CS6780: Digital Video Processing
Assignment 2
Panorama Creation

Divya K Raman
EE15B085

20 April 2019

1 Introduction

This assignment deals with panorama creation using multiple images of a scene. In the first part, we implement the Lucas-Kanade (Direct) registration method for image alignment. We see that it works only for images which have very less error in initial alignment. We then implement the affine as well as the projective registration versions using a Gaussian pyramid. It is essential to use image pyramids. The algorithms fail to produce good results otherwise. The images in the provided dataset are hard to register due to large differences between consecutive images. We hence perform an initial alignment using Oriented FAST and rotated BRIEF (ORB) features and ransac upon which our algorithm is applied to refine the results. Images are warped as per the homographies determined and finally stitched together to obtain a panoramic view of the given scene. We present the implementation details, results and a detailed analysis in the following sections.

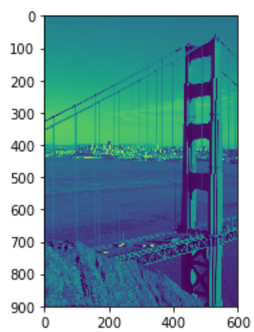
2 Lucas-Kanade(Direct) Registration

The Lucas Kanade method is a differential method for optical flow. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. We first calculate the gradients in x and y directions and determine the second order moment matrix. The matrix multiplication result of this matrix with a matrix equal to the dot product of difference in image intensities between the two images and the x and y gradients gives translation in x and y directions required for registering the two images. This method works only if the translation is below 1 pixel. For large motions, the method fails. Refer to LucasKanade.ipynb for the implementation.

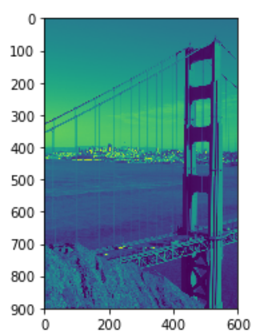
The images in the given dataset cannot be registered using Lucas Kanade as the motion is huge. We hence take 1 image, and translate it by x pixels(x varying from 0 to 1) horizontally and vertically and apply Lucas Kanade on the generated image pair.

Dataset 1: Golden gate

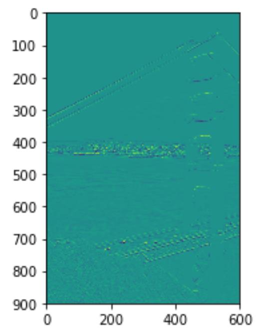
Golden image original image



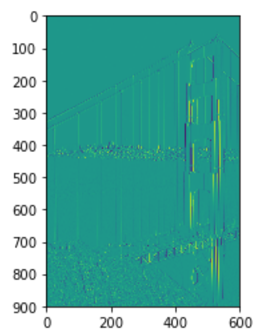
Golden image image translated by 1 pixel in both directions



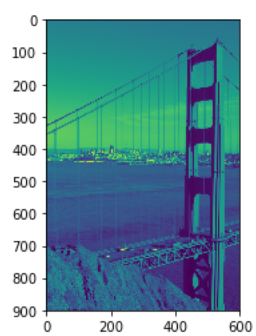
Horizontal gradients - horizontal edges



Vertical gradients - vertical edges



Transformation found by lucas kanade: deltax = 0.87, deltay = 0.95; transformed image:



We now estimate the transformations that are estimated for various values of translations. The results are summarised below.

x = 0.2 gives deltax = 0.2, deltay = 0.19

x = 0.4 gives deltax = 0.41, deltay = 0.4

x = 0.6 gives deltax = 0.58, deltay = 0.588

x = 0.8 gives deltax = 0.75, deltay = 0.78

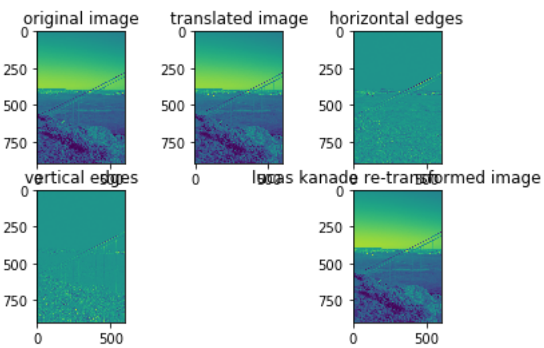
x = 1.0 gives deltax = 0.87, deltay = 0.95

x = 1.2 gives deltax = 0.81, deltay = 0.93

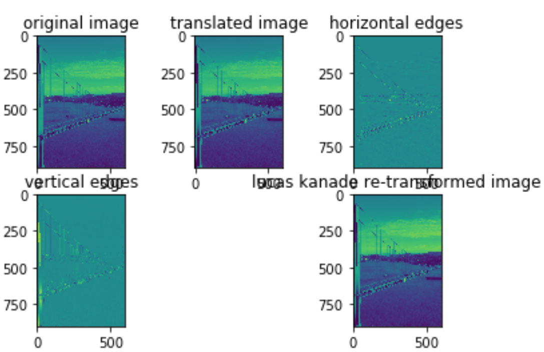
Clearly, the results are good for small values of x. As x increases, the error in the estimated transformation to register the two images also increases.

We now present results for a translation of 1 pixel on more images:

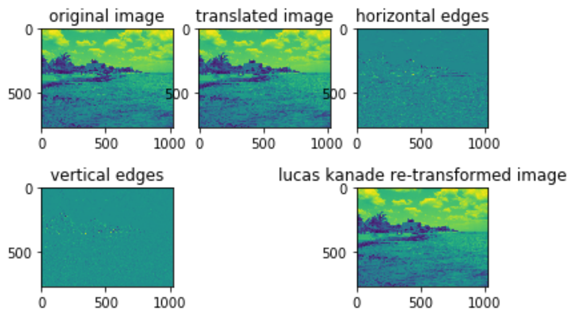
Golden gate 1: deltax=0.857, deltay=0.858



