Computer Vision, IIIT Sricity (Spring 2018)

Dr. Shiv Ram Dubey

Programming Assignment – 1

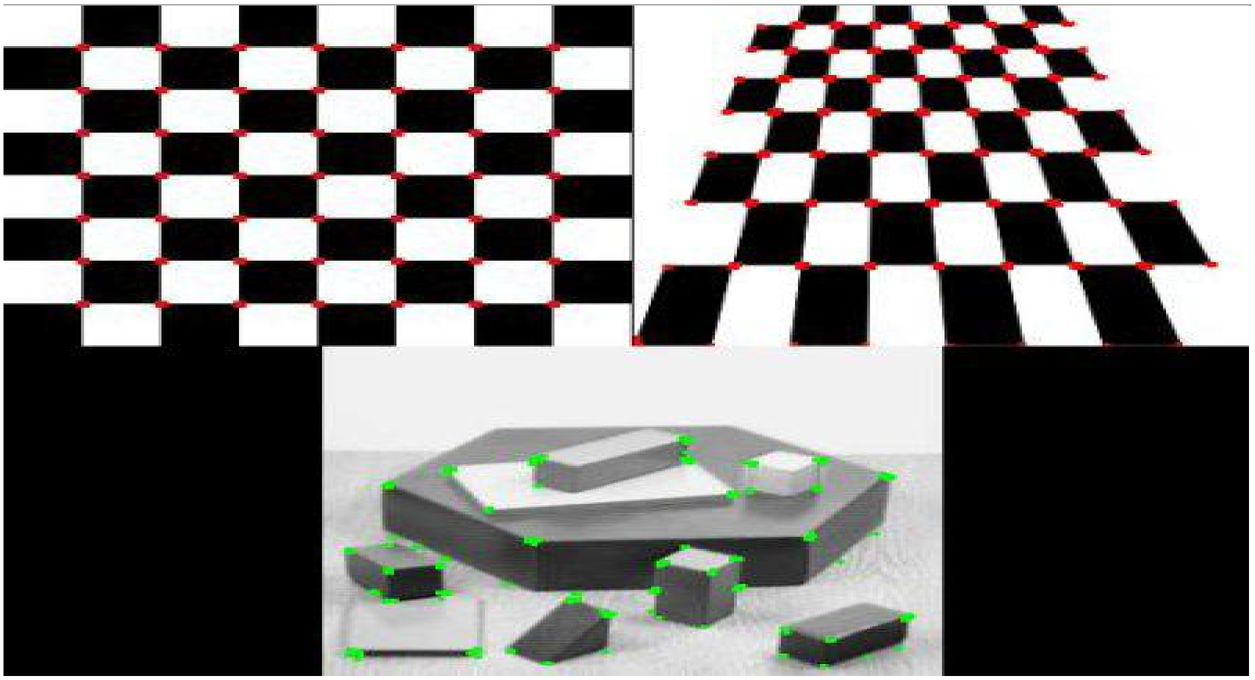
Release Date: 23-Jan-2018, **Deadline: 08-Feb-2018 (5.00 pm)**

1. Hybrid Images (30 Point)

The goal of this assignment is to introduce you to high frequency and low frequency information of the image. We will focus on the task of creating a hybrid image from two images. Your task is to sum the high frequency of one image and low frequency of other image to get the hybrid image. Some images are provided to work with.

- Implement a function to get the high frequency of Image1. Let's consider, the threshold is α for high frequency information.
- Implement a function to get the low frequency of Image2. Let's consider, the threshold is β for low frequency information.
- Implement a function to combine the high frequency of Image1 and low frequency of Image2.
- Analyze the hybrid image by changing the values of α and β. Note that, if you are using Gausian filter for constructing the hybrid images, then α and β can be implemented in terms of the standard deviation of the Gausian filter.
- **Extra Credit:** Consider three images. a) Compute the hybrid image by considering the high frequencies of Image1 and Image2 in a ratio of γ:δ and low frequency of Image3, b) Compute the hybrid image by considering the low frequencies of Image1 and Image2 in a ratio of γ:δ and high frequency of Image3, and c) Analyze the result by varying the ratio γ:δ.

2. Corner Detection (30 Point)



In this question, you will implement different versions of the corner detection algorithms for given three input images (Image1.jpg, Image2.jpg, and Image3.jpg).

- Implement Shi-Tomasi corner detection algorithm. Find out the max and min eigen-values (i.e. λ_{max} and λ_{min}) and use thresholding over min eigen-value (i.e. Consider cornerness measure (f) as the min eigen-value, λ_{min}) for corner detection.
- Implement Harris Corner Detection algorithm for the same input images you used in previous question. In this problem, consider the cornerness measure (f) as [Det(H) $-$ α.Tr(H)], where H is the hessian matrix, Det is the determinant of H, Tr is the trace of H, and α is a constant in between 0.02 and 0.04. Please use non-negative $\alpha \approx 1/25$ as a starting value and try to optimize it by trying different values and comment about it.
- In the previous question, replace cornerness measure with the following:[$\lambda_1\lambda_2$ - $\alpha(\lambda_1+\lambda_2)$] and and determine the efficiency of this system and the system in the previous question by measuring and reporting the time. You are supposed to get the same results in accuracy but different results in efficiency.
- In addition to the images provided, also run your code on at least three images of your own choosing.

3. Scale-space Blob Detection (40 Point)

The goal of this assignment is to implement a Laplacian blob detector.

Algorithm outline

1. Generate a Laplacian of Gaussian filter.
2. Build a Laplacian scale space, starting with some initial scale and going for n iterations:
    1. Filter image with scale-normalized Laplacian at current scale.
    2. Save square of Laplacian response for current level of scale space.
    3. Increase scale by a factor k.
3. Perform nonmaximum suppression in scale space.
4. Display resulting circles at their characteristic scales.

- Four images with expected output images are provided to test your code. Keep in mind, though, that your output may look different depending on your threshold, range of scales, and other implementation details. In addition to the images provided, also run your code on at least four images of your own choosing.
- **Extra Credit:** Implement the difference-of-Gaussian pyramid as mentioned in class and described in David Lowe's SIFT paper. Compare the results and the running time to the direct Laplacian implementation.

Acknowledgments: Some part of questions are taken from the projects developed by Svetlana Lazebnik (illinois.), Ulas Bagci (UCF), Derek Hoiem (UIUC) and James Hays (GaTech).