VR Assignment 1

Ananthakrishna K

February 25, 2025

1 Introduction

This report presents the implementation details of two VR assignment tasks: Coin Detection and Panorama Stitching. The first task involves detecting and counting coins in images, while the second task stitches multiple images to create a panorama. Both tasks are implemented using OpenCV and Python.

2 Coin Detection and Counting

2.1 Overview

The objective of the coin detection task is to identify, segment, and count the number of coins in a given image. This is achieved through image preprocessing, contour detection, and circularity filtering.

2.2 Implementation Details

1. **Preprocessing:**

    (a) The input image is converted to **grayscale** to reduce computational complexity while retaining structural details.

    (b) The image is **resized** while maintaining the **aspect ratio** to ensure uniform processing.

    (c) **Gaussian blur** is applied to smooth out noise and improve contour detection.

    (d) **Adaptive thresholding** is used to convert the image into a binary format for better segmentation.

2. **Coin Detection:**

   (a) **Contours** are extracted from the binary image using OpenCV's contour-finding algorithm.

   (b) A **circularity metric** is computed for each contour using the formula:

   $$\text{Circularity} = \frac{4\pi \times \text{Area}}{\text{Perimeter}^2}$$

   (c) Contours with a **circularity close to 1** are classified as coins.

3. **Segmentation and Marking:**

   (a) A binary mask is created to segment the detected coins from the background.

   (b) The contours of the detected coins are outlined in red for visualization.

   (c) The total count of coins is displayed on the output image.

2.3 Results

The implemented code correctly identifies all the coins in the images provided in the input folder of the submitted code.
Figure 1 shows an example of coin detection.

Figure 1: Detected coins marked with red contours.

3 Panorama Stitching

3.1 Overview

The objective of the panorama stitching task is to combine multiple images into a seamless panoramic image. The process involves keypoint extraction, feature matching, homography estimation, and image warping.

3.2 Implementation Details

1. **Feature Extraction and Matching:**

   (a) Keypoints and descriptors are extracted from each image using the SIFT algorithm.

(b) Feature matching is performed using a brute-force matcher to find correspondences between images.

(c) Lowe's ratio test is applied to filter out weak matches and retain only the most reliable ones.

2. **Homography Estimation:**

   (a) The homography matrix is computed using RANSAC, which helps eliminate outliers.

   (b) The transformation matrix ensures proper alignment of overlapping images.

3. **Image Warping and Stitching:**

   (a) The overlay image is warped using the estimated homography matrix to align it with the base image.

   (b) The images are merged together, ensuring that overlapping regions blend smoothly.

   (c) The final panorama is cropped to remove any unwanted black borders.

3.3 Results

Figure 2 shows an example of keypoint matching between two images. Figure 3 presents the final stitched panorama.

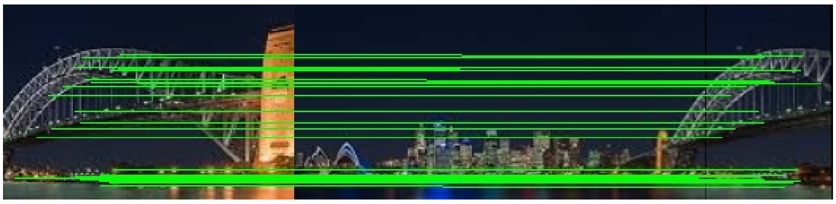

Figure 2: Keypoint matchings.

Figure 3: Final stitched panorama.

4 Repository

The complete implementation and additional details can be found in the GitHub repository: VR_Assignment1_Ananthakrishna-K_IMT2022086.