preprocessing: The image is loaded in grayscale to simplify edge detection. Gaussian blurring is applied to reduce noise.

2. Edge Detection: The Canny edge detector is used to identify edges in the image.

3. Contour Detection: Contours (outlines) are detected in the edge image.

4. Filtering and Visualization: Contours are filtered based on their area to eliminate noise. The remaining contours, representing coins, are highlighted on the original image.

5. Segmentation: Each coin is segmented by creating a mask from its contour and applying the mask to the original image.

Observations:

- The code successfully detects and segments the four coins present in the image.

- The use of edge detection and contour finding is effective in identifying the circular shapes of the coins.
- Filtering based on contour area helps to eliminate noise and focus on the relevant objects.
- The segmentation process accurately isolates each coin from the background.

Limitations:

- The current approach relies on distinct edges between the coins and the background. It might struggle with images where the coins blend with the background or have low contrast.
- The threshold for filtering contours based on area is manually set and might need adjustment for different images.

Results:

2. Image Stitching

Objective: To stitch multiple images into a seamless panorama.

Approach:

The provided code utilizes the OpenCV Stitcher class to create a panorama from a set of images. The process involves:

1. Image Loading and Keypoint Detection: Images are loaded from a folder, and ORB (Oriented FAST and Rotated BRIEF) is used to detect keypoints and compute descriptors.

2. Keypoint Filtering: Keypoints are filtered based on their response (strength) to retain only the most prominent ones.

3. Image Stitching: The OpenCV Stitcher class is used to stitch the images together based on the detected keypoints and descriptors.

Observations:

- The code successfully stitches the images in the specified folder into a panorama.
- The use of ORB for keypoint detection is efficient and provides good results.
- Filtering keypoints based on response helps to improve stitching accuracy.

Limitations:

- The success of stitching heavily relies on the quality of the input images and the presence of overlapping regions with sufficient keypoints.

- The current approach might struggle with images that have significant differences in exposure or perspective.

Results:


img1.png


img2.png


img3.png


img4.png


img5.png


img6.png

Keypoints:


result1.png


result2.png


result3.png


result4.png


result5.png


result6.png

Stitched image: