University of Girona

Spain

Visual Perception

Lab 4 - Problem-Based Learning: Applications of Invariant Features

Submitted by :
Mr. Nayee Muddin Khan
Dousai
Mr. Anirudh Puligandla

1 Objective

The main objective of this lab is to verify the invariant features by using SIFT (Scale Invarient Feature Transform). SIFT is considered as well known algorithm in computer vision developed by David Lowe in 1999. The objectives of this lab are as follows:

- Create a testing dataset intended to verify the invariance of the SIFT descriptor
- Test Lowe's code with a set of images and produce results about the percentage of correct matches for every pair of images
- Implement a modified version of the SIFT descriptor and compare it with the results obtained using the standard SIFT implementation (using Lowe's code)

2 Introduction

SIFT (Scale Invariant Feature Transform) is one of the most well known algorithm in computer vision which is used to detect the most relevant features of the given images using keypoints and to describe them through features. The main goal of this lab is to verify the invariance of SIFT descriptor based on many factors like rotation and scaling, which are explained in the below sections. This report divides into few sections which will describes about generating the testing dataset from the given large data image and then to test the performance of the SIFT algorithm. Later sections also compares with the original David Lowe SIFT algorithm and at the end all the implementation results are plotted and saved by using MATLAB.

3 Development

For generating the testing dataset image we have given a synthetic underwater scene which has the dimensions of 11688×11536, which has huge number of pixels as shown in Figure **??**. As this image is huge in size we will crop the window size into 1500×1500 from the center of the original image.This image as *large_crop* is shown in figure **??**. By using this underwater image we will generate three different sequences correspond to various movements of camera like Projective, Scaling and Rotation transformation. All this sequences are composed as below:

- A reference image, which is common for all the sequences
- A set of images describing different camera movements around the same region of interest of the scene

We will generate a reference image of 750×500 as *Image_00a* from the centre of the *large_crop* image as shown in figure **??**.

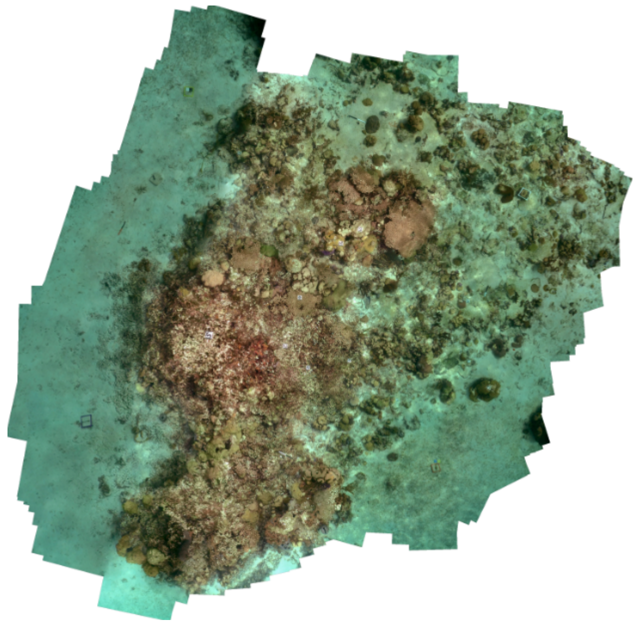

Figure 1: The original given underwater scene picture

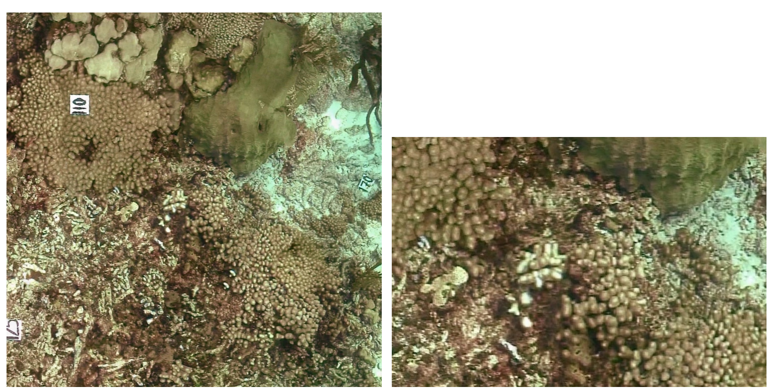

Figure 2: LEFT:The cropped image size of 1500X1500. RIGHT: Reference image for all the sequences

