VR Assignment 1

Anurag Ramaswamy
IMT2022103

March 2, 2025

Introduction

This report contains details about the two parts of VR Assignment 1. The first part involves detecting, segmenting and counting coins from an image. The second part stitches multiple overlapping images to create a panorama.

Part 1 - Coin Detection and Counting

Objective

The goal of this task is to detect, segment and count coins in an image containing scattered Indian coins using computer vision techniques.

Methodology

1. **Preprocessing:** The input image is converted to grayscale and resized for uniform processing. A Gaussian blur is applied to reduce noise and adaptive thresholding is used to create a binary image.

2. **Edge Detection:** Contours are identified using the OpenCV `findContours` function to detect circular edges corresponding to coins.

   Coins with $0.7 <$circularity < 1.2 (close to 1) and area $>$ min_coin_size are selected where Circularity$= \frac{4\pi \cdot \text{Area}}{\text{PERIMETER}^2}$ and min_coin_size $= 500 \cdot (\text{scale}^2)$.

3. **Segmentation:** A mask is created to extract only the coin regions, isolating them from the black background and the contours of the coin are highlighted in red.

4. **Counting Coins:** The number of detected contours satisfying coin-like properties is counted and displayed on the terminal.

Results, Observations and Experiments

Each detected coin is outlined, and the segmented output is saved with the total count displayed on the terminal. The code performed perfectly with different camera zooms and number of coins. However, when I tried it with overlapping / extremely closely spaced coins and dark backgrounds it gave a few errors. Figure 1 shows the output when the input contained 10 coins. I tried different values for scale,circularity and minimum area, adjusting them to find the best parameters that accurately detected coins while minimizing false positives. I kept the values that worked best for my images.

