ASSIGNMENT-1

SREYAS JANAMANCHI (IMT2022554)

GithubLink

PART-1

Image Preprocessing

- The image is resized while maintaining its aspect ratio, ensuring the maximum dimension does not exceed 700 pixels.
- Resized image improves computational efficiency.

Coin Detection

Final Approach

- The image is converted to grayscale.
- Gaussian blur is applied to reduce noise and enhance edge detection.
- Adaptive thresholding is performed to create a binary image.
- Morphological closing is applied to fill small gaps and refine the coin boundaries.
- Contour detection is performed to identify circular regions corresponding to coins.
- A circularity-based filter is used to eliminate erroneous contours and refine the detection.

Attempted Approaches

- Several kernel sizes were tested for Gaussian blur to determine the optimal choice for noise reduction and edge enhancement.
- Different thresholding methods were explored before selecting adaptive thresholding:-
    - **Adaptive thresholding** is better for images with varying lighting conditions.
    - **Otsu's method** is good for images with a clear bimodal histogram.
    - **Fixed thresholding** works well if you can manually determine a good threshold value.
- Although Otsu's method provided good standalone results, adaptive thresholding combined with morphological closing produced even better outcomes.
- To fine-tune the circularity filter, multiple parameter values were tested, and the best combination was chosen based on observed results.

Coin Segmentation

- Contours of detected coins are drawn on the image to visualize the results.
- A mask is created to isolate each coin from the background.
- Each detected coin is extracted and saved as an individual segmented image.

Coin Counting

- The total number of detected coins is counted based on the number of segmented objects.
- The count is printed as an output.

PART-2

Image Preprocessing

- Resized image improves computational efficiency.

Extracting Key Points

- The Scale-Invariant Feature Transform (SIFT) algorithm is used to detect key points in the overlapping images.
- The detected key points and corresponding feature descriptors are extracted for further matching.

Matching Key Points

Final Approach

- A brute-force matcher with k-nearest neighbors (k-NN) is utilized to find corresponding key points between consecutive images.
- The Lowe's ratio test is applied to filter out incorrect matches.
- Homography transformation is estimated using LMedS to align the images.

Attempted Approaches

- There is an alternative for KNN which was tried: Threshold-Based Distance Filtering. The KNN method gave more matches.
- Homography estimation was attempted multiple methods before finalizing **LMedS**:-

    - **RANSAC**: It works as well as LMeds, it has an extra parameter when compared with LMedS.
    - **LMedS** when there are many outliers.
    - **DLT**: It gave incorrect matches, this might be because it isn't noise resilient.
- Fine-tuned the ratio test parameter to improve match quality.

Image Stitching Process

- The first image is set as the reference.
- Each subsequent image is aligned with the previous one using the computed homography matrix.
- The images are combined using a perspective warp transformation.
- The final panorama is cropped to remove unwanted black regions.