CV Assignment 2

Vinit Ramesh Gore (M20CS064)

Question 1

Image stitching is being done using an in-built function from openCV.
There are four steps involved in image stitching:
Consider two adjacent images at a time

1. Extract features from the images. SIFT/SURF functions are generally used for the same.
2. Match the features from both images.
3. Derive homomorphic matrix H (could perform translation, affine, rotation as well as projective transform) for second image w.r.t first image.
4. Align the images and blend them.

Repeat till all images are sequentially stitched together.

Image Stitching of the following photos:

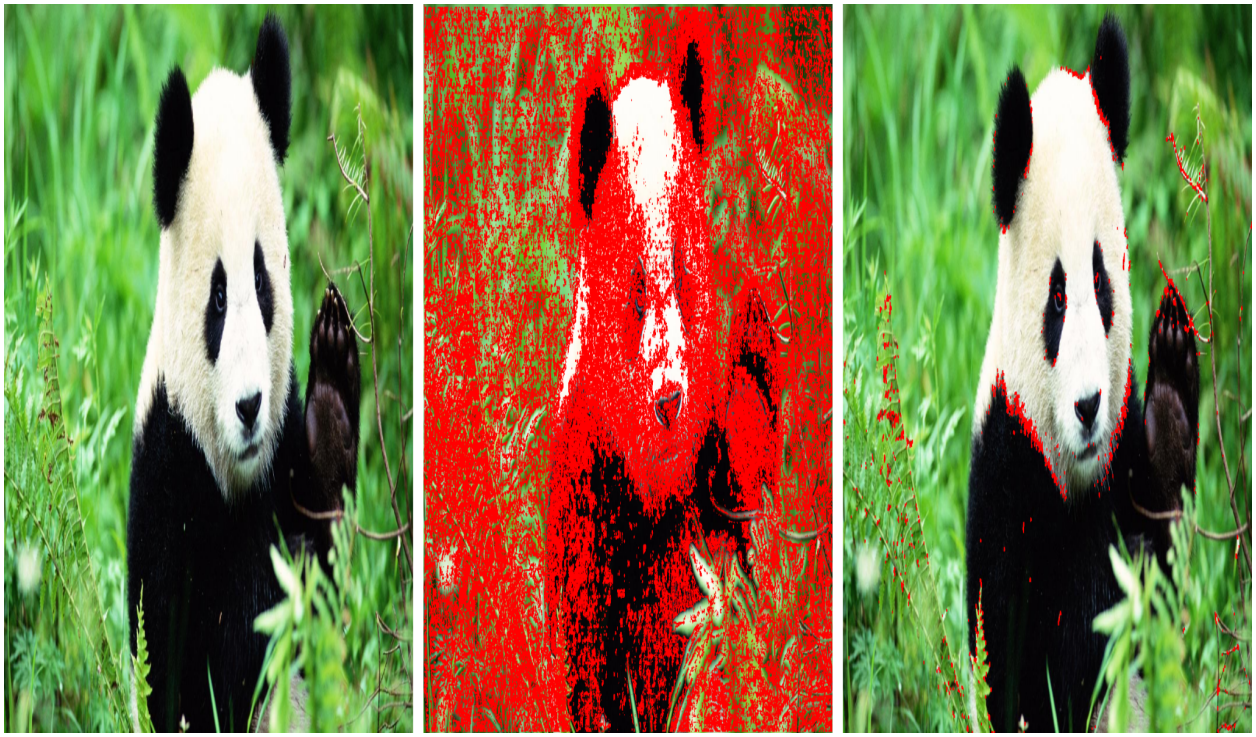
Result Panorama:

Unfortunately, the fifth image was not stitched most likely because enough features were not matched due to small light variations, small movements of leaves due to wind, etc.

Question 2

The Harris Corner detection approach is being used to detect corners. Referring to the opencv document, the window size was fixed to 2. Increasing window size led to detection of larger corners. It also detected The outputs for window size as 20 are displayed below. The approach to find the threshold was taken from lower value to higher value in a brute-force manner. There had to be a better way. Instead of returning the corners from the corner detection function, it was changed to return the R values. This result is then dilated and all values within 0.01 range below max value from the result are marked as corners. This resulted in similar results to in-built functions.
Results:
(a b c) a: Original Image, b: Bruteforce bottom up approach to find threshold on Harris method c: In-built Opencv Harris Corner Detector