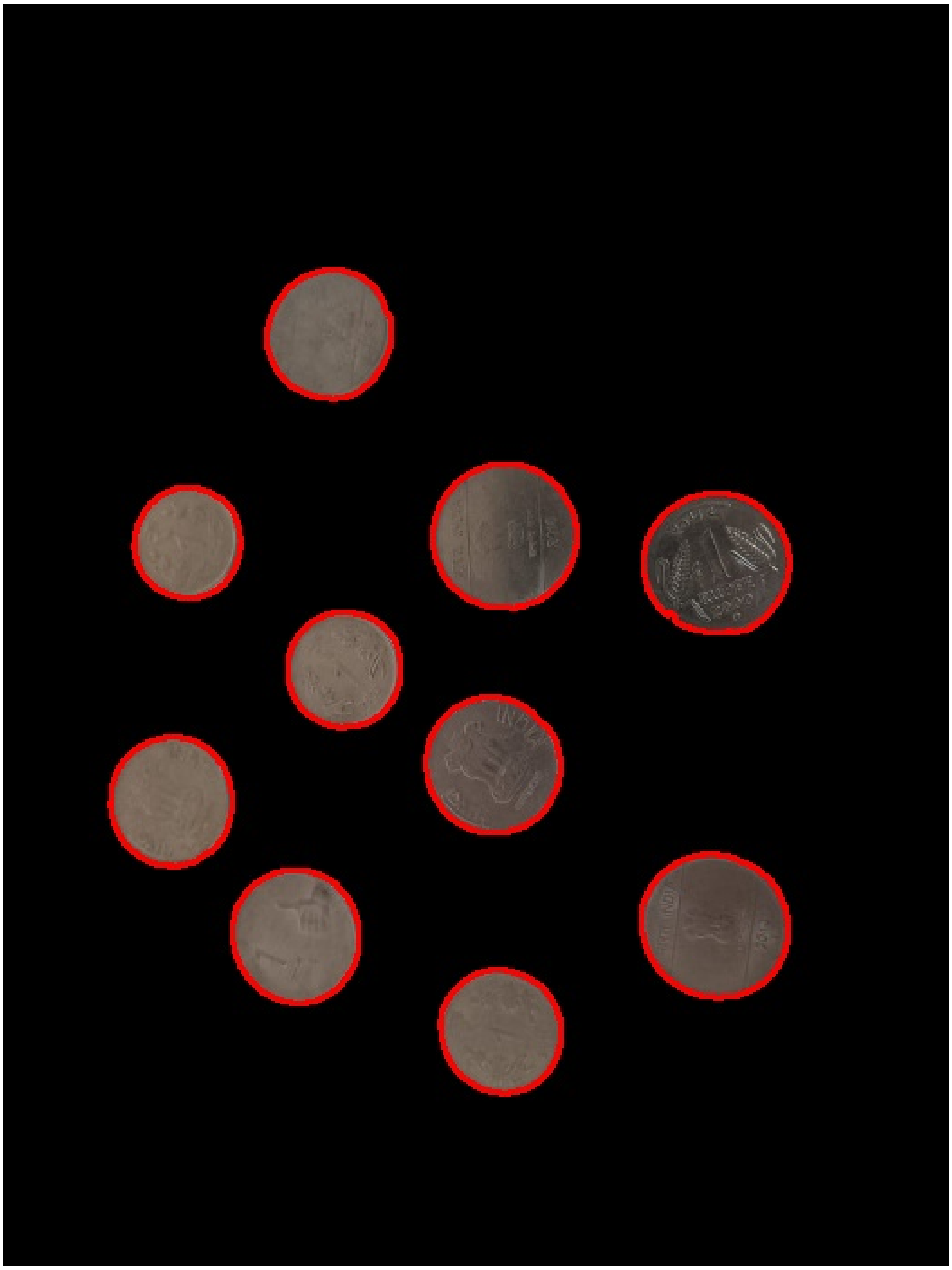


Figure 1: Output image showing detected coins with red contours

Part 2 - Panorama Stitching from Overlapping Images

Objective

The goal of this task is to generate a stitched panorama from multiple overlapping images using feature-based image alignment techniques.

Methodology

1. **Feature Extraction:** SIFT (Scale-Invariant Feature Transform) is used to extract key points and descriptors from each image.

2. **Keypoint Matching:** The BFMatcher algorithm is used to find correspondences between images, filtering matches using Lowe's ratio test.

3. **Homography Estimation:** RANSAC is applied to estimate the homography matrix, aligning images based on the best matches.

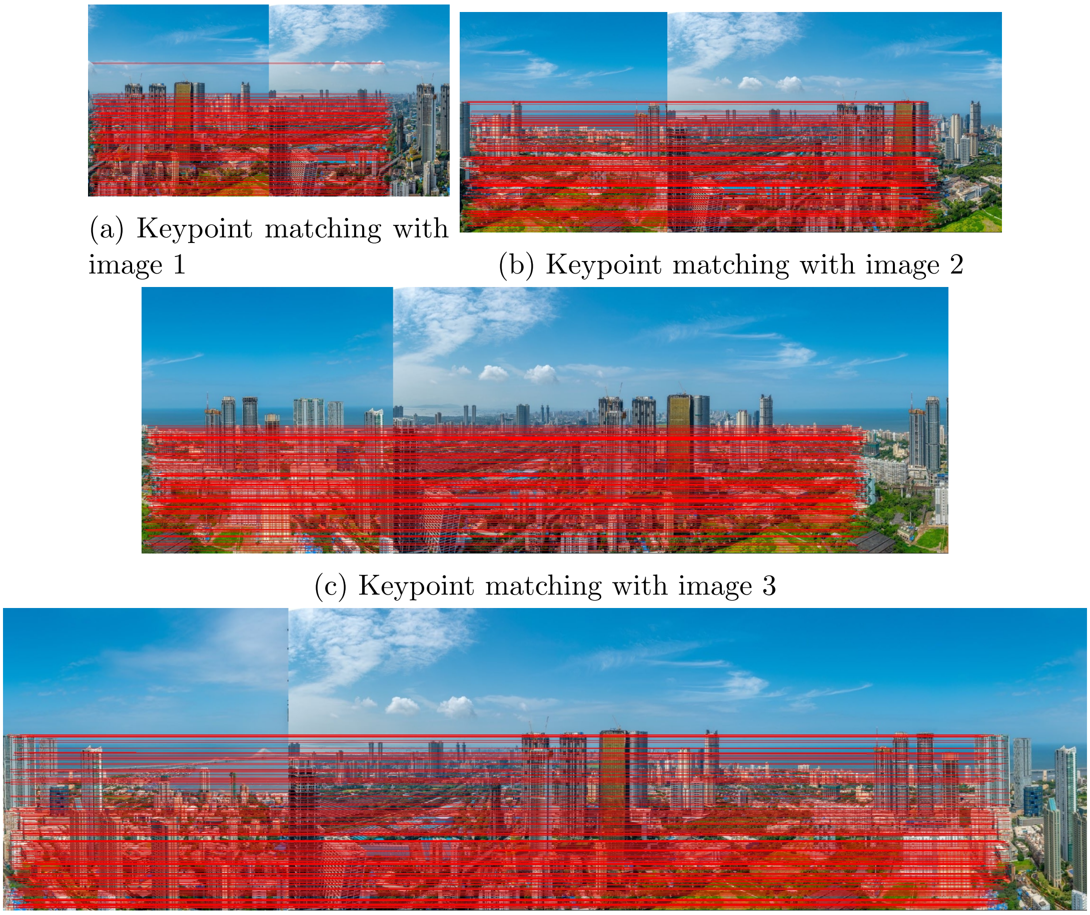
4. **Image Warping and Blending:** The aligned images are combined using perspective transformation, followed by cropping to remove unwanted black borders.

