E9 241: Digital Image Processing

Assignment 5

Harris Corner Detection

The following are the observations on Harris corner detection algorithm with respect to different image transformations.

Threshold and window size:

For an image with clearly distinctive corners as in the checkerboard image, Harris corner detection works well being able to detect the corners even on changing the window size and threshold as observed for the 'checkerboard' image. But the number of corners detected were less dense on increasing threshold. In natural images as in the second image, the threshold significantly affects the corners detected. Also, the algorithm is very sensitive to window size as it defines the area corners are contained. Window size must be carefully chosen depending on the image content and its resolution. Following are the results obtained for different threshold and window size:

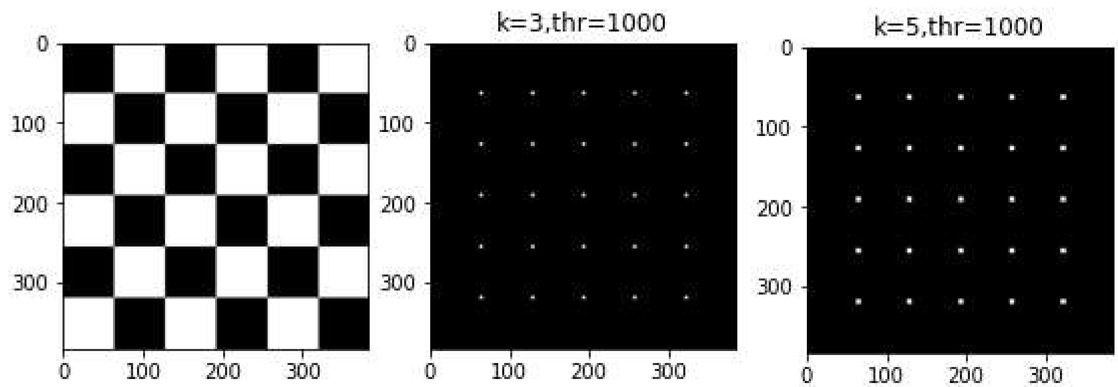

