**IMAGE REGISTRATION**

**AIM:**

To deform or warp the test images according to the original image considering it as the reference image.

Language: Python3

Libraries: opencv, numpy

**ALGORITHM:**

1. Import original image
2. Import test image
3. Convert to grey scale
4. Initiate ORB Detector
5. Find key points and descriptors
6. Match key points using Brute force Hamming
7. Use RANSAC to eliminate bad points
8. Register images using homology

**WORKING:**

The approach used here is feature based image registration which detects sparse set of features in one image and match it with the features in other image. A transformation is calculated based on the matched features and the test images are warped to original images. Image registration warps test images to original images, so that the features in both the images line-up perfectly.

**CODE EXPLANATION:**

Images are converted to grayscale to get most of the features and orb is initiated. ORB is abbreviated as Oriented Fast and Rotated Brief. ORB is used to detect the features of the image. ORB\_create() specifies the number of features to be detected. detectAndCompute(img) returns the key points of the features detected and its associated descriptors where the keypoints and descriptors are a bunch of list. A descriptor matcher is created to use Brute force Hamming. Brute force matcher takes the descriptors of one set and matches it with all other features in the second set using match(). Match() returns the matched features. Matched features returned by the matcher is stored in matches and the list is sorted based on the distance between the features.To apply homography good points are detected using RANSAC which stands for Random Sample Consensus. RANSAC detects and eliminates the outliers and provides with the inliers as the good points. To use RANSAC along with findHomography() we need to access the keypoints through their index. Key points are stored in the list created by unwrapping the key points returned by detectAndCompute(). Those points are used to find homography between the images which returns a 3\*3 homography matrix. The points are used to register the test images with respect to the original image.

Warped Images are stored in the name RegisteredImage1, RegisteredImage2, and so on.