3.1 Sequence Generation

Initially, before actually testing the functioning of sift, a set sequences for the main were to be generated. Sets of Rotation, Scaling and Projective sequences were to be generated along with the homography matrices between the transformations for each sequence. The guidelines mentioned in the instruction manual, folder structure, for example, were all followed.

As mentioned for **projective transformations**, a set of 16 images in top, bottom, left and right directions, with 4 projected images for each direction with a scale factor of 0.1 were generated. The projections were interpolated within the image using the function $fitgeotrans$.

Next, a set of images for **Zoom** were generated. The new images were generated at a scale factor of 0.05, incrementally from the reference image. The homography matrix for zooming is given s follows:

$$A = \begin{pmatrix} scale & 0 & 0 \\ 0 & scale & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Lastly, for **Rotation Sequences**, a set of 18 images, taken at angles ranging from -45 to 45 degrees were generated. The homography matrix for rotations can be given as follows:

$$A = \begin{pmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The homography matrices were tested by matching pairs of images for each of the sequences, by using the code provided in the instructions. Figure 3, shows a sample for a pair of images, with the comparison point in green.

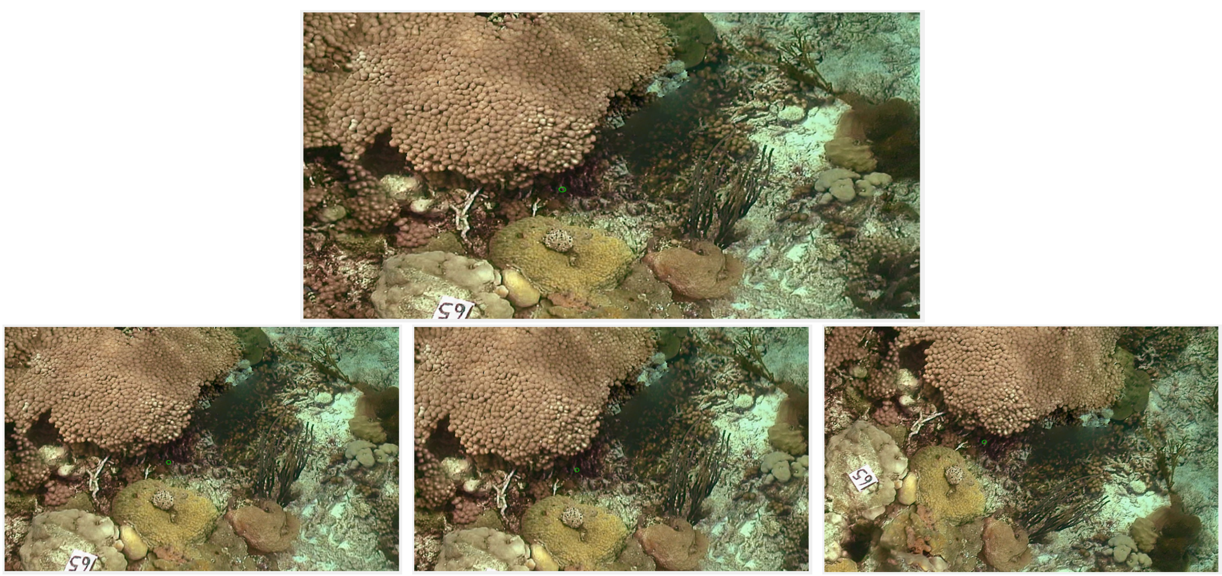

Figure 3: Matching the points for all the sequences **TOP**:Reference Image **BOTTOM:LEFT TO RIGHT**:Projective, Scaling and Rotation

3.2 Testing the performance of the SIFT Descriptor

Under this part, the closeness of the locations of the generated SIFT keypoints for the original image and each of the projected images was to be computed, and, if the distance between the keypoints lies within the distance threshold of 2, the keypoints are considered to be matched. Implicit functions provided in the David Lowe's database were used to get the matching keypoints. The accuracy of the matching points is plotted on the graph, for each of the projected image with different levels of noise. The results are shown in Figure 4, it plots the accuracy plots for keypoint matching between the original and each of the projected images for all the sequences.

