



جامعة أبوظبي
ABU DHABI UNIVERSITY

ABU DHABI UNIVERSITY

CEN 464 - DIGITAL SIGNAL PROCESSING

Project I Report
Digital Signal Processing of Video Signal
using MATLAB

Authors:

Muhammad Obaidullah 1030313

Supervisor:

Dr. Mohammed Assad Ghazal

Section 1

June 9, 2014

Contents

1	Introduction	2
2	List of equipment used	3
3	Project Code	3
4	Results & Discussions	4
5	Conclusion	13

Abstract

In this project we wrote a MATLAB code for continuously grabbing the image frame in real time and plotting 2 dimensional Fast Fourier Transform of the current frame. This will allow us to analyze and study how 2D image pixels map to frequency spectrum.

1 Introduction

Digital image is stored in computers as combination of matrices for each channel such as Red, Green, Blue, and Alpha.[3] Any digital image is composed of pixels spread in two dimensional space. Any one of the pixel is represented by its x and y positions on the screen. In gray-level image, a pixel has a value determining how close to black or white color a pixel is. If the pixel values are unsigned integer 8-bits, then value 255 is the whitest and value 0 is blackest. Any value in between is a shade of gray. On the contrary, a color image is composed of pixels which often have three values, the red value, the green value, and the blue value. These values indicate how close to the primary color the pixel is. For example, if a pixel has $R = 255$ $G = 0$ $B = 0$ values then it means the pixel is entirely red.

There are some image formats which also allow an additional value for each pixel to be stored in an image. This is the alpha value, which determines the transparency of the pixel. A value of 255 indicates that the pixel is completely opaque while the 0 value indicates that the pixel is completely transparent. This fourth alpha channel is crucial in many image applications and visual effects because it allows dynamics of the background and sense of reality.

For an image to be processed by MATLAB, it is often recommended to convert the pixel value from unsigned integer to double and map every value from 0-255 to 0-1. [2] This allows the image matrix to become double so that is image operations and processing is done on it, it can handle decimal values. At the end of image processing, the image is converted back to 0-255 range so that the real world displays can handle them.

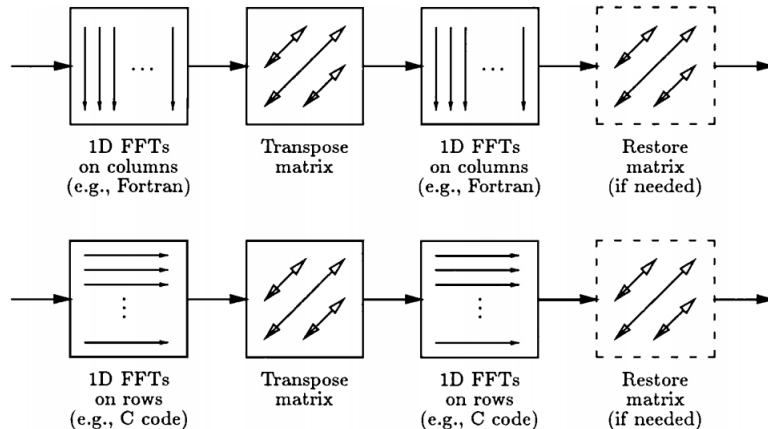


Figure 1: The figure shows how a sequential row-column 2D FFT algorithm is performed using two principal methods. First the Fast Fourier Transform is done column wise and the the transpose of the matrix is taken. Then the FFT is performed again on the resulting matrix column wise. Then the transpose of the matrix is again taken to restore the matrix.[1]

2 List of equipment used

- A Computer.
- MATLAB.
- MATLAB Image Acquisition Toolbox.