Fig1. Comparison for window sizes 3, 5

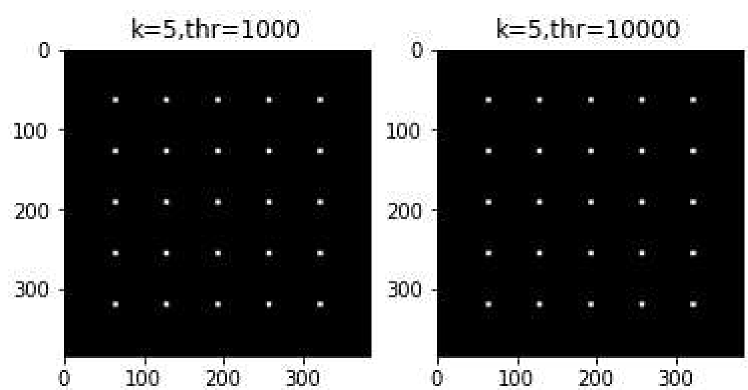

Fig2. Comparison for thresholds 1000, 10000

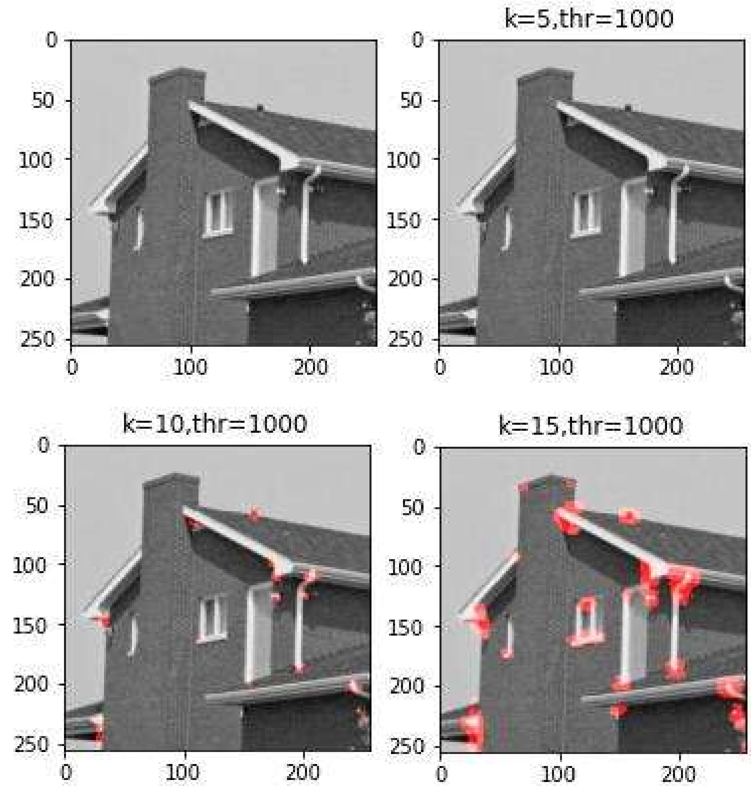

Fig3. Comparison for window sizes 5, 10, 15

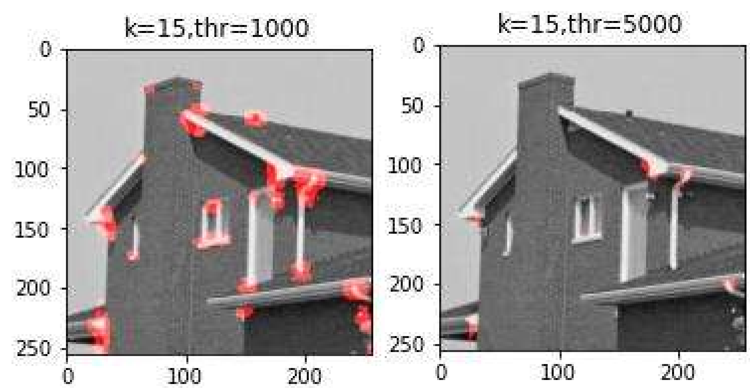

Fig4. Comparison for thresholds 1000, 5000

Resize:

On resizing the images, corners can be lost. The window size must often be modified appropriately to be able to detect corners. In checkerboard image, window size of 3 also was sufficient to detect corners in the original image. The following are the results after upscaling the image by 2. In the checkerboard image, corners were no longer detected for window size 3. However, a window size of 5 was able to detect the corners in the upscaled image. In the second image, a window size of 15 could detect corners in the original image but the same couldn't detect any corners. A window size of 20/30 was necessary to detect the corners in the upscaled image.