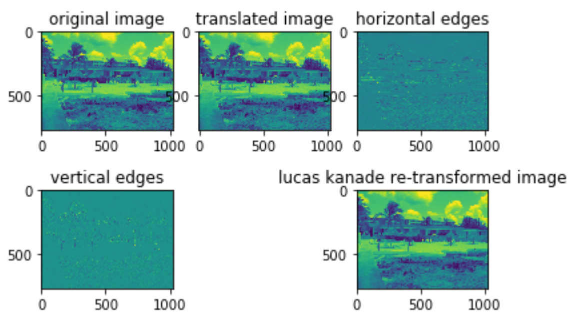
Golden gate 5: deltax=0.89, deltay=0.95



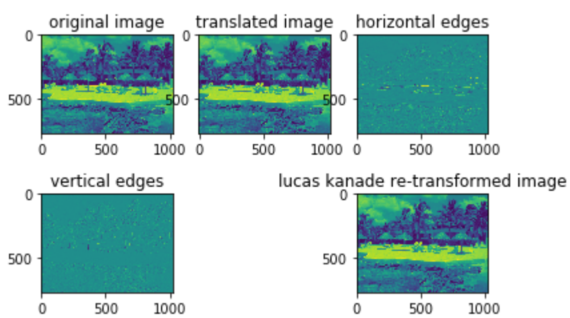
Hotel 1: deltax=0.95, deltay=0.84



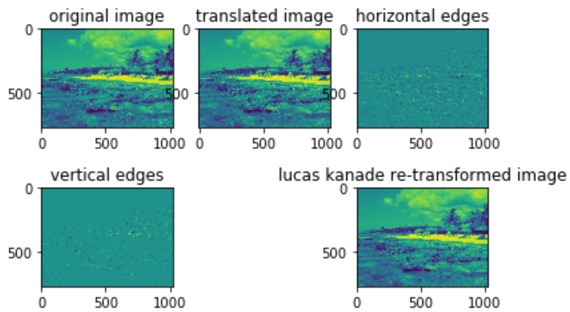
Hotel 3: deltax=0.93, deltay=0.88



Hotel 5: deltax=0.9, deltay=0.859



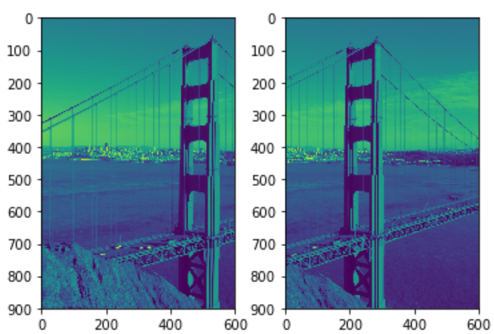
Hotel 7: deltax=0.91, deltay=0.82



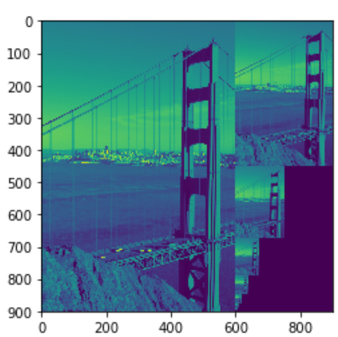
3 Affine/Projective Gaussian Pyramid method

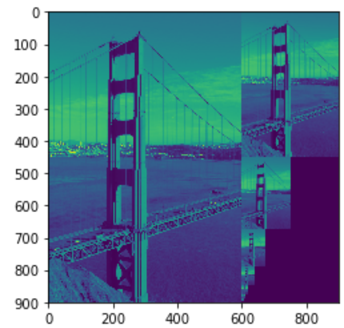
In this section, we will first walk you through the image registration process for a pair of golden gate images. We then demonstrate the results of our algorithm on other images. Refer to AffineScaleSpace.ipynb for implementation details.

Original Images



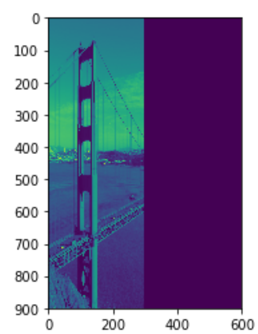
We now construct gaussian pyramids of both the images.





We then define a function incrementp. Given 2 images, it finds the homography between them. We see that this algorithm also works well only on images that have very less motion. To combat this problem, we find ORB features and use RANSAC to determine the homography between the 2 images. The projective transformation matrix is then initialised with this homography and the alignment is refined using the affine scale space algorithm. The homography is updated with the affine parameters, the images are warped and finally stitched together to get a panorama.

Registered(Transformation/Warping + Stitching) images:

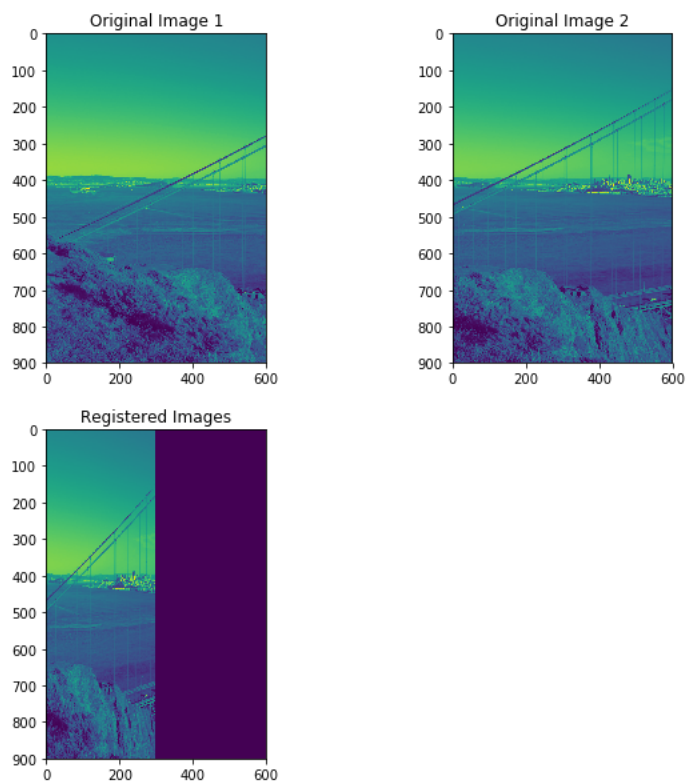


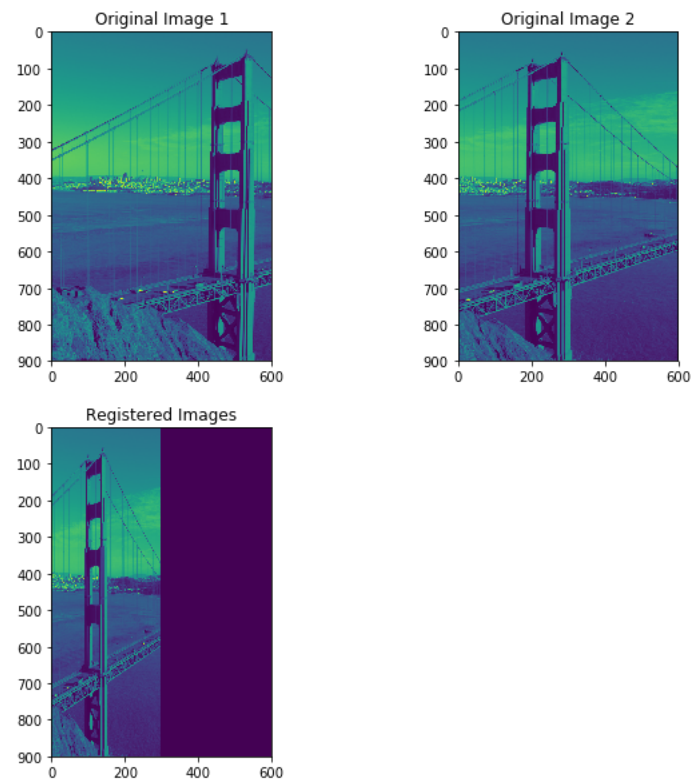
Why is scale space required?

Scale space helps in getting rid of some detail from the image. This improves results immensely. Also, while getting rid of these details, we need to ensure that we do not introduce new false details. The only way to do that is with the Gaussian Blur.

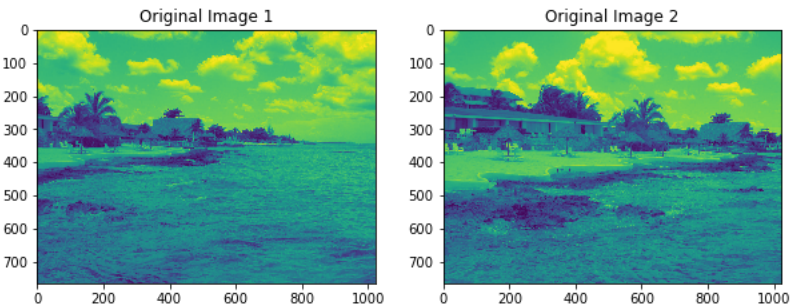
We now present more results.

