BBM418 Computer Vision Lab.
Assignment 2: Image Panorama Stitching
Spring 2023

Esad Boran
21827206
Department of Computer Engineering
Hacettepe University
Ankara, Turkey
esatsuphi@gmail.com

Overview

The aim of this assignment is to develop a feature matching algorithm that uses various feature extraction and matching techniques. This algorithm will find correspondences between keypoints in two input images, which can be applied to tasks such as image stitching, object detection, and image recognition. The algorithm will employ popular feature extraction techniques like SIFT, SURF, and ORB, as well as matching methods like k-NN matching and ratio matching to find the best matches between the keypoints in the images. The developed algorithm will be evaluated on different datasets to assess its performance and accuracy.Then, the extracted keypoints will be compared and matched to merge the sub-images into one panorama image.

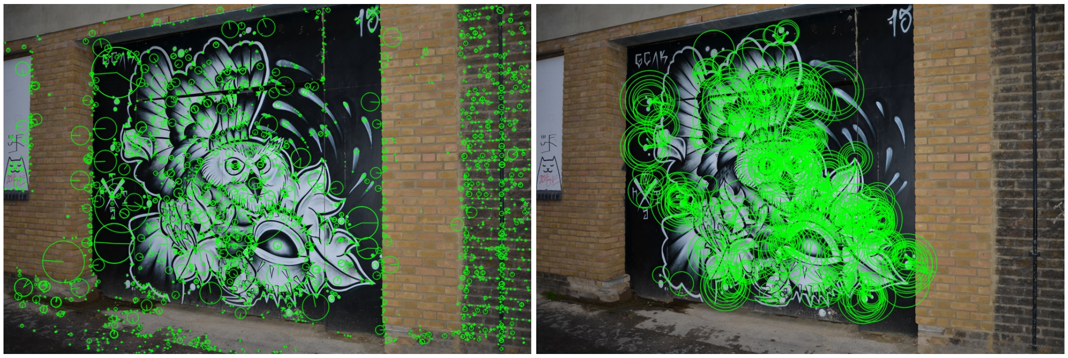
1 Part 1 - Image Panorama Stitching Algorithm

The Image Panorama Stitching Algorithm section presents the steps involved in merging multiple sub-images into a single panorama image. This algorithm is based on the extraction of key features in the input images, followed by the matching of corresponding features across the images, and the computation of a homography matrix to map the features between the images. Finally, the images are merged and transformed into the final panorama using the homography matrix. Each of these steps will be described in detail in the following subsections.

1.1 Feature Extraction

Feature extraction is the process of extracting salient information from an image that can be used for further analysis. In this subsection, we extract features from two input images using SIFT, SURF, or ORB feature extraction methods. We use OpenCV's built-in feature detection and extraction functions to compute the keypoints and descriptors for both input images. The output of this step is a tuple containing the keypoints and descriptors for both images, which will be used for feature matching in the next subsection.

These images display the locations of the keypoints in the corresponding images. The green circles represent each keypoint.



(a) Keypoints detected using the SIFT feature extraction method.

(b) Keypoints detected using the ORB feature extraction method.

Figure 1: Visualization of keypoints

1.2 Feature Matching

Feature matching is the process of finding corresponding feature points between two images. These feature points are distinctive regions or keypoints of an image that can be easily identified and described. Feature matching is a fundamental step in many computer vision tasks, such as object recognition, image stitching, and 3D reconstruction.

The provided code implements a feature matching function featurematching() that takes in keypoints and descriptors of two images and returns the corresponding matches and their source and destination keypoints. The function uses the specified descriptor method (SIFT, SURF, or ORB) to extract descriptors for each keypoint and then matches the descriptors using the k-NN algorithm. The matches are then refined by applying the ratio test and incrementally increasing the ratio until the minimum number of matches is reached. Finally, the matches are sorted by distance, and their corresponding source and destination keypoints are returned.

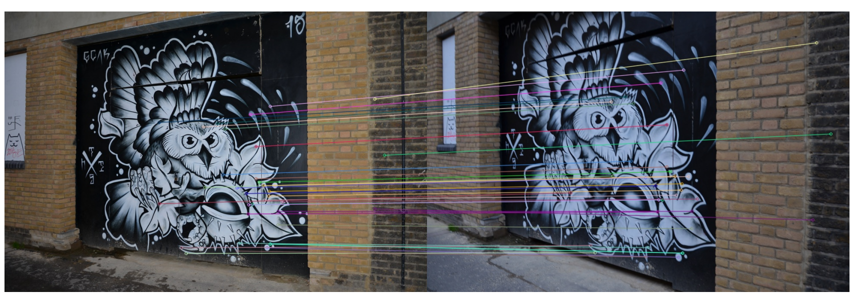


Figure 2: Example of feature matching result between two images using the implemented algorithm. The image shows correspondences between the source and destination keypoints.