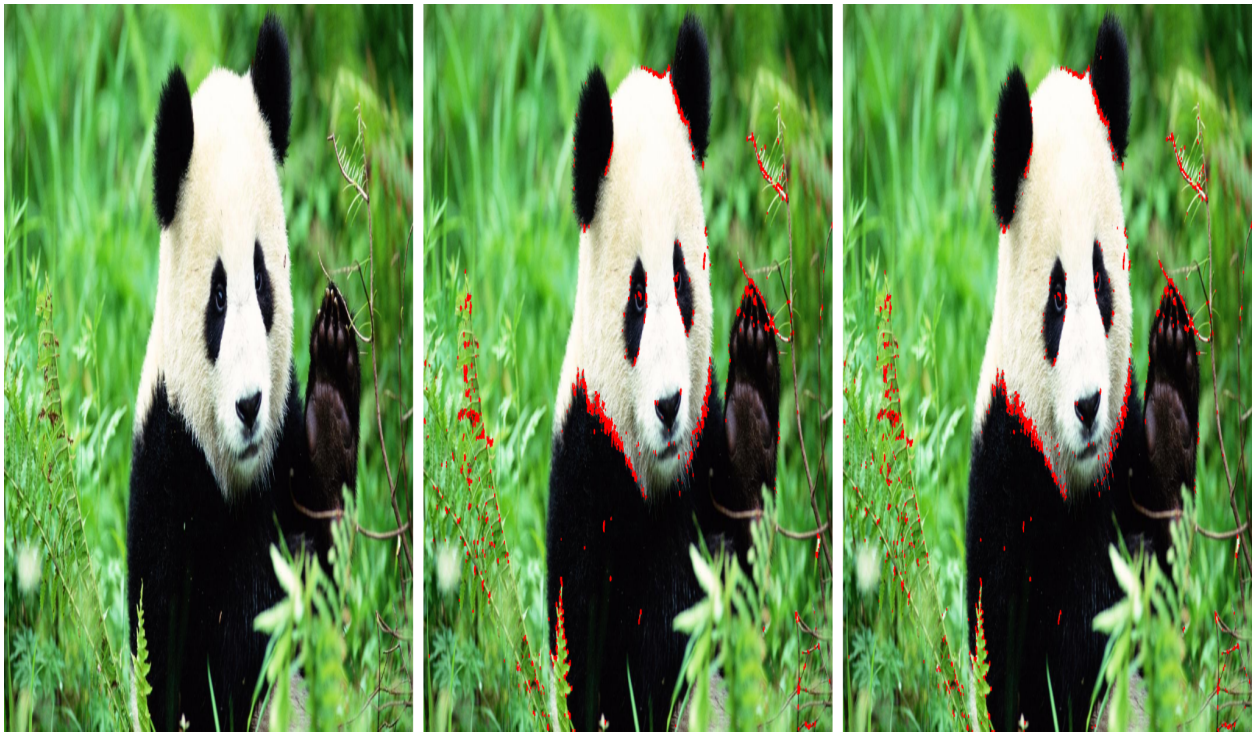
1



Result for window size = 20 :
(a b c) a: Original Image, b: Harris method using values within 0.01 range below max value. Sobel gradient used. c: In-built Opencv Harris Corner Detector
1

10

Results:
(a b c) a: Original Image, b: Harris method using values within 0.01 range below max value. Sobel gradient used. c: In-built Opencv Harris Corner Detector
1



