AIM 825 Assignment 1

IMT2022038

March 1, 2025

Contents

1 Introduction **2**

2 Part 1: Coin Detection and Counting **2**

2.1 Input and Preprocessing 2

3 Part 2: Image Stitching and Blending **3**

1 Introduction

The assignment consists of two parts:

1. **Part 1:** Detecting coins in an image using Canny Edge Detection, segmenting the coins, and counting the number of coins.
2. **Part 2:** Stitching and blending images to form a panorama using three images.

2 Part 1: Coin Detection and Counting

2.1 Input and Preprocessing

A Gaussian blur (5x5) was applied to the input image to reduce noise.

Figure 1: Input Image

The process consists of three main steps:

(a) **Coin Detection using Edge Detection:** Canny Edge Detection was applied.

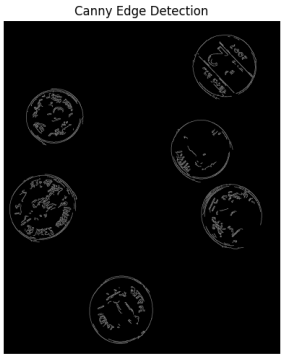

Figure 2: Edge-detected Image

(b) **Segmentation of Each Coin:** The segmentation process involved thresholding followed by morphological closing to fill holes within the segmented coins. Contours were extracted using OpenCV's `findContour` method. To eliminate noise, contours with an area smaller than ten pixels were discarded. The remaining contours were filled and displayed individually, generating the required masks for each coin.

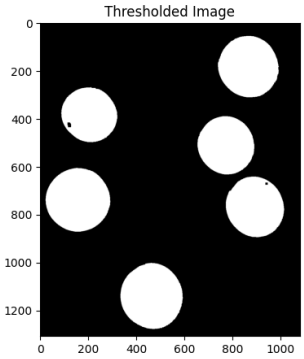

Figure 3: Thresholded Image (After Morphological Operations)

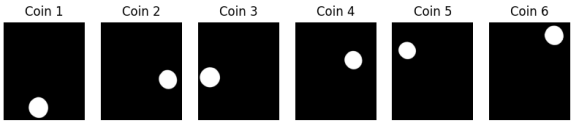

Figure 4: Masks for Individual Coins Obtained by Contours

(c) **Counting the Total Number of Coins:** The number of coins was determined by finding the length of the list of filtered contours.

3 Part 2: Image Stitching and Blending

The `create_panorama` function was used twice to create a panorama from three images.

1. **Extracting Key Points:** SIFT was used to compute keypoints and descriptors for both images.
2. **Image Stitching:**
   - Keypoint matching was performed using a brute-force KNN matcher (k=2) which provides for each point, the k best matches.
   - Lowe's ratio test was used to refine the matches (Fig. 5). The ratio test computes the following ratio:
     $$\text{ratio} = \frac{\text{distance of Match 1}}{\text{distance of Match 2}}$$
     where Match 1 is the best match and Match 2 is the second-best match for a given keypoint.
     The match is considered valid only if the condition
     $$\text{ratio} < \text{threshold}$$
     is met. The threshold was set to 0.75 in the implementation.
     The underlying intuition is that a ratio close to 1 indicates ambiguity, as the best match is not significantly better than the second-best match. Conversely, a smaller ratio suggests a larger difference between the matches, implying that the best match is much more reliable.
   - Then the homography matrix was computed, and `warpPerspective` was applied to transform the right image.
   - Blending was achieved using a weighted average of pixel values from both images, with the weights derived from distance transforms. The distance transform assigns higher weights to pixels farther from the image edges, promoting smooth transitions between images.

- The weights for the right image were computed before warping and then transformed using the same homography as the image. For the left image, the weights were computed and then padded with zeros to match the dimensions of the final panorama. This padding ensured proper alignment of weights with the blended image.
- The blending operation was implemented using the `blend` method, while the distance transform was computed using the `distance_transform` method.

Figure 5: Good Matches for middle.jpeg and right.jpeg

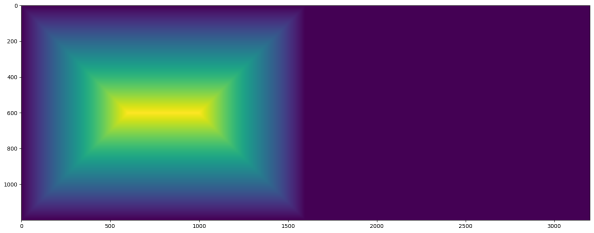

Figure 6: Padded weights for the Left Image

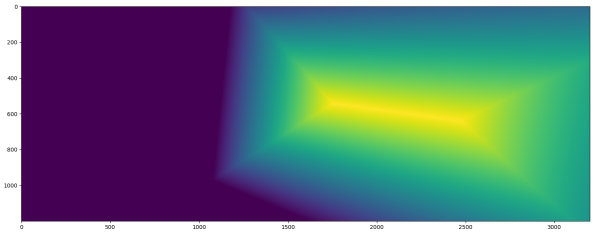

Figure 7: Warped weights for the Right Image

Figure 8: Blended Image for middle.jpeg and right.jpeg