3 Project Code

```
1 function realVideo()
3 % Define frame rate
  NumberFrameDisplayPerSecond=10;
5
6 % Open figure
7 hFigure=figure(1);
9 % Set-up webcam video input
  try
11     % For windows
      vid = videoinput('winvideo', 1);
13 catch
      try
15         % For macs.
            vid = videoinput('macvideo', 1);
17         catch
            error('No webcam available');
19         end
      end
21
22 % Set parameters for video
23 % Acquire only one frame each time
  set(vid, 'FramesPerTrigger', 1);
25 % Go on forever until stopped
  set(vid, 'TriggerRepeat', Inf);
27 % Get a grayscale image
  set(vid, 'ReturnedColorSpace', 'RGB');
29 triggerconfig(vid, 'Manual');
31
32 % set up timer object
  TimerData=timer('TimerFcn', {@FrameRateDisplay, vid}, 'Period', 1/
      NumberFrameDisplayPerSecond, 'ExecutionMode', 'fixedRate', 'BusyMode', 'drop');
33
34 % Start video and timer object
35 start(vid);
  start(TimerData);
37
38 % We go on until the figure is closed
39 uiwait(hFigure);
41
42 % Clean up everything
  stop(TimerData);
43 delete(TimerData);
  stop(vid);
45 delete(vid);
  % clear persistent variables
```

```

47 clear functions;
49 % This function is called by the timer to display one frame of the figure
51 function FrameRateDisplay(obj, event, vid)
persistent IM;
53 persistent handlesRaw;
persistent handlesFft1;
55 persistent handlesifft;
57 % FDetect = vision.CascadeObjectDetector;
59 trigger(vid);
IM=getdata(vid,1,'uint8');
61
if isempty(handlesRaw)
63     % if first execution, we create the figure objects
subplot(2,1,1);
65     handlesRaw=imagesc(IM);
title('CurrentImage');
67
subplot(2,1,2);
69     smallf = rgb2gray(IM(:, :, :));
img = fftshift(smallf);
71     F = fft2(img);
73
F = fftshift(F); % Center FFT
75
F = abs(F); % Get the magnitude
F = log(F+1); % Use log, for perceptual scaling, and +1 since log(0) is
undefined
77     F = mat2gray(F); % Use mat2gray to scale the image between 0 and 1
79
handlesFft1=imagesc(F);
title('Frequency Image of Current Frame');
81
else
83     % We only update what is needed
set(handlesRaw, 'CData',IM);
85     smallf = rgb2gray(IM(:, :, :));
img = fftshift(smallf);
87     F = fft2(img);
89
F = fftshift(F); % Center FFT
91
F = abs(F); % Get the magnitude
F = log(F+1); % Use log, for perceptual scaling, and +1 since log(0) is
undefined
93     F = mat2gray(F); % Use mat2gray to scale the image between 0 and 1
95     set(handlesFft1, 'CData',F);
end

