ENGN 2560 Computer Vision: Homework #1
Feature Detection / Extraction

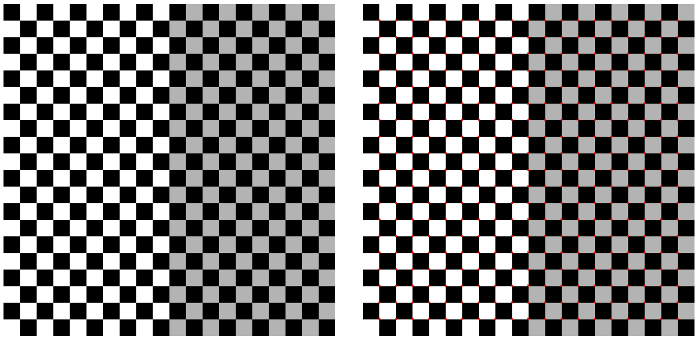
Due: February 14, 2019

In this programming assignment you will practice detecting feature points in images and extracting features from them.

A cell mode MATLAB script, main.m, is provided. Please develop your code in its respective section. Each cell in the script can be run in isolation using the **“Run Section”** on the editor tab or using **“Ctrl + Enter”**. Organize your code such that there is no data dependency across sections as each section will be run and evaluated independently. Make use of comments to improve readability. The organization of the first code section provided serves as an example. Your homework submission should include your source code and a brief PDF report answering the questions with supporting visuals. Hand in your submission in a ZIP file with filename “FirstName_LastName_Homework1.zip”.

Question 1

a) Implement the Harris Corner Detection algorithm detailed in lecture notes on pages 4-7. Your implementation should go inside the detectCornerKeypoints.m file. Before detecting corner points on real images, this part will verify the correctness of your implementation. Your algorithm should be able to detect the corners of a checkerboard pattern. An expected corner detection result on a synthetic checkerboard image is shown below. Run your algorithm on the checkerboard image and include a visual of the detected corner points.



(a) (b)

Figure 1: (a) A 1000 × 1000 synthetic checkerboard image. (b) Corners detected and marked with red dots. Zoom in for better visibility.

b) Extract corner keypoints from the “goldengate − 02.png” and “goldengate − 03.png” images by calling your corner detection algorithm. Include visuals of the detected keypoints shown on top of the original images. An expected corner detection result on the “goldengate − 02.png” image is shown below. The images provided are taken from slightly different viewpoints. Are the extracted corner points around the regions common to both images repeatable across different viewpoints? Is “repeatability” of interest points under viewpoint variation important? If so, in what ways?

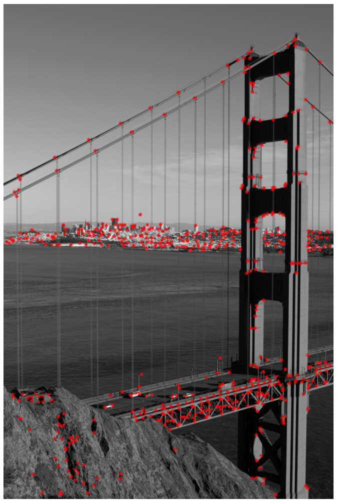


Figure 2: Corner detection result on the “goldengate − 02.png” image.

Question 2

a) Extract FAST keypoints from the “goldengate − 02.png” and “goldengate − 03.png” images. You are allowed to make use of the function detectFASTFeatures in MATLAB. Include the visuals of the extracted keypoints. An expected keypoint detection result on the “goldengate − 02.png” image is shown below. Are FAST keypoints repeatable under viewpoint variation?

