INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

DIGITAL IMAGE PROCESSING LABORATORY

A REPORT ON

Mini Project

Automatic Image Registration

Parakh Agarwal
17EC35016

Anand Jhunjhunwala
17EC35032

14.04.2021

Group No. 2

DEPT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

VISUAL INFORMATION AND EMBEDDED SYSTEMS

Contents

1 Objective 2
2 Introduction 2
3 Scale-invariant function transform (SIFT) 3
4 Affine Transformation 6
5 Implementation 7
5.1 Feature matching 8
5.2 Ratio test 8
5.3 Outlier Removal 8
5.4 Estimation of Affine Matrix 9
5.5 Final Registration 9
6 Results 10
6.1 Result - I 10
6.2 Result - II 11
6.3 Result - III 12
6.4 Result - IV 13
7 Discussions and Scopes of Improvement 14
8 Conclusion 14

1 Objective

Implement Coarse-to-Fine Scheme for Automatic Image Registration Based on SIFT and Mutual Information using the reference ”A novel coarse-to-fine scheme for automatic image registration based on SIFT and mutual information.”[15] leaving the optimisation step.
Test the performance from the test images of the reference publications mentioned.

2 Introduction

Image Registration is a critical approach to determine the most accurate match between two photographs of the same scene taken at different times, with different sensors, and from different viewpoints[1]. The reference and sensed images are geometrically aligned during this process. Computer vision, pattern matching, medical image analysis, and remote sensing image processing are just some of the areas where image registration is used. It is critical to achieve an accurate match and obtain precise geometric correction in remote sensing applications, in particular. The ability to achieve subpixel precision is critical for applications like image fusion and shift detection. For instance, to achieve a change detection error of less than 10%, a registration accuracy of less than 0.2 of a pixel is needed[2]. The use of automatic image registration is a realistic way to accomplish this aim. Despite the fact that a number of algorithms have recently been proposed[3][4][5][6], automatic image registration remains a challenge due to the existence of unique challenges in the field of remote sensing. Geometric deformations (translation effect, rotation and scale distortion, occlusion, and viewpoint difference) and radiometric differences are the key challenges (illumination change and sensor and spectral content difference). As a result, further research studies are required to enhance the efficiency of current registration methods. Intensity[1] and feature-based[7] methods are the two broad types of image registration methods. Feature-based methods extract important features before matching them with similarity measures to determine geometric correspondence between two pictures.

These methods are fast and resilient to noises, complex geometric distortions, and large radiometric variations, which is one of their key advantages. However, they only work well if the right features are extracted and matched by reliable algorithms. Point, edge, contour, and

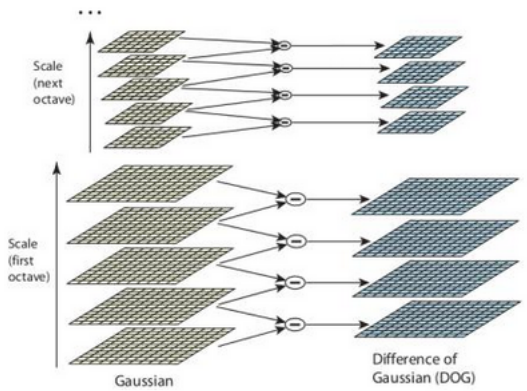
area are some of the most widely used features, and feature matching methods include invariant descriptor, spatial reference, and relaxation methods. For affine transformation, scale shift, rotation, image blur, jpeg compression, and illumination change, Mikolajczyk et al.[8] compared the output of descriptors. For the most part, they discovered that the scale-invariant function transform (SIFT)[9] works best.

SIFT is capable of extracting invariant features from images and can be used to perform accurate matching through a wide range of affine distortion, shift in 3-D perspective, noise addition, and illumination change. Despite SIFT's appealing advantages, it has some drawbacks when applied directly to remote sensing images, such as the number of detected feature matches being limited and their distribution being unequal due to the dynamic content design of remote sensing images[10]. Furthermore, several feature matches include outliers due to substantial variations in image intensity between overlay regions of remote sensing images. As evidenced by recent algorithms applying SIFT to remote sensing images, using SIFT alone cannot yield optimal results.