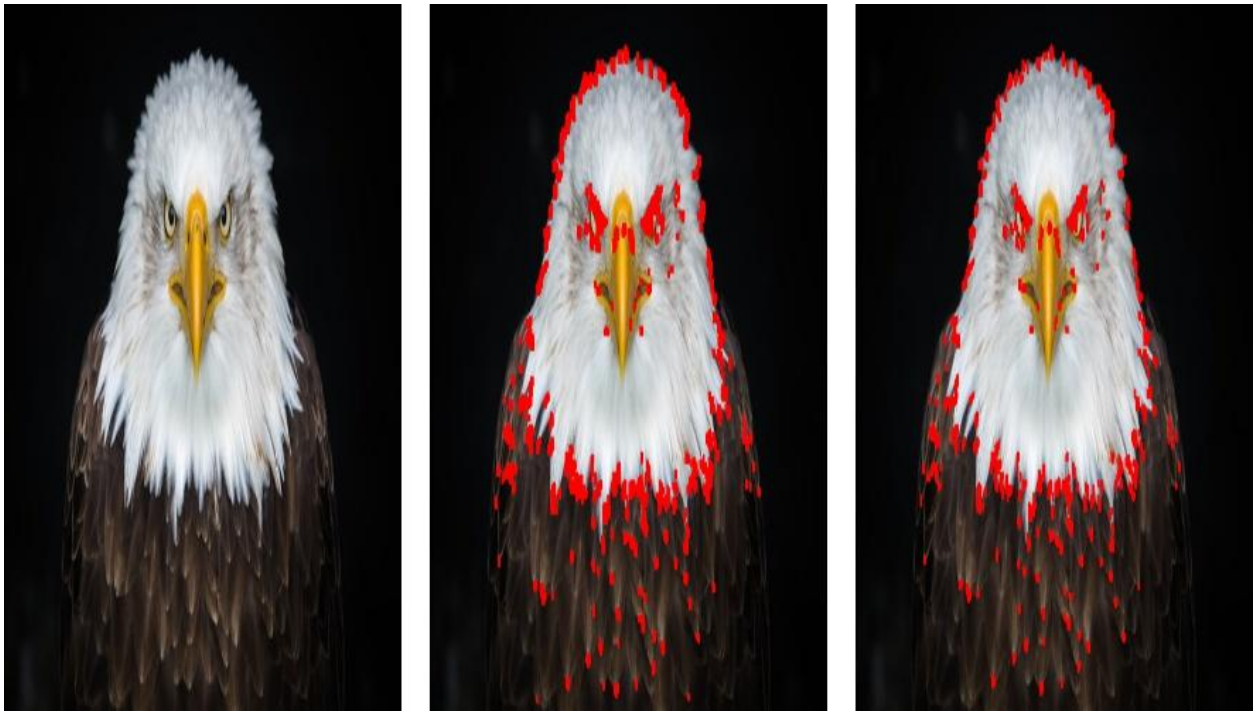
2

3

4

5

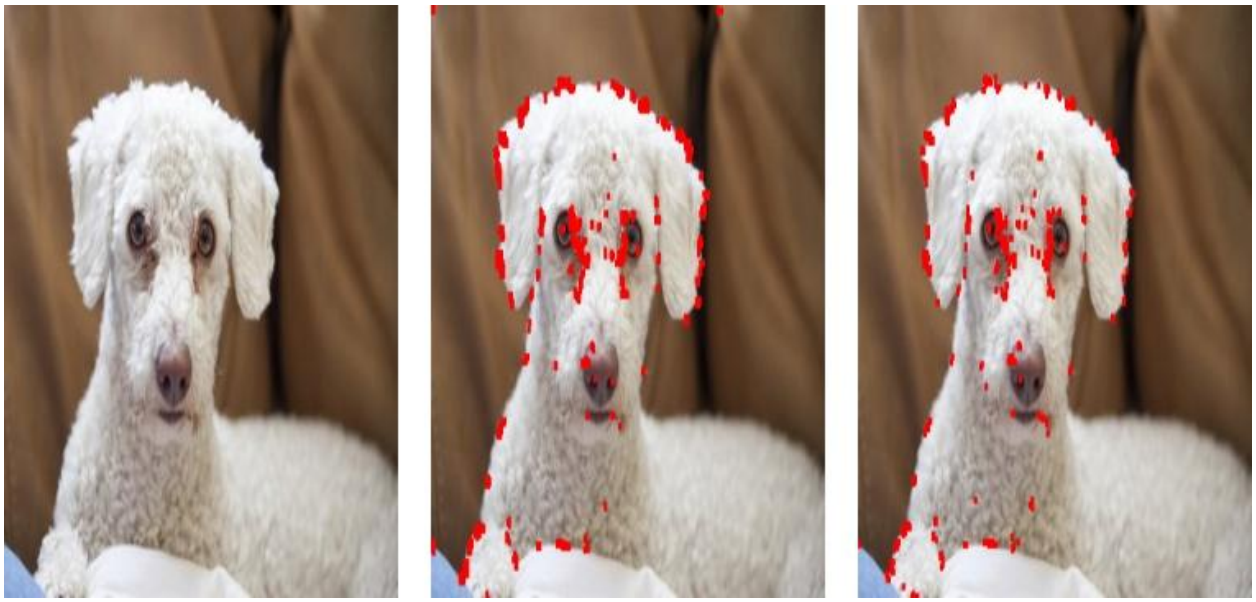
6



7

8

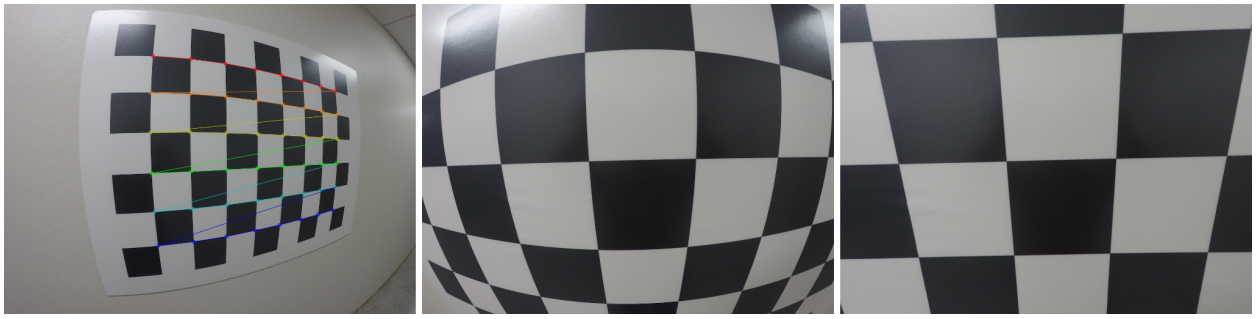
10



Some differences are visible in the two outputs of the last three images. These are most likely to be coming from the way of usage of the gradient. The Scharr kernel is being used for the above results.

Question 3

Camera Calibration is required to remove the radial and tangential distortions from the image. It is performed by learning the intrinsic and the extrinsic parameters with the help of 3D-to-2D mappings derived from the input data. Since the object (chessboard) is stationary and only the camera is moving around, the z-component of the 3D (object) points would remain zero. This makes assigning 3D points easier (by assigning the x and y components with grid values in range [0:8, 0:6] in the current problem). 