From the plotted graphs, we can note that although the implementation has high computational cost, it generates very accurate results. We can see in the graph, that the keypoints match accuracy is almost constant for any projected image. But, fewer keypoints match when the noise is very high.

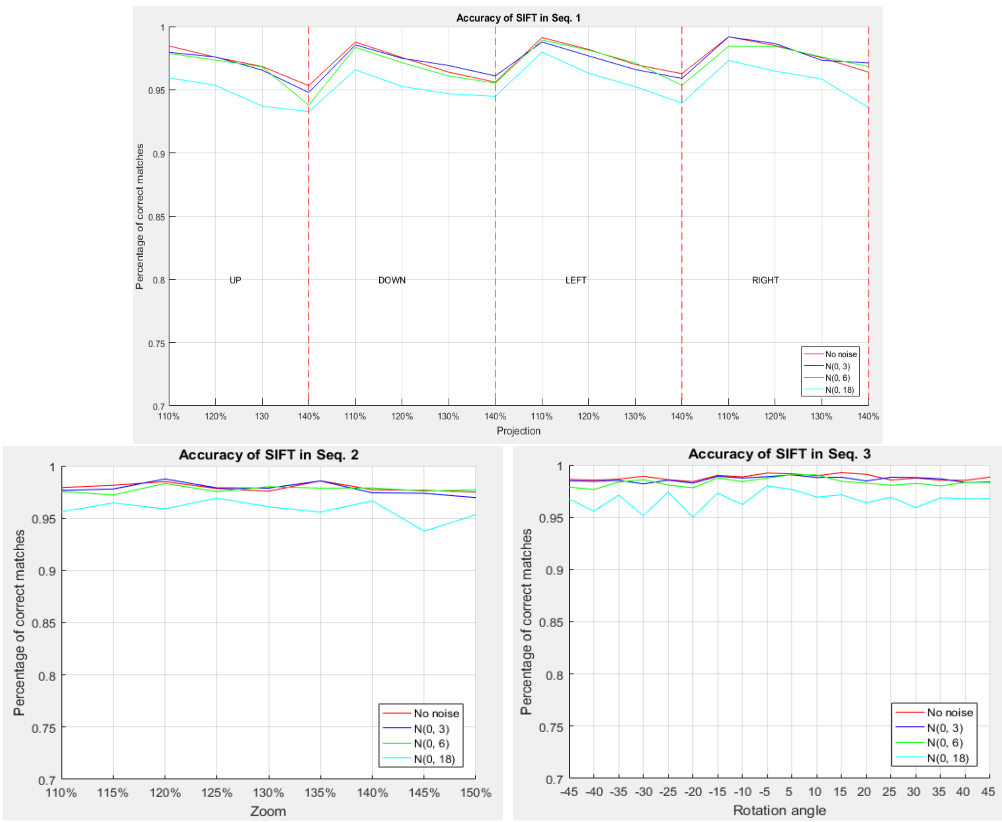

Figure 4: Results showing percentage of matching keypoints comparing with David Lowe Features

3.3 Implementing Modified SIFT Descriptor

An implementation of SIFT, other than that of David Lowe's was to be used under this section. *vl_sift* from *vl_feat* library was used to accomplish this task. Our task was to implement SIFT in scale space representation and compare the original (without any scale) was to be compared with scaled images (TEAM B). The *vl_sift* function provides an implementation to obtain descriptors in scale representation. We initially, get the descriptors with the default parameters and get the features. The third row in the features represents the scale factor and has to be replaced with $1's$. We then compute the descriptors again by the passing back the features to obtain scaled descriptors.