3 Scale-invariant function transform (SIFT)

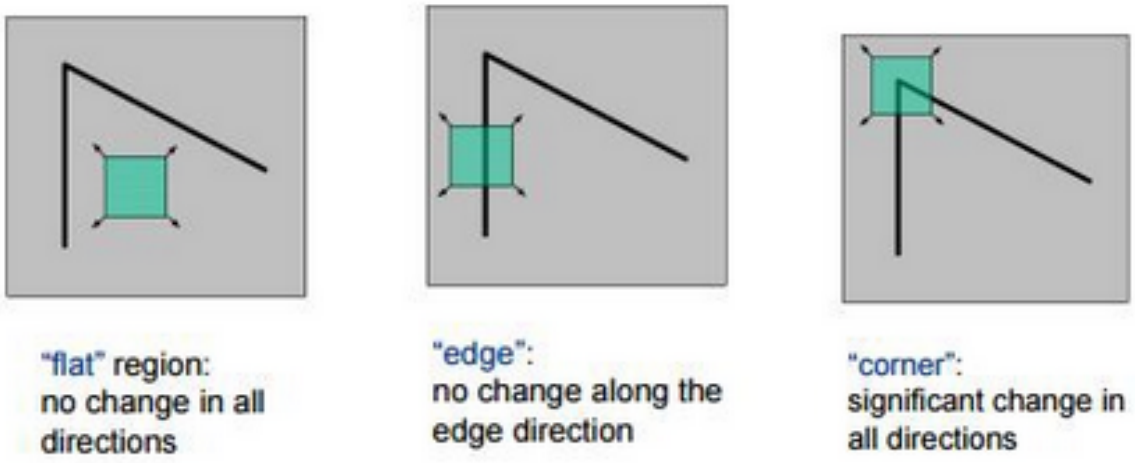
In his paper, Distinctive Image Features from Scale-Invariant Keypoints[11], published in 2004, D.Lowe of the University of British Columbia proposed a new algorithm called Scale Invariant Feature Transform (SIFT), which extracts keypoints and computes their descriptors. The SIFT algorithm consists primarily of four stages. We'll go through them one by one.

1. **Scale-space Extrema Detection**
   Since LoG[12] is an expensive method for detecting corners, the SIFT algorithm uses Difference of Gaussians[13], which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different sigma, let it be sigma and k*sigma. The difference of Gaussian blurring of a picture with two different sigma, let's call them sigma and k*sigma, is obtained. In the Gaussian Pyramid, this process is repeated for different octaves of the picture. After locating this DoG, photos are



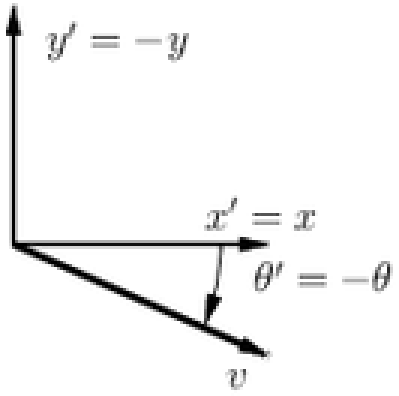
scanned for local extrema through scale and space. One pixel in an image, for example, is compared to its eight neighbors, as well as 9 pixels in the next scale and 9 pixels in previous scales. It's a possible keypoint if it's a nearby extrema. It essentially means that the keypoint is best represented on that scale.

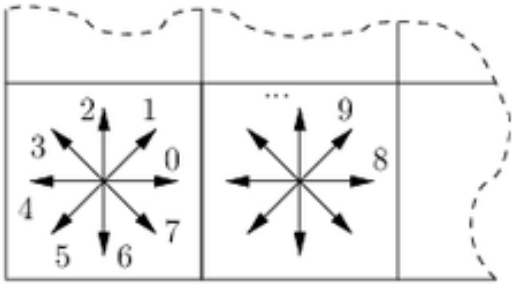
2. **Keypoint Localisation**
   Once possible keypoint locations have been identified, they must be optimized in order to obtain more reliable results. They used a Taylor series expansion of scale space to get a more precise position of extrema, and if the amplitude at these extrema is less than a threshold value (0.03, according to the paper), it is rejected. Edges have a higher response in DoG, so they must be eliminated as well. A concept similar to the Harris corner detector is used for this. The principal curvature was calculated using a 2x2 Hessian matrix (H). One eigenvalue is greater than the other for corners, as we know from the Harris corner detector. They evaluate the ratio of the two eigenvalues to see if it crosses an edgeThreshold in OpenCV, and if it does, the keypoint is discarded. As a result, low-contrast keypoints and edge keypoints are removed, leaving only strong interest points.