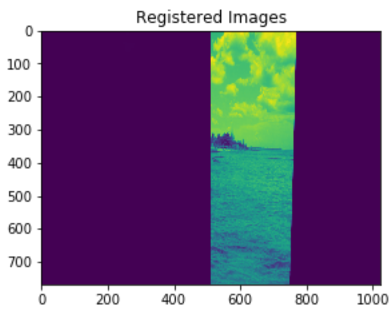
Golden gate dataset

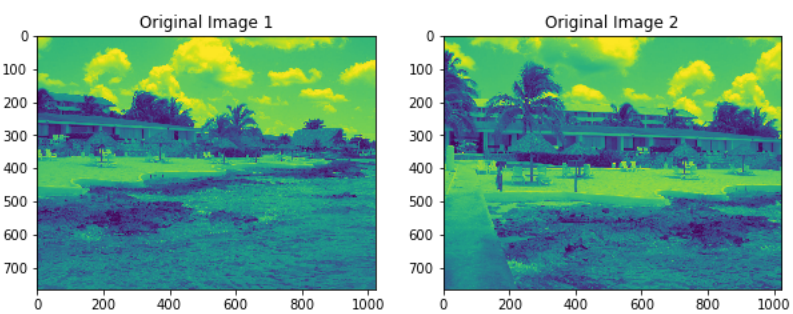


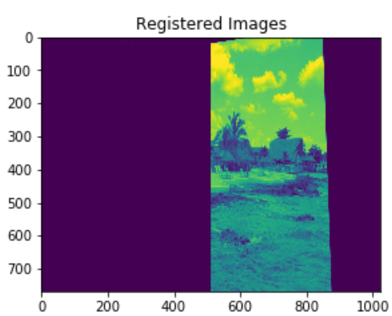


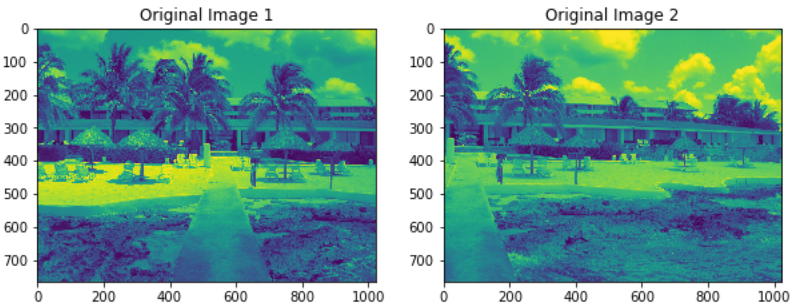
Hotel dataset

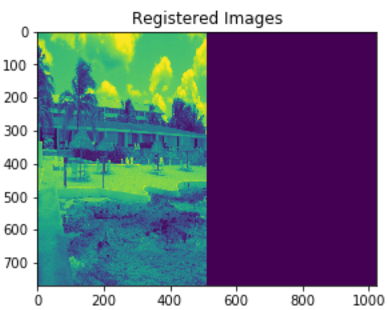

Original Image 1
Original Image 2
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

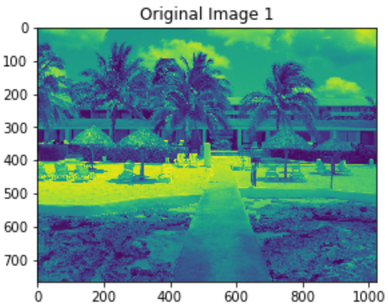

Registered Images
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

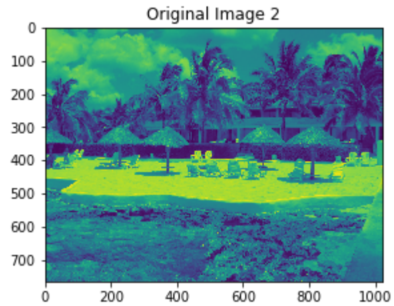

Original Image 1
Original Image 2
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