Figure 5 shows the results for keypoint matching accuracy, for scaled descriptors. We may note from the figures that the new scaled descriptors greatly affect the zoom sequence of images. But, for the other sequences, the results are consistent. This behaviour can be justified by saying that, as the scale reduces to a great extent where the image is not even clearly visible, very few SIFT keypoints would be extracted. Thus only small percentage of keypoints match as the scale keeps reducing. Accuracy tends to zero with greater decrease in scale.

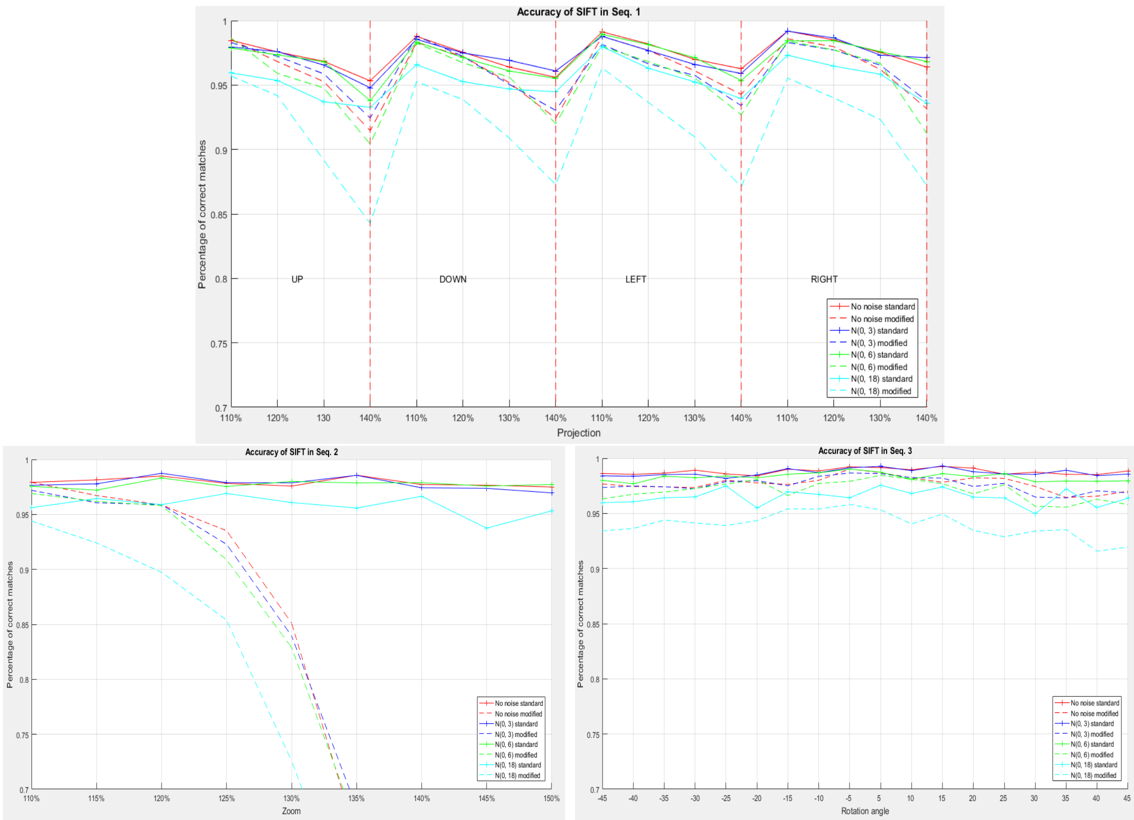

Figure 5: Results showing percentage of matching keypoints

4 Conclusion

For this lab we have worked in a team of two for SIFT invariant features. At first we have projected three different sequences from the given huge size of underwater scene by cropping it to the desired size. All the transformations of Projection, Scaling and rotation are generated and compared by pointing. The next section we will plot the graphs by comparing from the original David Lowe features and at the end we will conclude the result as per the TEAM B, which describes about the scale-space representation and discuss the results without scaling factor.