3. **Orientation Assignment**
   To achieve image rotation invariance, each keypoint now has an orientation assigned to it. Depending on the scale, a neighborhood is drawn around the keypoint spot, and the gradient magnitude and direction are determined in that region. The result is a 360-degree orientation histogram of 36 bins. (It's weighed by gradient magnitude and a gaussian-weighted circular window with a sigma of 1.5 times the keypoint's scale.)

   To measure the orientation, the highest peak in the histogram is used, as well as any peak above 80% of it. It produces key points that are the same size and position, but face in different directions. It helps to keep the matching stable.

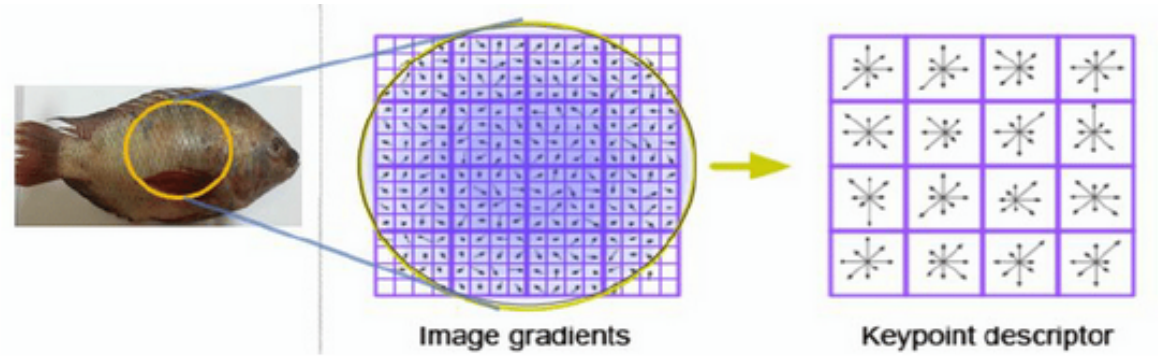




bin indexes when stacked

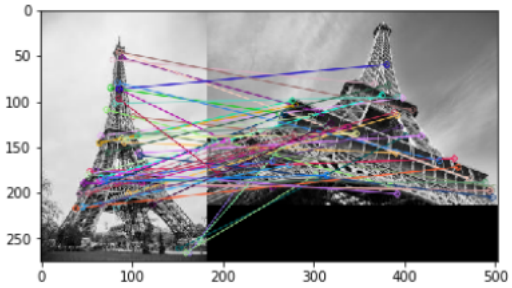
4. **Keypoint Descriptor**

   A keypoint descriptor has now been developed. A 16x16 neighborhood is drawn around the keypoint. It's broken down into 16 4x4 sub-blocks. An 8-bin orientation histogram is generated for each sub-block. As a result, there are a total of 128 bin values available. To shape a keypoint descriptor, it is represented as a vector. In addition, several steps are taken to ensure robustness against changes in illumination, rotation, and other factors.



5. **Keypoint Matching**

   The closest neighbors of keypoints in two images are identified and matched. However,



the second closest match can be very similar to the first in some situations. It may happen as a result of noise or for other reasons. The closest-distance to second-closest-distance

ratio is used in this situation. They are refused if it is greater than 0.8. According to the paper, it eliminates about 90% of false matches while discarding just 5% of right matches.

4 Affine Transformation

Any transformation that retains collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and distance ratios (e.g., the midpoint of a line segment remains the midpoint after transformation) is called an affine transformation[14]. Below are some types of affine transforms shown below.

Identity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Reflection

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear-X

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \lambda_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear-Y

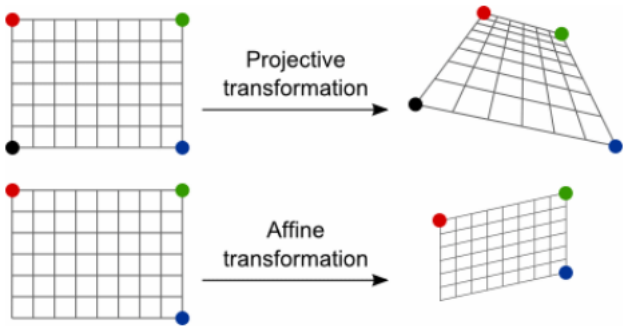
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \lambda_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

An affine transformation is also called an affinity. In general, an affine transformation is a composition of rotations, translations, dilations, and shears. While an affine transformation preserves proportions on lines, it does not necessarily preserve angles or lengths. Any triangle can be transformed into any other by an affine transformation, so all triangles are affine and, in this sense, affine is a generalization of congruent and similar.

