Panorama estimation using Image features and Homography

Daniel Amirtharaj
damirtha@buffalo.edu

1 Objective

To detect keypoints in 2 given images, match them, find the Homography matrix that transforms one image to the other image's plane, using these matches and warp the first image to the second image's plane.

2 Analysis

2.1 SIFT

SIFT, short for Scale Invariant Feature Transform, is a powerful tool for feature detection and matching in Computer Vision. As the name suggests it detects keypoints or distinctive points of interest that stand out in an image, gives them a 128-bit descriptor that encodes information regarding the keypoint's orientation (that can be guaranteed to be unique for a keypoint), ensuring that they can be matched between different images that have the same keypoints with different scales, intensities or orientations.

2.2 k-nearest neighbours algorithm

Once features are detected, they must be matched with corresponding features in other images, in order to perform advanced tasks such as computing disparity maps or stitching images. This can be achieved using k-nearest neighbours with k=2. This maps one feature in one image to its closest match in the other image based on a distance measurement of the respective keypoint descriptors. The matched feature pairs can then be filtered based on their distance measures to get fewer matched pairs that are better matches.

In this project the brute force matcher is used. It works by picking a descriptor from one image, calculating the distance measure (L2 norm can be used with SIFT), with every keypoint descriptor in the other image and returning the nearest one.

2.3 RANSAC

RAndom SAmple Consensus, is a powerful algorithm that can effectively remove outliers during parameter estimation, done while modelling a dataset. It can be applied with any parameter estimation problem, and hence is used widely to give robust fits while modelling a given dataset. It works by applying an iteravtive algorithm which selects random samples from the dataset and fitting a model to these samples, and seeing if they give the best fit. It returns the best fit, which leaves out outliers. It can be tuned and its threshold changed to determine what is decided to be an outlier and an inlier.

2.4 Homography

Homography is a transformation that transforms pixels in one image plane to another. The matrix that describes this transformation is known as the homography matrix. This is useful when the image needs to be viewed from the perspective of another observer, to whose image plane this image needs to be mapped. This is useful in stereo vision, where in order to find disparity for depth, two features in the image pair must be matched.

When an image is transformed to another image's plane, it ensures that the epilines in both images are parallel to this image plane. This is a useful property that can be harnessed to reduce the number of searches while matching keypoints in 2 images, since a given point in an image will lie on the corresponding epiline in the other image (observed from a different point in space).

As defined in OpenCV documentations, Homography can be mathematically expressed as,

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

3 Method

The following steps were followed to find the Homography and the perspective warp of image1 to image2.

1. SIFT was used to compute keypoints and keypoint descriptors in image1 and image2. Keypoints in both images were captured.

2. Using the keypoint descriptors, good matches with distance lesser than 0.75 were filtered. Matching was done using brute force k-NN matcher with k=2 and L2 norm distances.

3. Good matches obtained were used to find the Homography matrix, using RANSAC with the findHomography() function. 10 inlier pairs were randomly chosen out of this and plotted.

4. Corner points of image1 transformed to image2's plane were computed, to estimate the size of the output warp of image1 to image2's plane, as well as the translation necessary to get all pixels of image1 when warped to image2's plane (since some pixels ended up getting -ve locations initially).

5. The translation vector was computed by taking the coordinate of the left most and top most points in the warped image (if they had negative pixel locations). Using a matrix multiplication of this translation vector and the Homography matrix, the warped image of image1 to image2's plane was obtained.

6. The size of the output image was decided by comparing and taking max of the size of the warped image and size of image obtained by translating image2 by the translation vector.

7. The final image was formed by super-imposing image2 onto the warped image, where the origin of image2 was translated by the translation vector obtained in the previous step.

Source code for this implementation is included at the end of this report.

4 Results

The Homography matrix for the given images was calculated using RANSAC with projection error threshold 3 and floating point representations of pixel coordinates, $H = \begin{bmatrix} 1.588 & -2.914e\text{-}1 & -3.956e{+}2 \\ 4.453e\text{-}1 & 1.438 & -1.906e{+}2 \\ 1.196e\text{-}3 & -3.753e\text{-}5 & 1 \end{bmatrix}$

(a) SIFT keypoints detected in image1.

(b) SIFT keypoints detected in image2.

Figure 1: SIFT keypoints detected.