2D (Image) points are detected from the points in the images where two black squares are touching each other.
When we have these two sets, we can use them to calibrate the camera and find the parameters. Using these parameters, we can correct the distortion from a given distorted image.



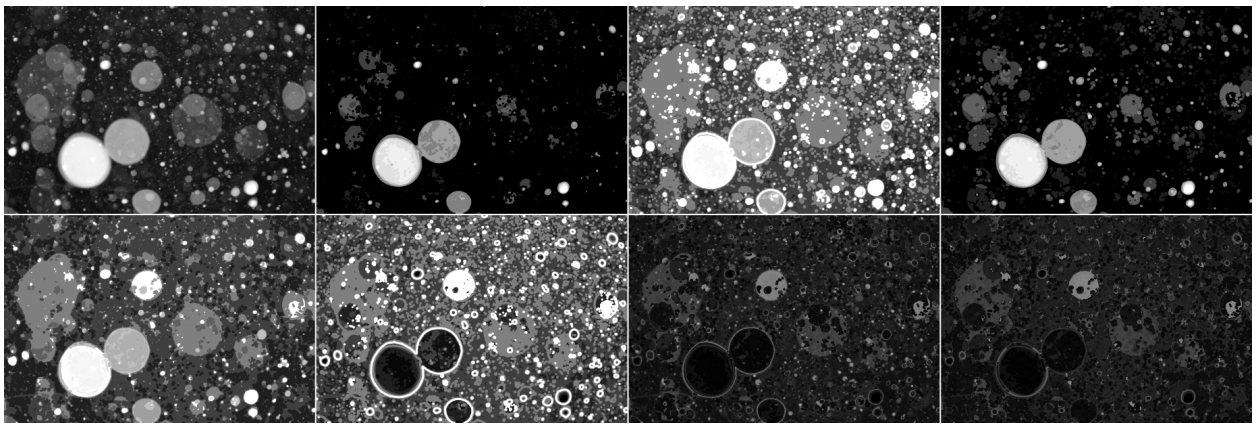
(a b c) a: Image (2D) points drawn on 1.jpg, b: Distorted image (test_image), c: Calibrated version of (b).

The result of the calibrated image could be focussed more correctly using `cv2.getOptimalNewCameraMatrix` but to retain the size, the original camera matrix was used. The result is not 100% accurate. The total error incurred is `0.07848402246105138`.

