# Introduction

The image stitching pipeline for constructing a panoramic image by merging two images is implemented in the cw2.py script. The OpenCV library stochastic and SciPy are used for this process of image processing and spatial counts respectively. It consists of detecting keypoints and matching them before calculating homography and blending the images into one seamless large angle panorama.

# Overview of Implementation

Here are the steps that are followed in stitching two images using cw2.py:

* Image Acquisition: Loading of input images.
* Feature Detection and Description: Detecting keypoints as well as computing their descriptors with SIFT (Scale-Invariant Feature Transform).
* Keypoint Matching: Descriptors between both images are matched in a brute-force manner.
* Homography Computation: Based on matched keypoints, the homography matrix that lines up both photos is computed.
* Image Warping and Stitching: Using this homography matrix to distort the second photo so it can be combined with the first one will yield a single image.
* Black Border Removal: This function crops off any dark edges resulting from distortions on the combined image.
* Result Saving: The final stitched picture is stored away as a file.

# Code Explanation

1. Importing Libraries:

* cv2: OpenCV library for image processing.
* numpy: Library for numerical operations, particularly useful for handling arrays.
* logging: Used for tracking events that happen during execution.

2. Reading images: This function reads two images and returns them as NumPy arrays.

3. SIFT (Scale Invariant Feature Transform) is used to detect keypoints and compute their descriptors in this step. It is resistant to changes in scale and rotation as such; therefore, it can be applied for panorama stitching.

4. Keypoint matching of features from two images is done using Euclidean distance by this function. A ratio test filters out poor matches, where only good matches are considered.

5. The homography matrix is computed by this function using matched keypoints, with at least four matches needed for reliable computation of homographies.

6. Image warping and stitching: This function warps the second image with the computed homography and stitches it together with the first image. The resultant image will have black borders because of the warping process.

7. Black borders removal: This function finds the bounding rectangle of non-black pixels then crops the image accordingly to remove black borders.

8. Main Function: Each of the above discussed processes are used in the ‘main’ function, from reading images to saving the final stitched image.

# Ending Remarks

The implemented panorama stitching application effectively combines two images into a single panoramic image using feature detection, matching, and homography computation. The choice of SIFT for feature detection ensures robustness against scale and rotation changes, while the ratio test for matching enhances the quality of matches. The final step of removing black borders ensures a clean output image.

Credits

1. Image credits for p1 and p2: <https://towardsdatascience.com/image-stitching-using-opencv-817779c86a83>