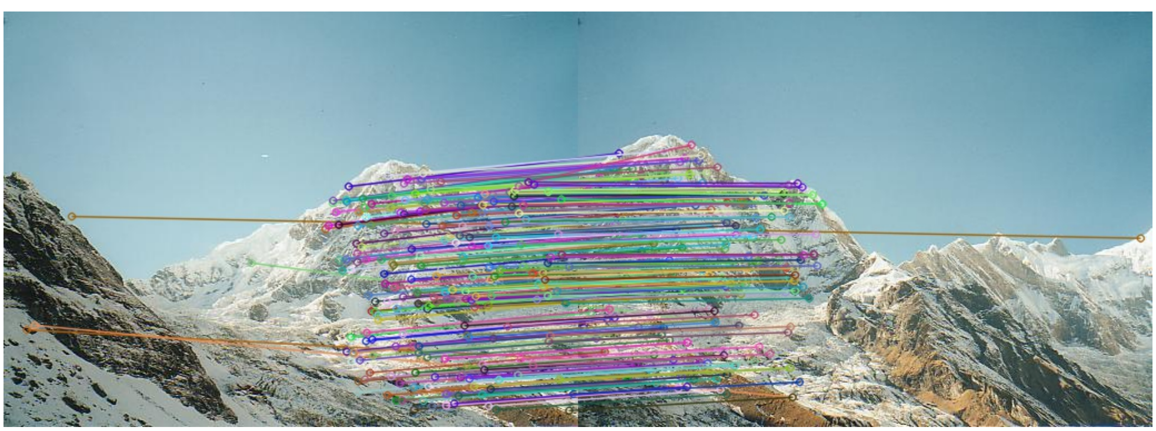

Figure 2: Matches detected with k-NN, with distance lesser than 0.75 between pairs.

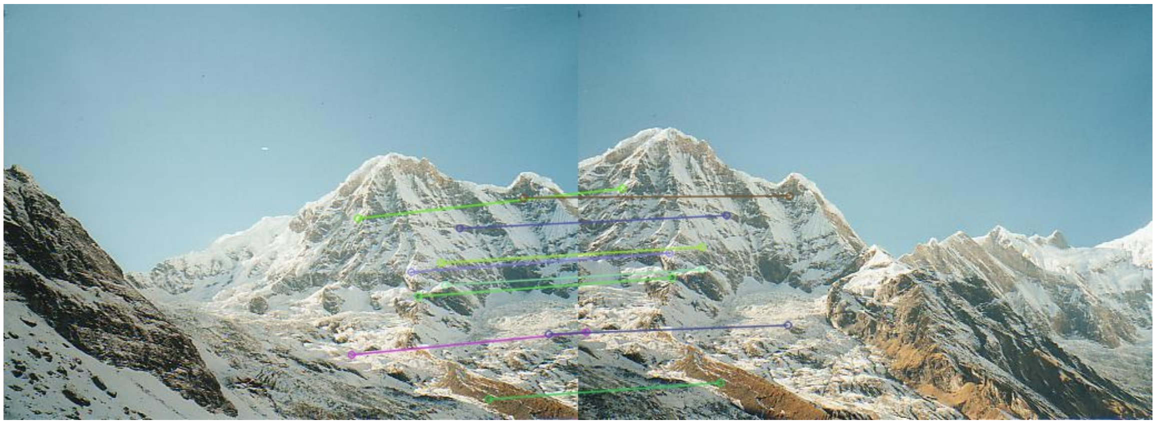

Figure 3: 10 Inlier matched pairs after applying RANSAC.

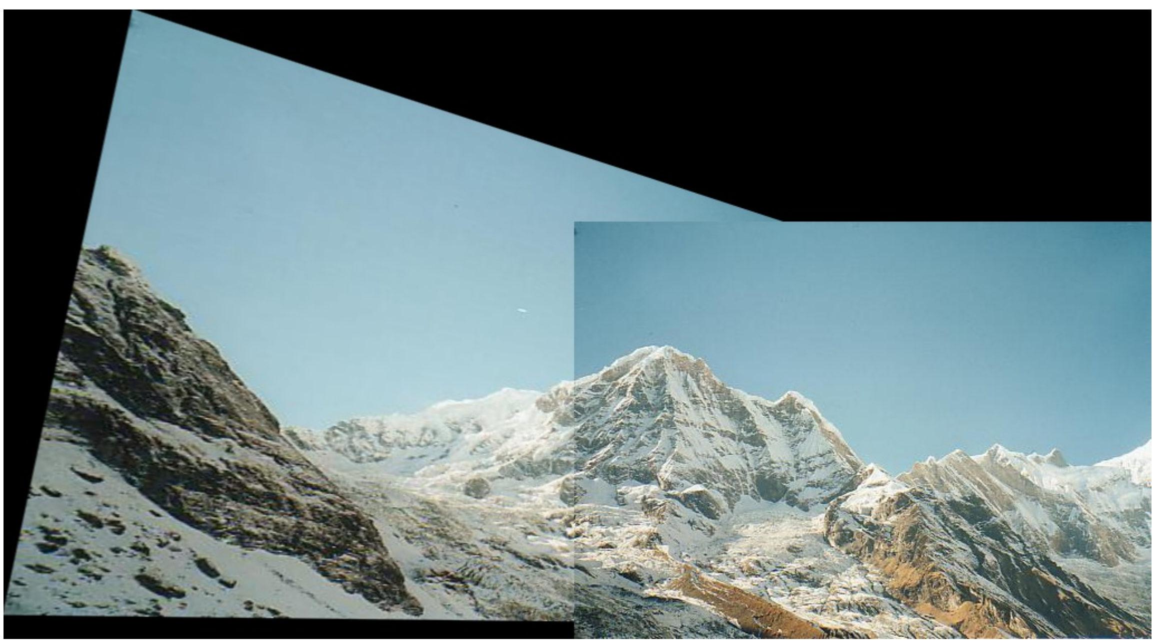

Figure 4: Image1 warped using the Homography matrix to image2's plane, stitched with image2, aligned based on their pixel values.