1.3 Finding Homography

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography is a transformation matrix used in computer vision to map points from one image to another image of the same plane. To find the homography matrix, we use a set of corresponding points in the two images and estimate the matrix that best maps one set of points to the other. The RANSAC algorithm is a popular method for estimating the homography matrix from noisy data.

The findhomography function takes two sets of points as input and uses the RANSAC algorithm to estimate the homography matrix. The function computes the homography matrix using singular value decomposition (SVD) and computes the number of inliers for each estimate of the matrix. The function returns the homography matrix that produces the highest number of inliers.

To solve the system of linear equations that finds the homography matrix, we use SVD. The homography matrix is the right singular vector corresponding to the smallest singular value of the matrix. We can compute the SVD of the matrix using np.linalg.svd and extract the homography matrix from the right singular vector.

1.4 Merging by Transformation

This code segment uses a transformation matrix, H, and the coordinates of source points to merge two different images. In the first step, the x and y coordinates of the source points are taken to create the primary matrix. Then, the inverse of the primary matrix is calculated by multiplying it with the H matrix. As a result of these calculations, the x-coordinates are adjusted to be used for the merged image. Finally, the first and second images are adjusted according to the calculated x-coordinates and concatenated horizontally to create the panoramic image.

2 Part 2 - Results

In the second part of this report, we will present the results of our image stitching algorithm on the dataset. Specifically, we will show plots of feature points and matching lines for each pair of sub-images, as well as the constructed panorama image for each pair. Additionally, we will provide a table comparing the runtime and visual quality of the SIFT, SURF, and ORB feature description methods. Finally, we will provide a commentary on our results, including an explanation of our intermediate results and an assessment of the performance of our image stitching algorithm in creating visually appealing panoramas.

2.1 Comparision Methods

The ORB (Oriented FAST and Rotated BRIEF) feature descriptor is known for its high efficiency and speed compared to other feature description methods, such as SIFT and SURF. ORB uses the FAST algorithm for feature detection, which is an efficient corner detection algorithm, and the BRIEF descriptor, which is a binary descriptor that requires less memory and computation time than SIFT and SURF.

Table 1: Runtime Comparison of Feature Description Methods

Feature Description Method	**Elapsed Time (s)**
SIFT	40
SURF	60
ORB	13

2.2 Feature Matching İmages

The following images were extracted using the ORB (Oriented FAST and Rotated BRIEF) feature extractor. These images demonstrate a feature matching technique used to match objects across different photographs.

Figure 3: Bird

Figure 4: Boat

Figure 5: Circus

Figure 6: Weapon

Figure 7: Graffiti

Figure 8: Soldier

2.3 Orb Panoramic İmages

In this section, we present panoramic images created using the ORB feature extraction technique. ORB is a popular algorithm for detecting and describing key points in images, which makes it useful for creating high-quality panoramic images by stitching together multiple overlapping images.

Figure 9: Orb- Bird

Figure 10: Orb- Boat

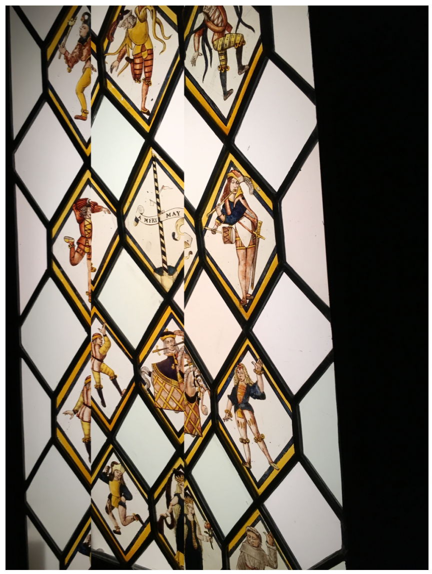


Figure 11: Orb- Circus

Figure 12: Orb- Weapon

Figure 13: Orb- Graffiti

Figure 14: Orb- Soldier

References

[1] https://web.cs.hacettepe.edu.tr/~nazli/courses/bbm416/index.html

[2] https://pyimagesearch.com/2016/01/11/opencv-panorama-stitching/

[3] https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html

[4] https://math.stackexchange.com/questions/494238/how-to-compute-homography-matrix-h-from-corresponding-po