Results, Observations and Experiments

The final stitched panorama image is generated, with intermediate keypoint match visualizations saved. Figure 2 shows all the intermediate keypoint matches for a set of images and Figure 3 shows the final panorama image for this set of images. I experimented with Lowe's ratio threshold with values from 0.6 to 0.8 and found that 0.75 provided the best balance between filtering out false matches and retaining good ones. The RANSAC reprojection threshold was also adjusted between 3.0 and 5.0, with 4.0 proving optimal for maintaining accuracy without losing too many correspondences. To improve keypoint matching visualization, I limited the number of lines drawn to 250 which prevented visual clutter while still effectively showing key correspondences.

Repository

The code and additional details on how to run it can be found in the GitHub repository: VR_Assignment1_AnuragRamaswamy_IMT2022103.



(a) Keypoint matching with image 1

(b) Keypoint matching with image 2

(c) Keypoint matching with image 3

(d) Keypoint matching with image 4

Figure 2: Keypoint matching results

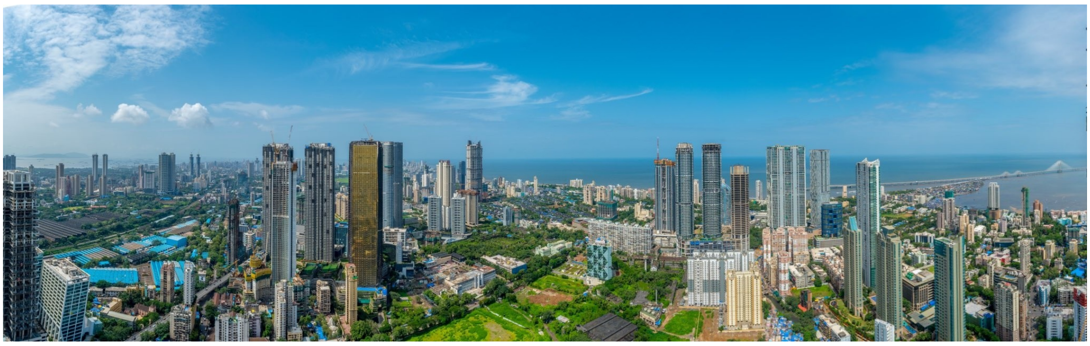


Figure 3: Final panorama image