In order to incorporate the idea that both the basis and the origin can change, we augment the linear space u,v with an origin t. Note that while u and v are the basis vectors, the origin t is a point. We call u,v, and t (basis and origin) a frame for an affine space. Thus, we represent a change of frame as:

$$p' = x.u + y.v + t \tag{1}$$

his change of frame is also known as an affine transformation. To represent transformations



among affine frames, we can loft the problem up into 3-space, adding a third component to every point:

$$\begin{aligned}
\mathbf{p}' &= \mathbf{M}\mathbf{p} \\
&= \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= x \cdot \mathbf{u} + y \cdot \mathbf{v} + 1 \cdot \mathbf{t}
\end{aligned}$$

5 Implementation

In this section, we tried to explain our implementation steps used in this project. We used python for our implementation. Moreover we used python-opencv library to implement the required function.

Note: Make sure opencv-python version is at least: **3.4.2.17**, and opencv-contrib-python version is at least: **3.4.2.17**.

5.1 Feature matching

Given the reference(destination) and sensed image(source) we need to find the corresponding matching points in both.

For doing so, we used SIFT features for matching.
We first find the features descriptors of reference and sensed image using sift.detectAndCompute() function of opencv.

Once we have these features, we use brute force matching to find the matching features points. For doing so we used knnMatch() function of cv.BFMatcher() which will give us the 2 nearest match of each feature point of reference w.r.t sensed image.

5.2 Ratio test

Once we have these matching points, we can't use it directly for estimation of the affine matrix, as it may contain many false matches.

For removing faulty matches and to select only good matches among all, we perform ratio tests, i.e we find the ratio of distance of each best match with distance of second best match, the reason for this is very simple, if the match is good then this ratio will be very less and if this match will be faulty then this ratio will be close to 1.

So, using the ratio test we only accept those points whose ratio is less than 0.6 and reject all the rest.

5.3 Outlier Removal

Even after the removal of false initial matches through a ratio threshold, some incorrect matches may still exist. Therefore, a reliable outlier removal process is necessary. For doing so we perform the outlier removal process mentioned in [paper].

Suppose after the ratio threshold we find m matched keypoints $\{R_i\}$ and $\{S_i\}$ belonging to the reference and sensed images, respectively, let R_iR_j be the distance between keypoints R_i and R_j and S_iS_j be the distance between keypoints S_i and S_j. A distance ratio D_{ij} is defined as:

$$D_{ij} = \frac{R_iR_j}{S_iS_j} \tag{2}$$

This distance ratio is computed based on all $\frac{m(m-1)}{2}$ possible combinations, and the number of D_{ij} in the intervals is counted in a statistical way, forming a scale histogram.

The denser cluster in the scale histogram corresponds to the true scale difference between

the images. The keypoint pairs that contribute to the cluster are accepted as correct matches, while the ones that are scattered and away from the cluster are considered as incorrect matches and are then eliminated, given remaining points are at least 3.

5.4 Estimation of Affine Matrix

Once we have performed the above outlier removal process, we are left with only good matches, then we use these matches to estimate the Affine Matrix using the least square method. We used cv.estimateAffinePartial2D() to find the Affine Matrix.

5.5 Final Registration

Now we have our affine matrix, the last step is to perform the affine transform of the sensed image to align it with the reference image and get the final result.
We does so by following steps:

1. We find the minX, minY, maxX, maxY values of the sensed image after affine transform to determine how much padding is required in the reference image to align the both images.

2. Once we have this minX, minY, maxX, maxY we calculate the new size of the reference image.

3. Now if any of the minX or minY is negative, we need to translate the image to make them positive, so we do this and modify our affine-matrix so that after applying affine transform the sensed image doesn't get clipped.

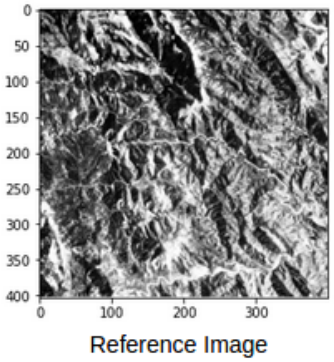
4. Now we just copy our reference image at the correct location depending upon the above adjustment, and perform an affine transform of the sensed image, with a new affine matrix, for doing so we used cv.warpPerspective() function of opencv.