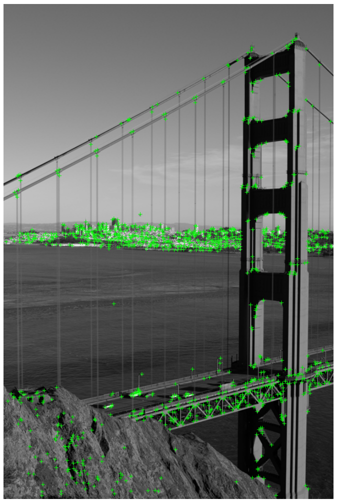


Figure 3: FAST keypoint detection result on the “goldengate − 02.png” image.

b) Extract SURF keypoints from the “goldengate − 02.png” and “goldengate − 03.png” images. You are allowed to make use of the function detectSURFFeatures in MATLAB. Read the documentation of the detectSURFFeatures function. In particular, familiarize yourself with extracting the Orientation and Scale of the detected keypoints. Include the visuals of the extracted keypoints. An expected keypoint detection result on the “goldengate − 02.png” image is shown below. Are SURF keypoints repeatable under viewpoint variation?

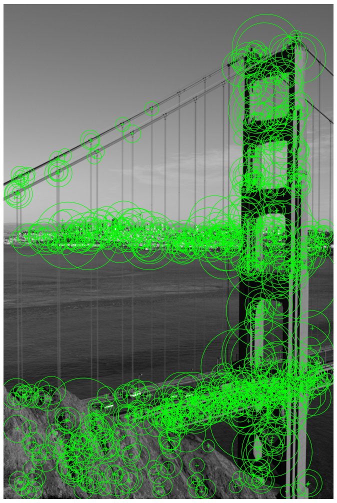


Figure 4: SURF keypoint detection result on the “goldengate − 02.png” image.

Question 3

Download and install the VLFeat library from `http://www.vlfeat.org/`. Read the documentation of extracting SIFT features from an image available on `http://www.vlfeat.org/overview/sift.html`. Detect SIFT keypoints and extract SIFT features on “goldengate − 02.png” using vl_sift. Overlay a SIFT descriptor frame at sub-pixel $(X, Y) = (377, 145.6)$. Include the visual of the extracted keypoints and the feature frame. An expected result is shown below.

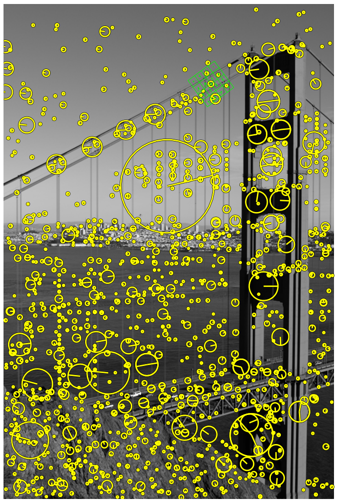


Figure 5: SIFT keypoint detection result on the “goldengate − 02.png” image. A feature frame at sub-pixel location $(X, Y) = (377, 145.6)$ is also shown.

Question 4

It is sometimes useful to describe an interest point or an interest region using multiple descriptors. For this question, you will first detect SURF keypoints and extract a SIFT descriptor on a given keypoint using the scale and orientation identified by the SURF keypoint detector. Extract SURF keypoints on the “goldengate − 03.png”. Unless there are differences in implementation between different versions of MATLAB, this should yield a keypoint detected at pixel $(X, Y) = (494, 441)$. If there is no detected keypoint at this pixel, you can pick another keypoint. Using the “scale” and “orientation” of the detected keypoint, extract a SIFT feature using a custom feature frame.

Include the visual of the SIFT frame and the SURF keypoint. An expected result is shown below. Notice how the SIFT feature frame properly covers the interest region (shown as a circle) of the detected SURF keypoint with the use of scale information returned by the SURF detector.

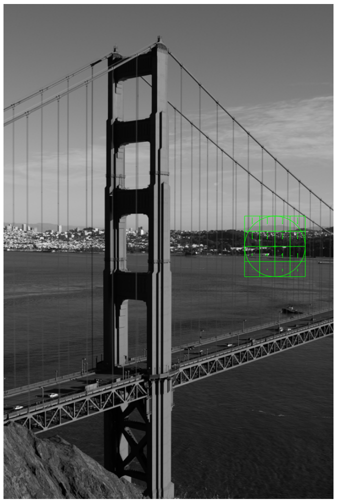


Figure 6: A SURF keypoint and SIFT feature frame result on the "goldengate − 03.png" image. The feature frame is located at pixel $(X, Y) = (494, 441)$.