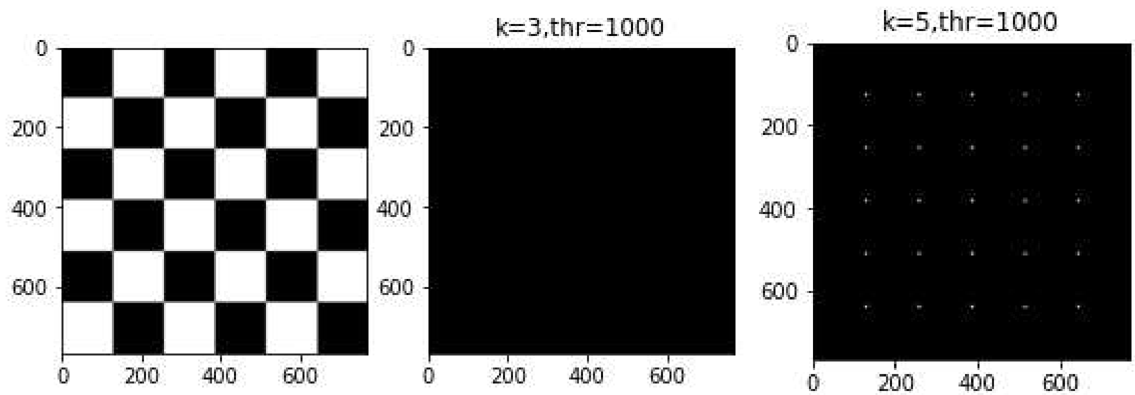

Fig5. Comparison for window sizes 3, 5 for 2X upscaled image

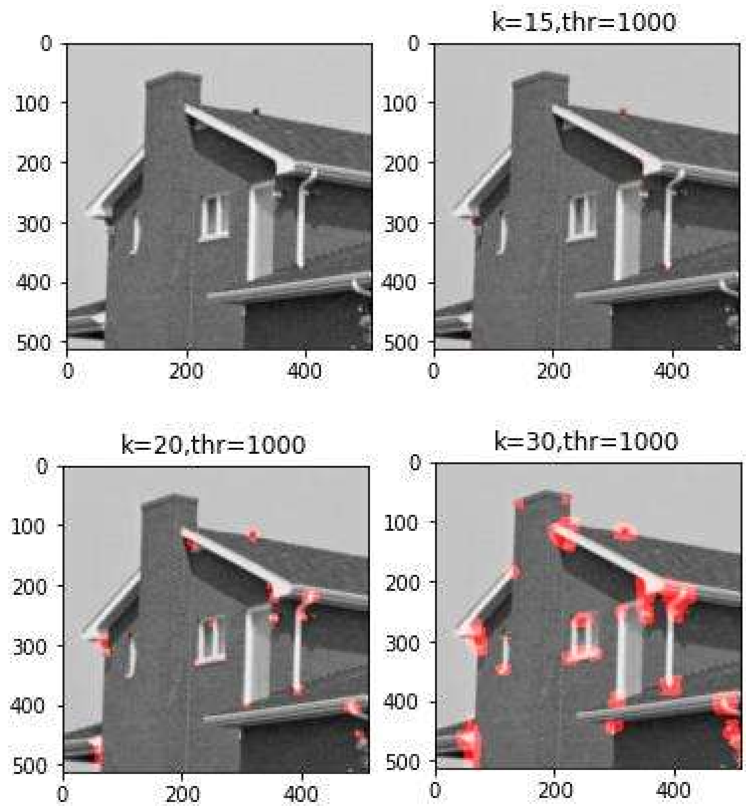

Fig6. Comparison for window sizes 15, 20, 30 for 2X upscaled image

Rotation:

This method is invariant to rotation. So, corners are detected on rotating an image without any change in parameters used for the original image. Following are the results for image rotated by 30°. The threshold maps look like rotated versions of the previously shown results for the same parameters.

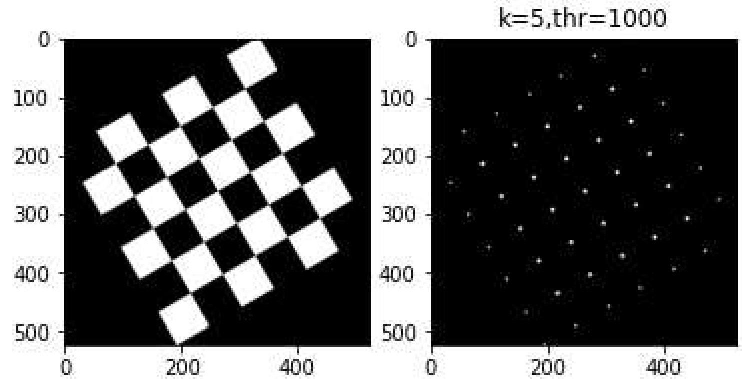

Fig7. Corners detected for 30° rotated image

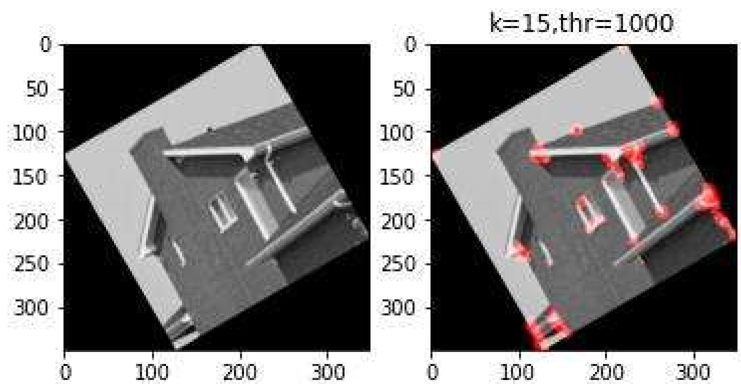

Fig8. Corners detected for 30° rotated image

Noise:

This method is very sensitive to noise. It often results in detecting a lot of corners resulting in extracting a lot of unnecessary information. In general, the parameters tuned to extract corners in clean images might not extract any meaningful information from that noisy image and highly depends on the noise level. In checkerboard image, along the true corners more corners were detected. Threshold had to be increased in presence of noise to get better results. In the second image, only for a very small noise variance of 0.001, the corners are detected. But increasing the noise level to sigma=0.1, a lot of corners throughout the image are detected and hence not providing any useful information.

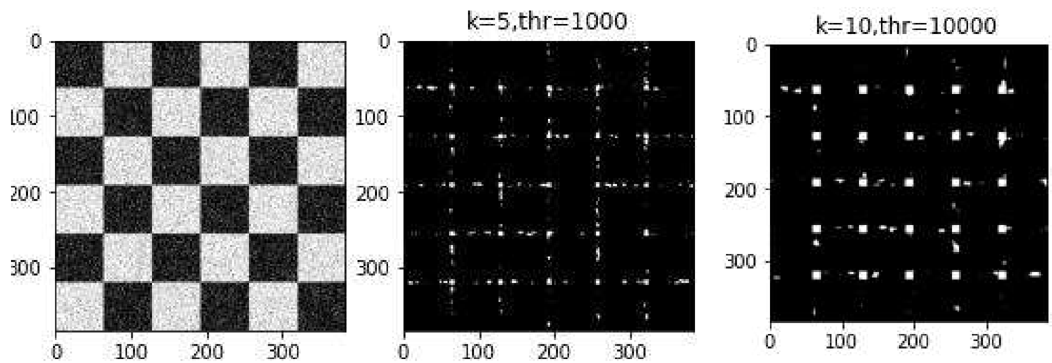

Fig9. Corners detected for noisy image (sigma = 0.1)

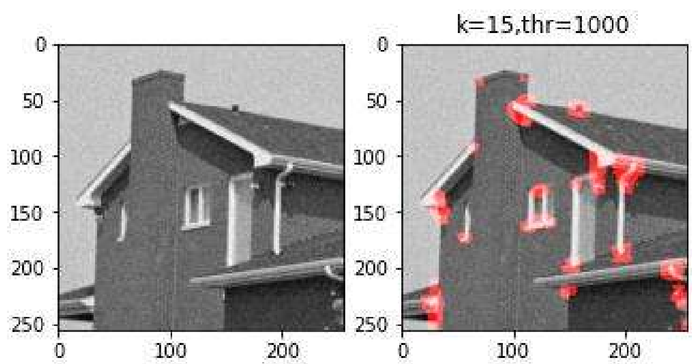

Fig10. Corners detected for noisy image (sigma = 0.001)

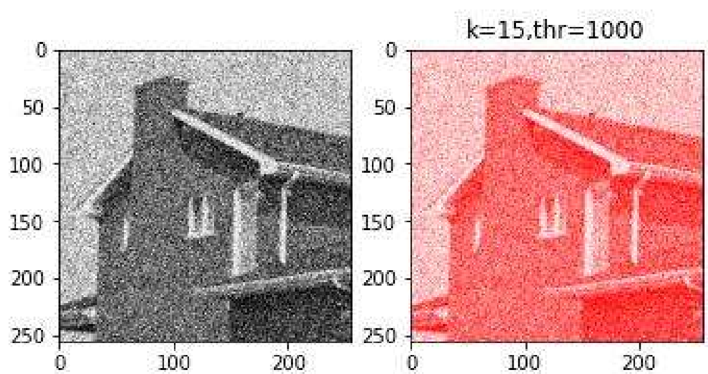

Fig11. Corners detected for noisy image (sigma = 0.05)