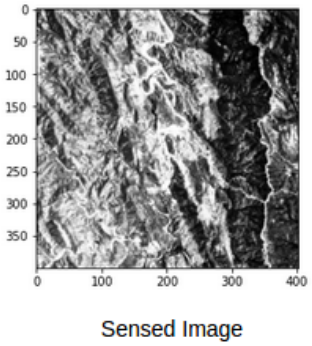
5. At last we just superimpose both the new padded reference image and affine transformed sensed image to make the final registered image.

6 Results

We here show the results of our implementation on images present in the paper.

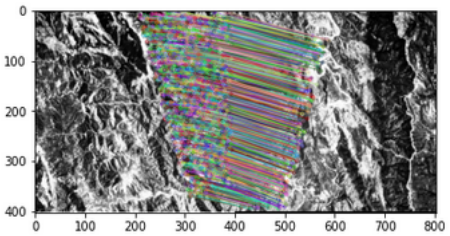
6.1 Result - I





Total Matches found without ratio test and outlier removal: 5148
Total Matches found after ratio test(0.6) and outlier removal: 1193

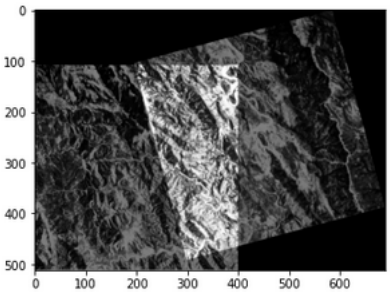
Feature Matching Output:



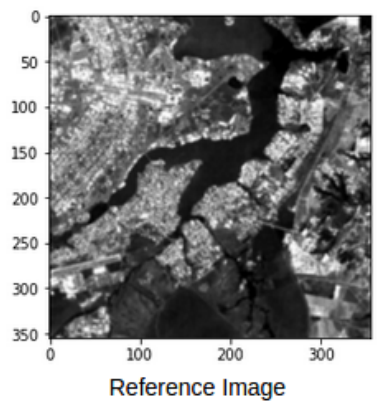
Estimated Affine Matrix:

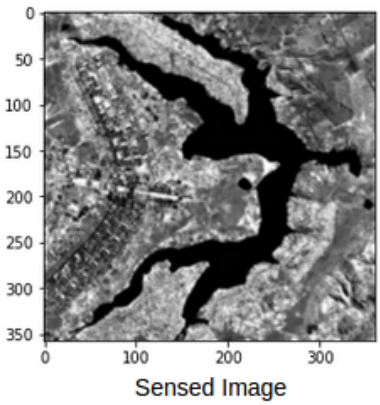
$$\begin{bmatrix} 0.96676143 & 0.25913406 & 195.58716828 \\ -0.25913406 & 0.96676143 & -4.23483726 \\ 0 & 0 & 1 \end{bmatrix}$$

Final Image Registration:



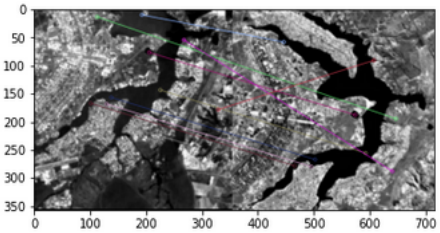
6.2 Result - II





Total Matches found without ratio test and outlier removal: 3039
Total Matches found after ratio test(0.6): 9 and outlier removal: 5

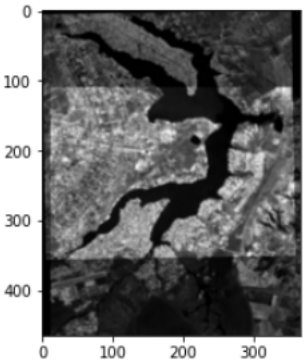
Feature Matching Output:



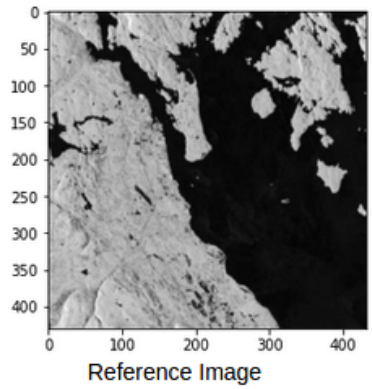
Estimated Affine Matrix:

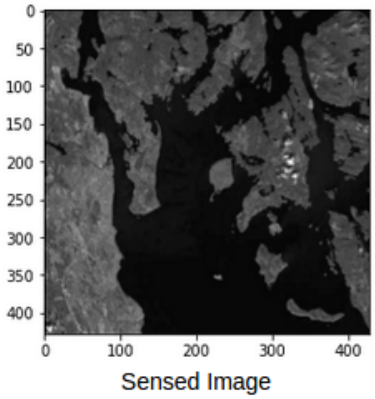
$$\begin{bmatrix} 0.981899707 & 0.016302754 & -12.0709003 \\ -0.016302754 & 0.981899707 & -103.912759 \\ 0 & 0 & 1 \end{bmatrix}$$

Final Image Registration:



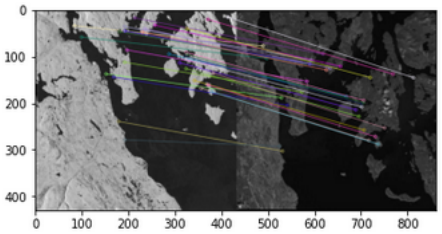
6.3 Result - III





Total Matches found without ratio test and outlier removal: 1733
Total Matches found after ratio test(0.6): 33 and outlier removal: 33

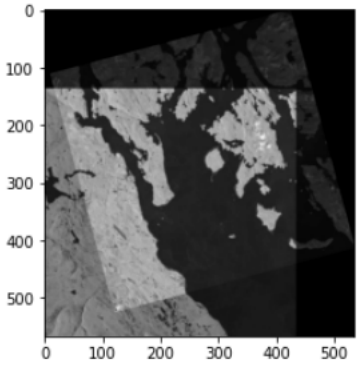
Feature Matching Output:



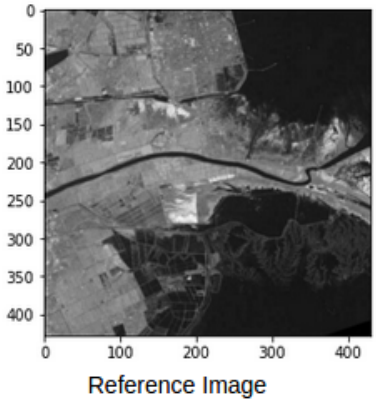
Estimated Affine Matrix:

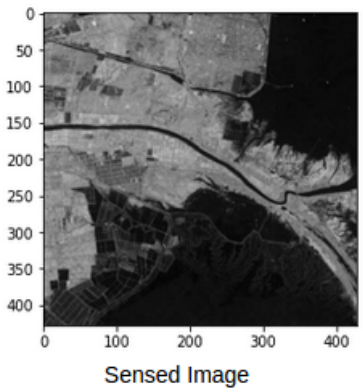
$$\begin{bmatrix} 0.96739352 & 0.26075663 & 9.68143235 \\ -0.26075663 & 0.96739352 & -25.60950674 \\ 0 & 0 & 1 \end{bmatrix}$$

Final Image Registration:



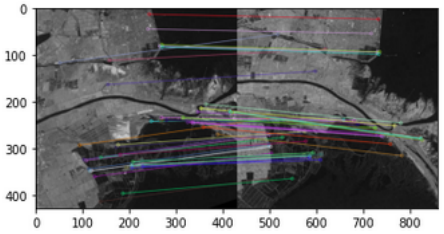
6.4 Result - IV





Total Matches found without ratio test and outlier removal: 1385
Total Matches found after ratio test(0.6): 47 and outlier removal: 47

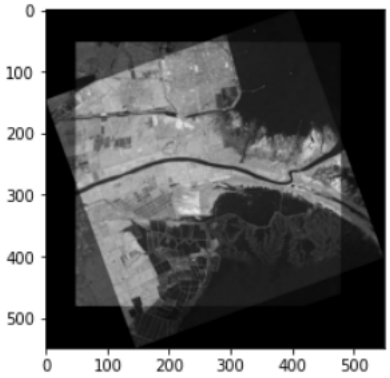
Feature Matching Output:



Estimated Affine Matrix:

$$\begin{bmatrix} 0.93997673 & 0.34172495 & -48.73525589 \\ -0.34172495 & 0.93997673 & 93.74544735 \\ 0 & 0 & 1 \end{bmatrix}$$

Final Image Registration:



7 Discussions and Scopes of Improvement

1. SIFT has some drawbacks when applied directly to remote sensing images, such as the number of detected feature matches being limited and their distribution being unequal due to the dynamic content design of remote sensing images.