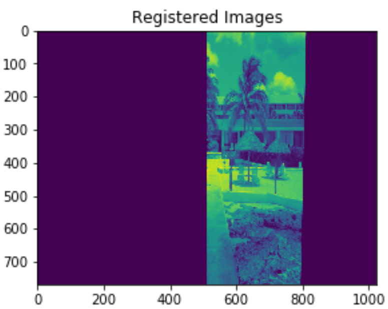

Registered Images
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

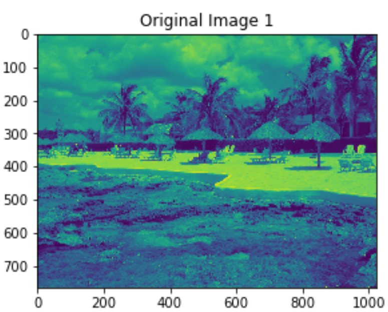

Original Image 1
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
Original Image 2
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

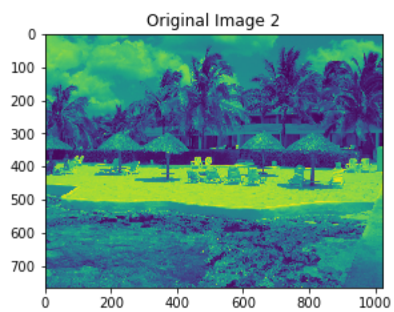

Registered Images
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

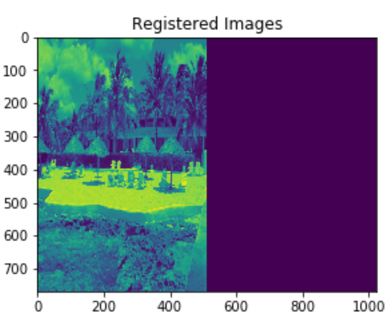

Original Image 1
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

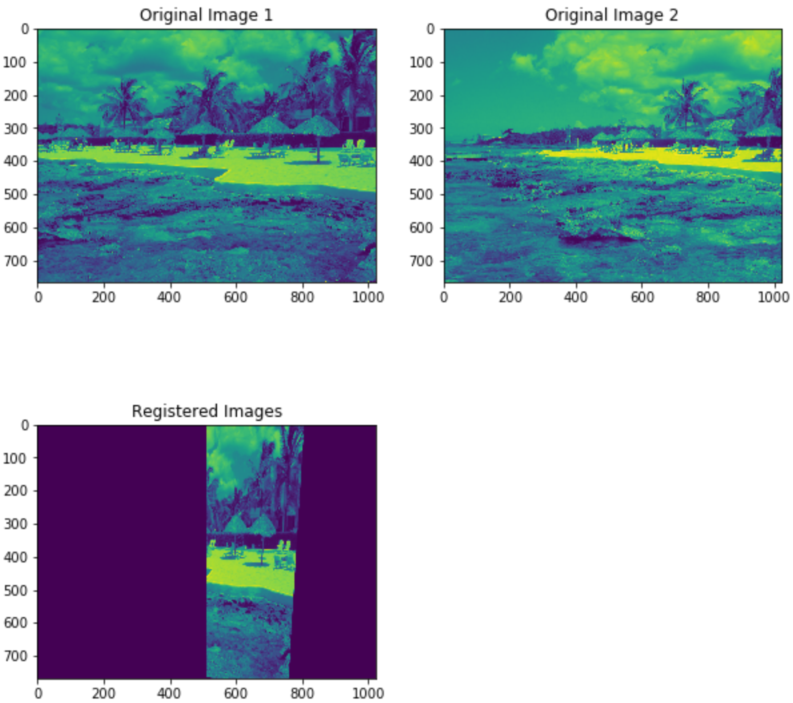

Original Image 2
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

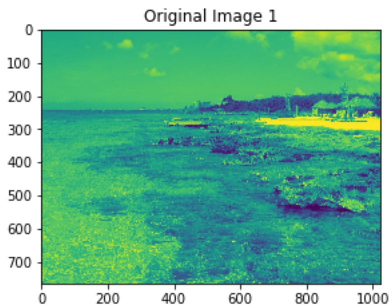

Registered Images
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