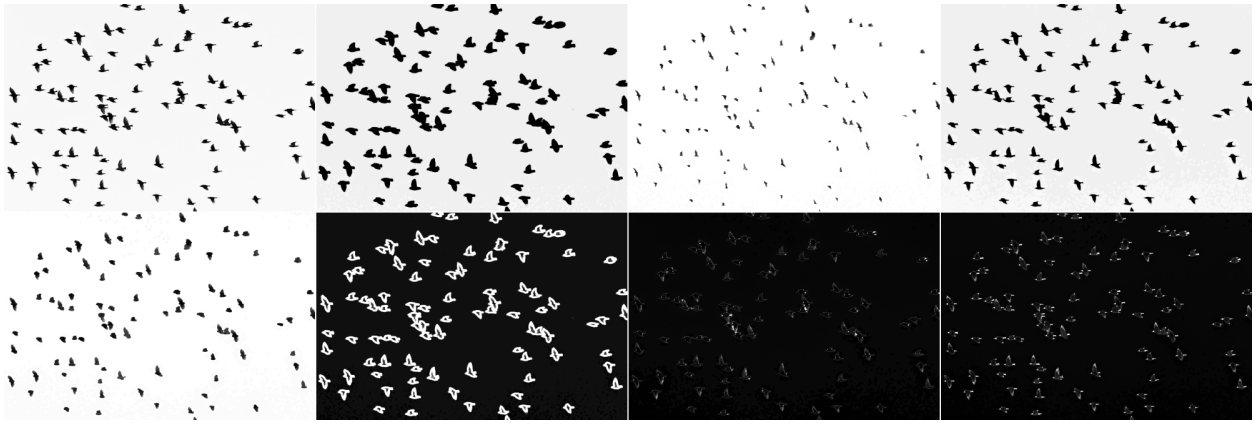
Question 4

The morphological operations are being performed on bit planes and the results of all bit planes are being again added up to get the result image for every operation on every image. The window of 30x30 size is being used for every image.
Results:
(a b c d) a: Original b: Erosion, c: Dilation, d: Opening
(e f g h) e: Closing, f: Morphological Gradient, g: Top Hat, h: Black Hat

1

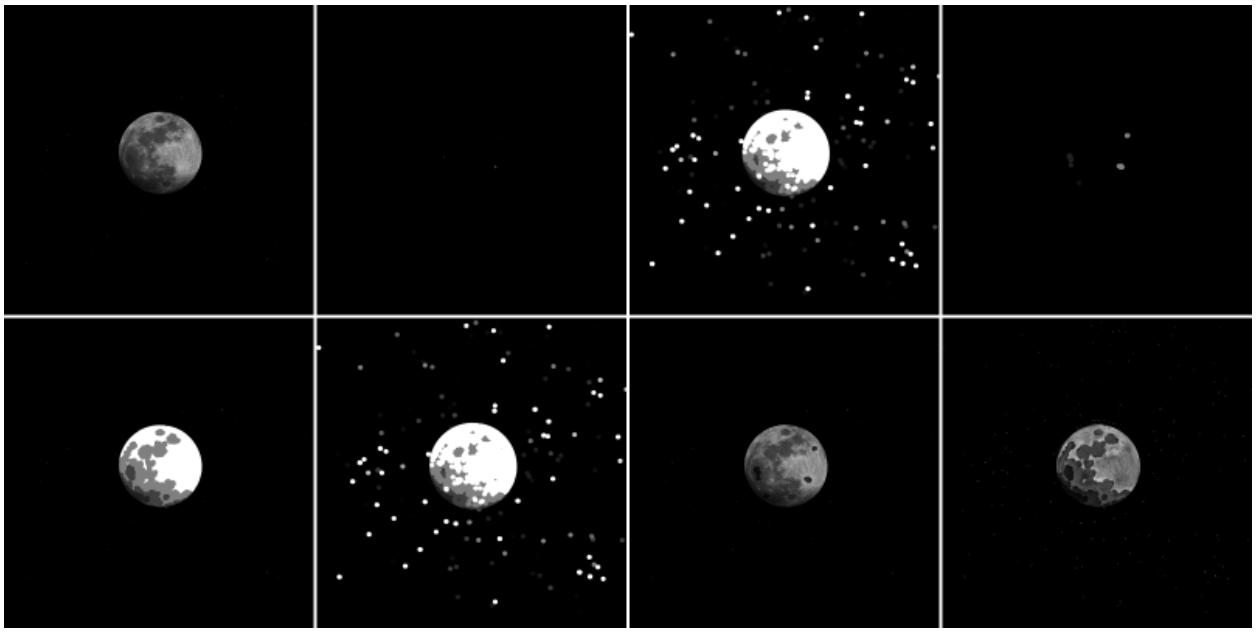


2



In this case, since the objects were black and background was white so erosion worked like dilation and dilation as erosion.

3



4

5