```

4 Results & Discussions

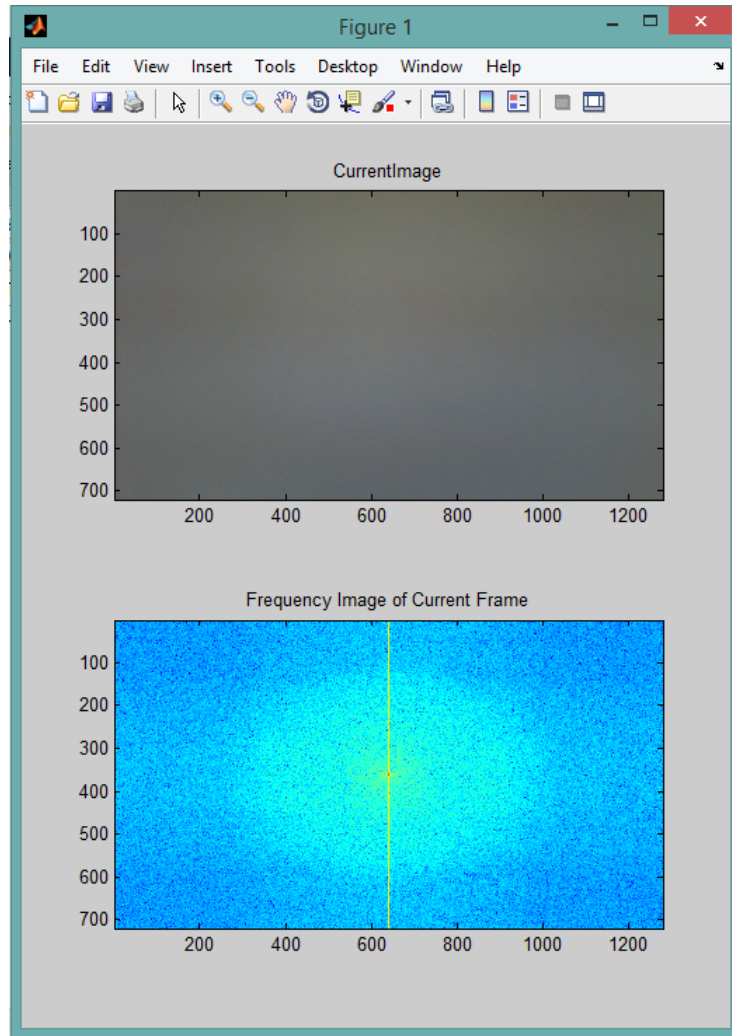


Figure 2: The output figure window. The image capture resolution is 1366 x 720 pixels. The FFT2 plot of the current frame is shown in the subplot 2 of the figure. Since the image captured by the camera contains almost no changes (all white), The FFT plot is quite empty.

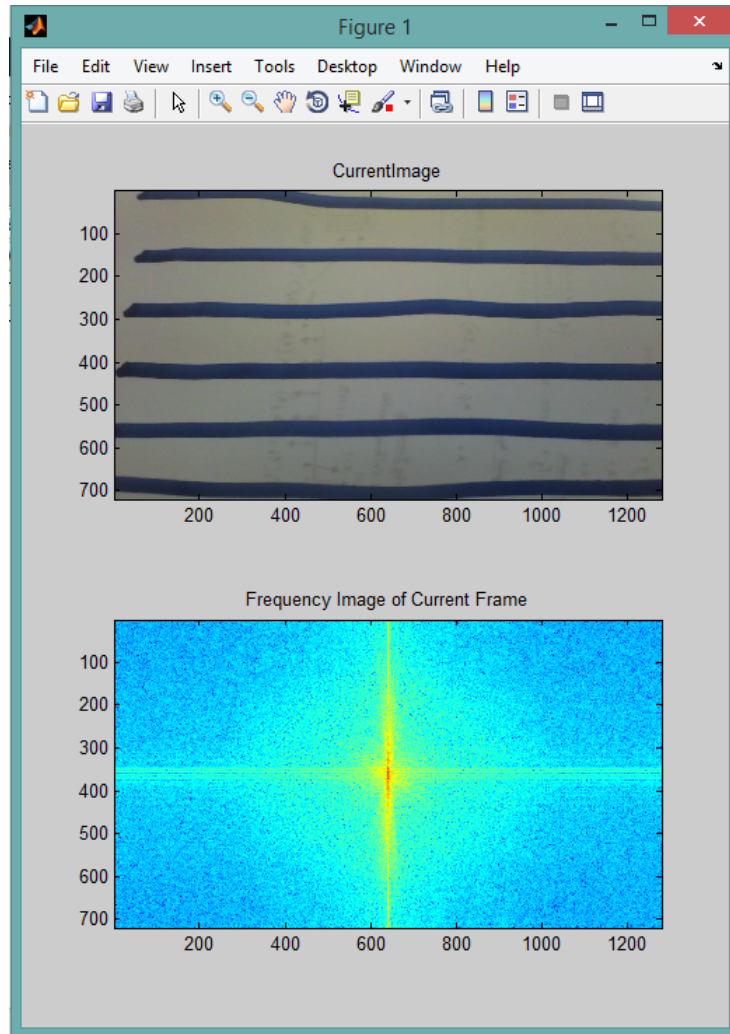


Figure 3: The figure shows FFT plot of horizontal pattern. As the pattern is quite near the camera, the frequency plot is concentrated around the center. This is because the pattern frequency is low. The horizontal lines produce high magnitude at phase $\frac{\pi}{2}$. This is the reason we see thick vertical line in the FFT plot.