2. KNN algorithm is used to find the most related matches between the detected matches using the SIFT algorithm.

3. The feature matches for SIFT include outliers due to substantial variations in image intensity between overlay regions of remote sensing images due to which only SIFT cannot give reliable results.

4. Affine transformation is a composition of rotations, translations, dilations, and shears thus it preserves proportions on lines.

5. While using the modified outlier removal, it is possible that one point on an image might match to multiple on the other image so it is necessary to check it and remove them.

6. Before registering the image it is important to pad the reference image so that when we overalp the transformed sensed image, final image don't get clipped.

7. For further improvement we can fine tune the found affine matrix by maximizing the mutual information between the reference and transformed sensed image. In the paper it is achieved by using Parzen histogram to estimate joint discrete histogram of images.

8. B-spline function is used for Parzen window.

9. However, the implementation of maximization of mutual information was not performed due to mainly 2 reasons, 1^{st} The paper didn't mention any optimal optimizer for performing the optimization, and 2^{nd} Implementation of all differential equations for performing optimization from scratch was out of scope of this project.

8 Conclusion

A full pipeline has been constructed to perform image registration based on SIFT feature detection and outlier removal. With the use of ratio test and outlier removal maximum of false matches had been removed. Even though we have not used the mutual information module, still the resulted registered image was comparable with the one obtained in paper. With the mutual information module the error of registration can be further reduced.

Bibliography

[1] B. Zitová and J. Flusser, "Image registration methods: A survey," Image Vis. Comput., vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[2] X. Dai and S. Khorram, "The effects of image misregistration on the accuracy of remotely sensed change detection," IEEE Trans. Geosci. Remote Sen., vol. 36, no. 5, pp. 1566–1577, Sep. 1998.

[3] H. Gonçalves, J. A. Gonçalves, and L. Corte-Real, "HAIRIS: A method for automatic image registration through histogram-based image segmentation," IEEE Trans. Image Process., vol. 20, no. 3, pp. 776–789, Mar. 2011.

[4] C. Xing and P. Qiu, "Intensity-based image registration by nonparametric local smoothing," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 10, pp. 2081–2092, Oct. 2011.

[5] M. Debella-Gilo and A. Kääb, "Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation," Remote Sens. Environ., vol. 115, no. 1, pp. 130–142, Jan. 2011.

[6] J. Liang, Z. Liao, S. Yang, and Y. Wang, "Image matching based on orientation-magnitude histograms and global consistency," Pattern Recognit., vol. 45, no. 10, pp. 3825–3833, Oct. 2012.

[7] L. G. Brown, "A survey of image registration techniques," ACM Comput. Surv., vol. 24, no. 4, pp. 325–376, Dec. 1992.

[8] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, Nov. 2004.

[10] A. Sedaghat, M. Mokhtarzade, and H. Ebadi, "Uniform robust scale invariant feature matching for optical remote sensing images," IEEE Trans. Geosci. Remote Sens., vol. 49, no. 11, pp. 4516–4527, Nov. 2011

[11] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004).

[12] Kong, Hui Cinar Akakin, Hatice Sarma, Sanjay. (2013). A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications. Cybernetics, IEEE Transactions on. 43. 1719-1733. 10.1109/TSMCB.2012.2228639.

[13] Assirati, Lucas Rosa, Núbia Berton, L. Lopes, Alneu Bruno, Odemir. Performing edge detection by Difference of Gaussians using q-Gaussian kernels. Journal of Physics Conference Series. 490. 10.1088/ 1742-6596/490/1/012020.

[14] George Bebis, Michael Georgiopoulos, Niels da Vitoria Lobo, Mubarak Shah, Learning affine transformations, Pattern Recognition, Volume 32, Issue 10, 1999, Pages 1783-1799, ISSN 0031-3203, https://doi.org/10.1016/S0031-3203(98)00178-2.

[15] M. Gong, S. Zhao, L. Jiao, D. Tian and S. Wang, "A Novel Coarse-to-Fine Scheme for Automatic Image Registration Based on SIFT and Mutual Information," in IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 7, pp. 4328-4338, July 2014, doi: 10.1109/TGRS.2013.2281391.