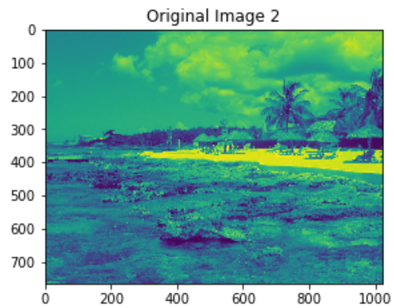

Original Image 1
0
100
200
300
400
500
600
700
0 200 400 600 800 1000

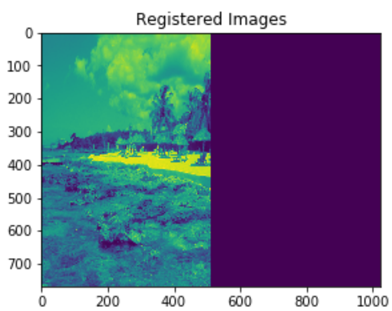

Original Image 2
0
100
200
300
400
500
600
700
0 200 400 600 800 1000


Registered Images
0
100
200
300
400
500
600
700
0 200 400 600 800 1000


Original Image 1
Original Image 2
Registered Images







CV Lab dataset

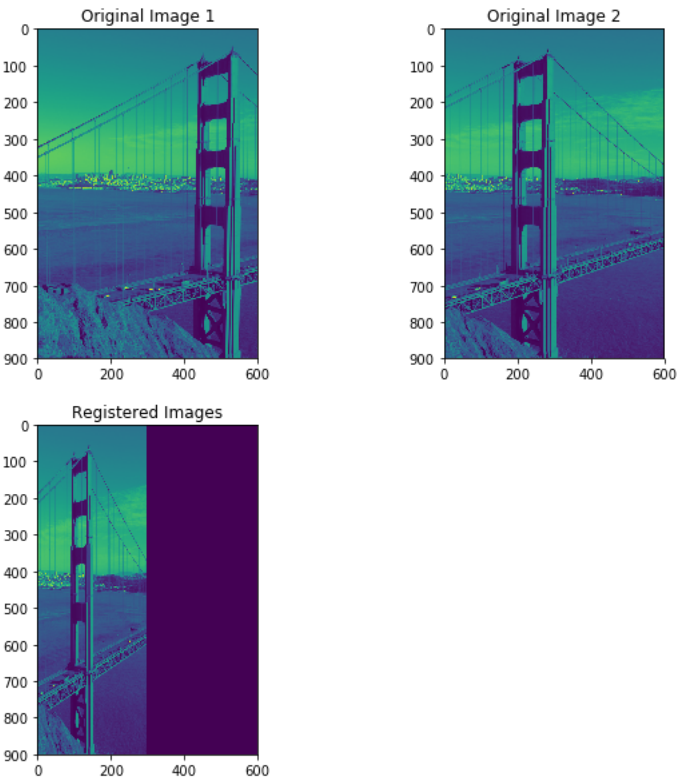
Results with cvlab dataset are not good as the difference between any two consecutive images is too high making it hard for optical flow to track.

Clearly, registration is best when the two images aren't too different from each other. The alignment error increases as the distance between the two images increases.