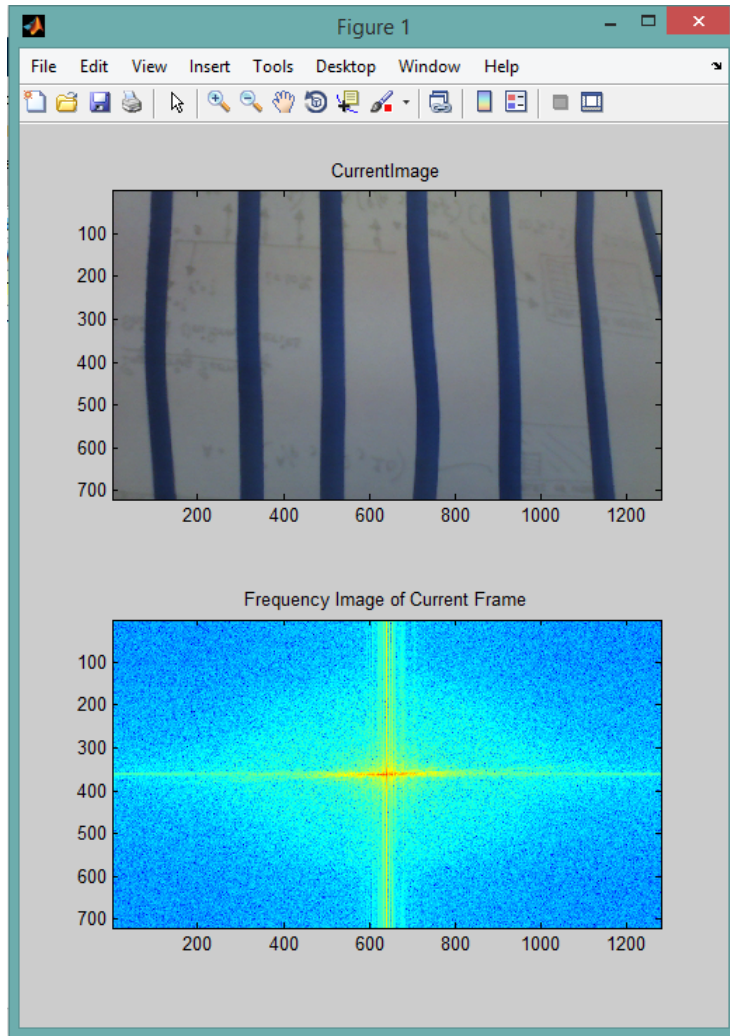


Figure 4: The figure shows FFT plot of vertical pattern. As the pattern is quite near the camera, the frequency plot is concentrated around the center. This is because the pattern frequency is low. The vertical lines produce high magnitude at phase $\phi = \pi$ and $\phi = 0$. This is the reason we see thick horizontal line in the FFT plot.

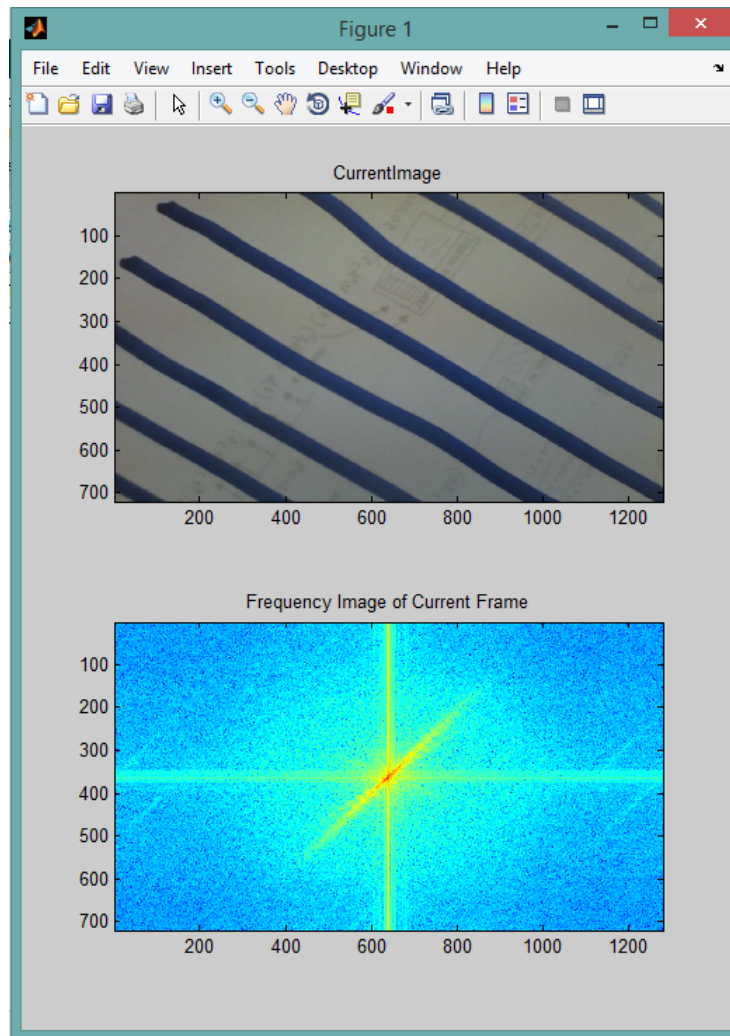


Figure 5: The figure shows FFT plot of slant pattern. As the pattern is quite near the camera, the frequency plot is concentrated around the center. This is because the pattern frequency is low. The slant lines produce high magnitude at phase $\approx \frac{\pi}{4}$. This is the reason we see thick line in the FFT plot at $\phi \approx \frac{\pi}{4}$.

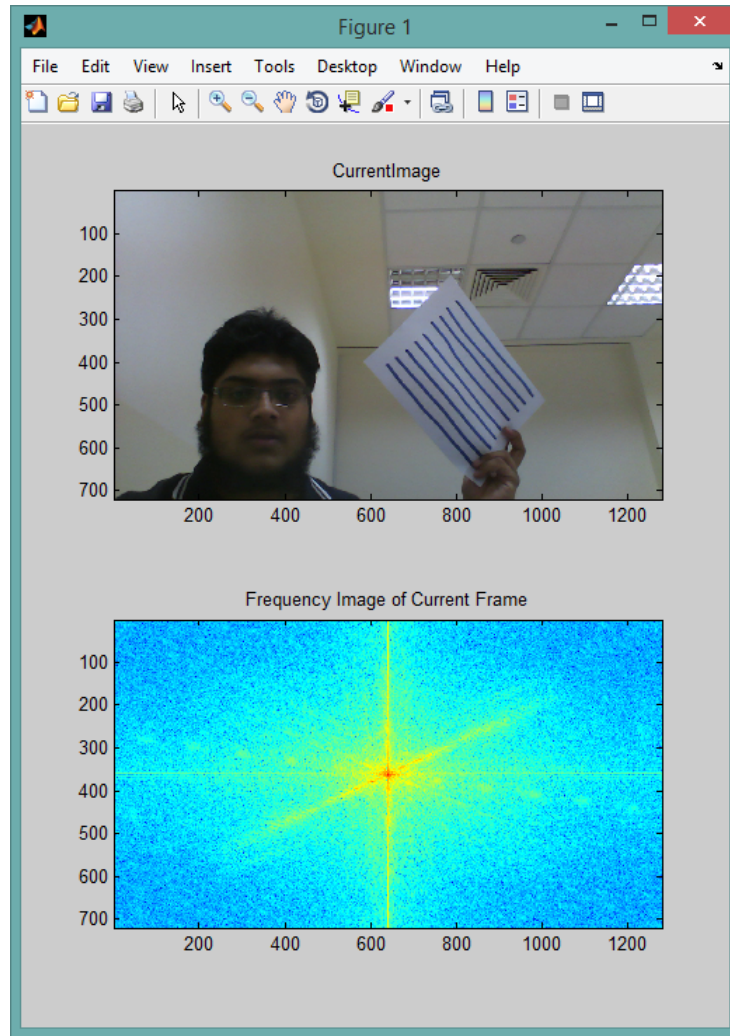


Figure 6: The figure shows FFT plot of slant pattern. As the pattern is quite far from the camera, the phase $\frac{\pi}{4}$ is quite stretched towards the higher frequency range. This is because the pattern frequency is high. The slant lines produce high magnitude at phase $\approx \frac{\pi}{4}$. This is the reason we see thick long line in the FFT plot at $\phi \approx \frac{\pi}{4}$.