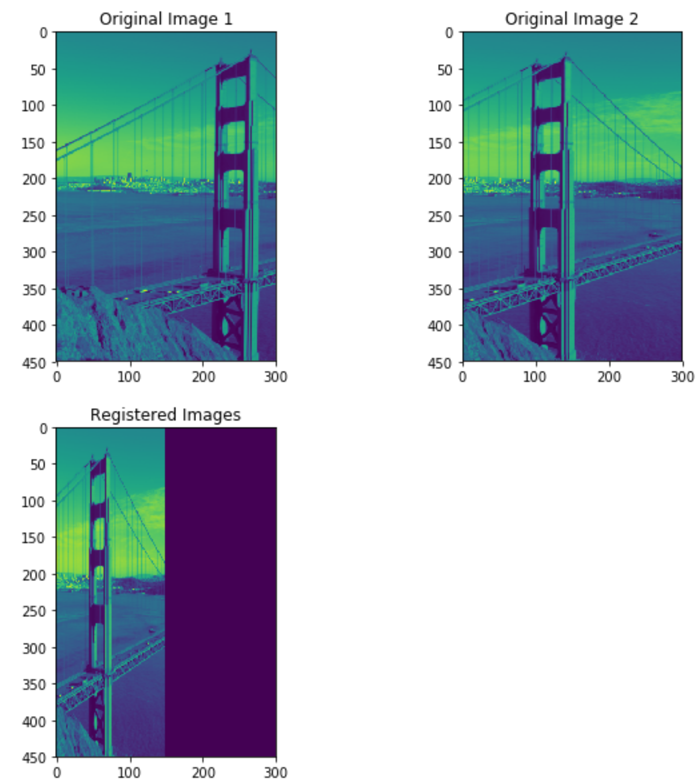
We now present a study on the effect of scale space in image registration.

Case 1:

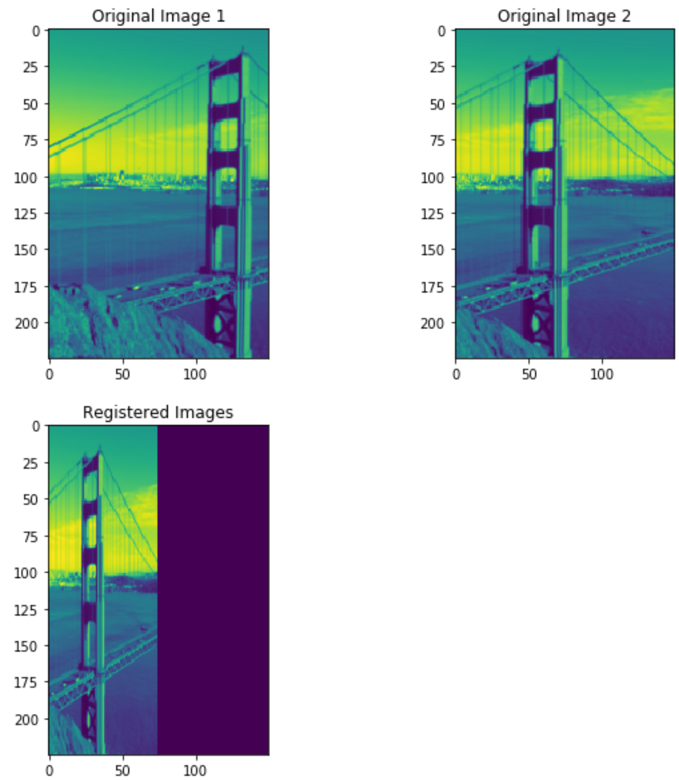
Original Images-pyramid level 0:


Original Image 1
Original Image 2
Registered Images
0
100
200
300
400
500
600
700
800
900
0
200
400
600

Level 1


Original Image 1
Original Image 2
Registered Images
0
50
100
150
200
250
300
350
400
450
0
100
200
300

Level 2



The images get smoother and the registration gets better as we go up the image pyramid. We are not able to register images beyond level 3 of the pyramid. This is because at this stage, the images are too smooth and RANSAC can't find sufficient matches causing it to throw an error.

4 Conclusion

In this assignment, we implemented and analysed various optical flow methods, namely, lucas kanade registration and affine scale space registration. We presented an analysis on various datasets and did a short study on the effect of gaussian pyramids in image registration.

5 References

1. https://en.wikipedia.org/wiki/Lucas–Kanade method
2. http://aishack.in/tutorials/sift-scale-invariant-feature-transform-scale-space/
3. https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/