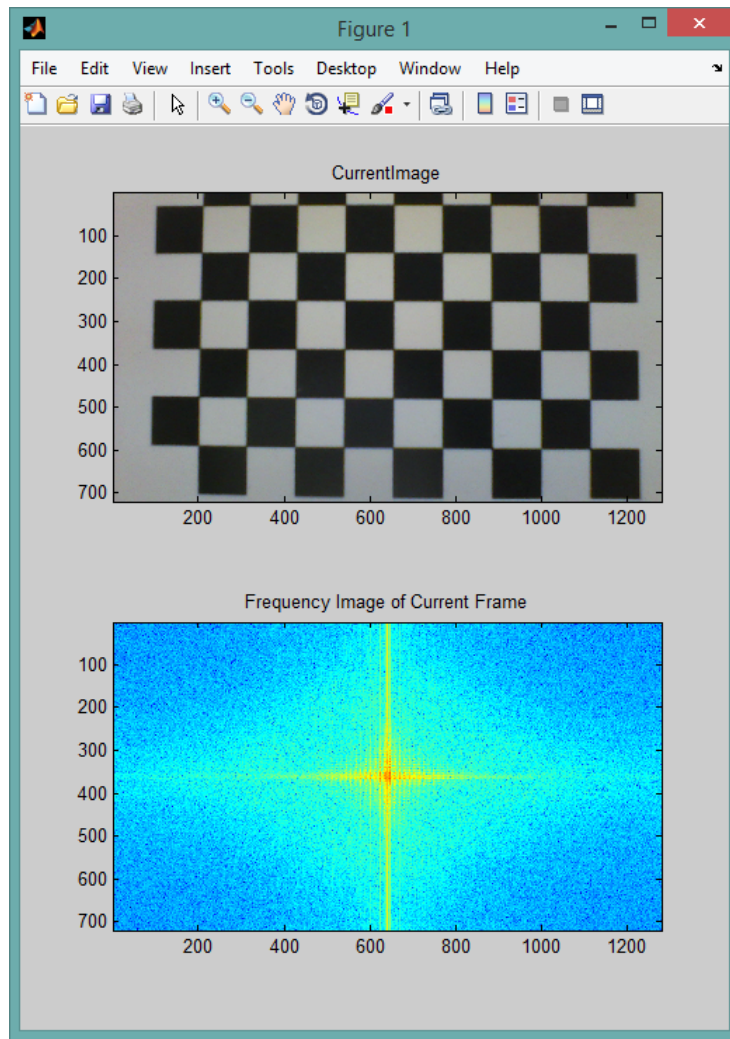


Figure 7: The figure shows FFT plot of check pattern. As the pattern is near the camera, the frequency domain peaks are near the center. This is because the pattern frequency is low. The check pattern produce box pattern in frequency domain. The box pattern as a whole has the phase of about $\frac{\pi}{2}$

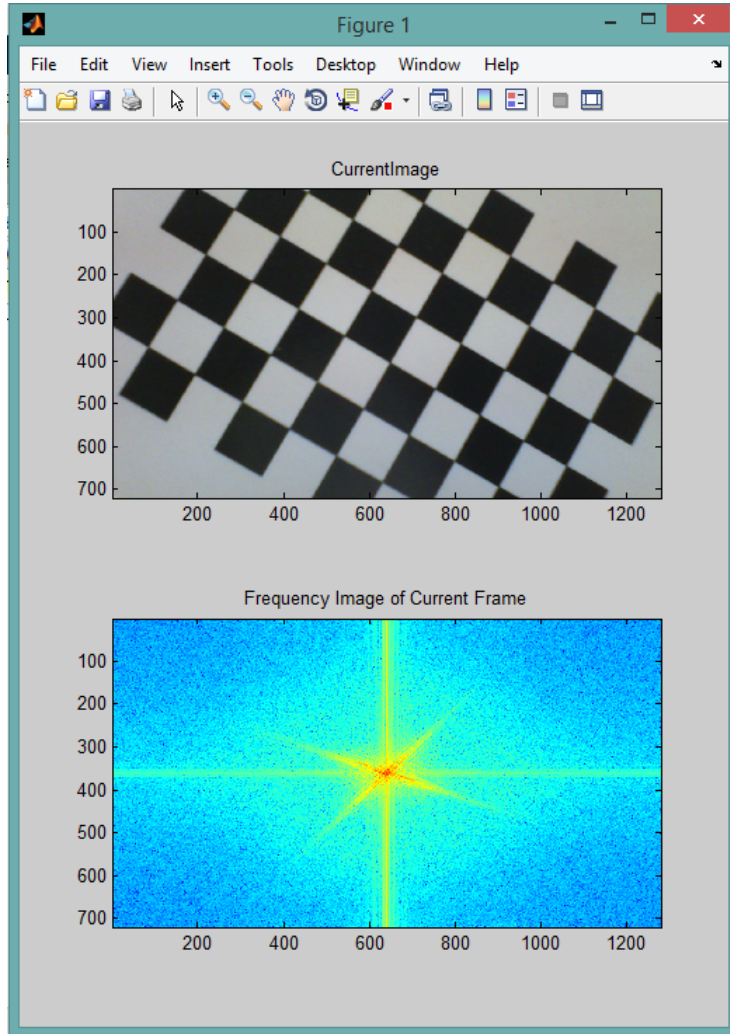


Figure 8: The figure shows FFT plot of check pattern. As the pattern is near the camera, the frequency domain peaks are near the center. This is because the pattern frequency is low. The check pattern produce box pattern in frequency domain. The box pattern as a whole has the phase of about $\frac{\pi}{4}$. That is why we see the boxy pattern rotated to $\frac{\pi}{4}$

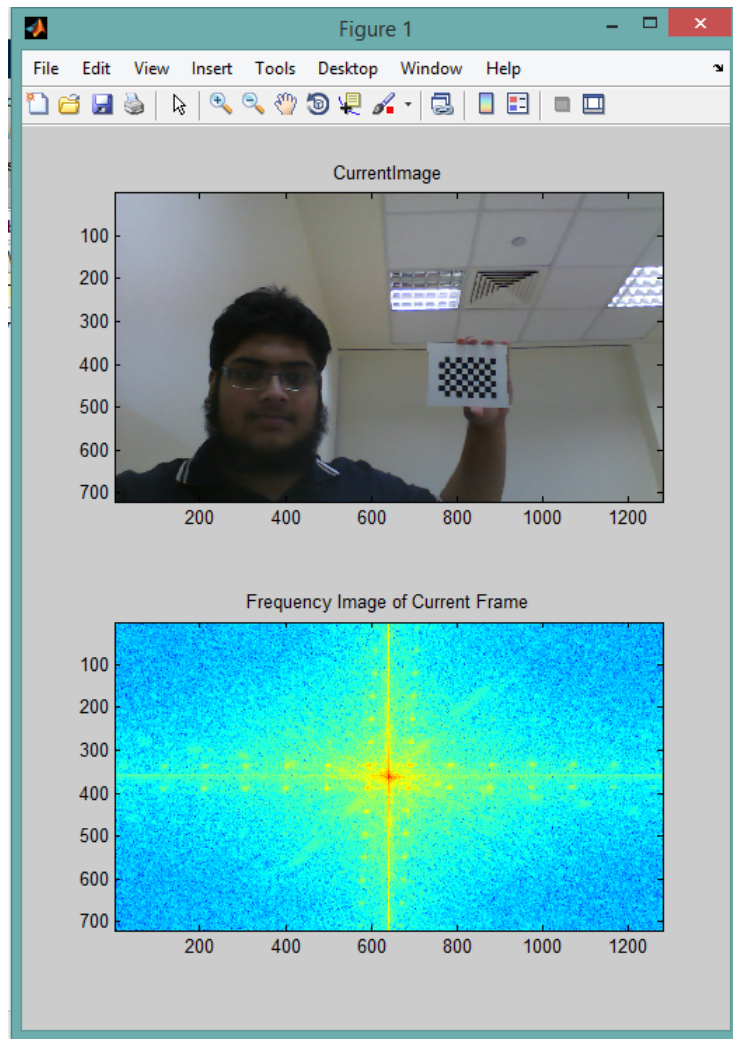


Figure 9: The figure shows FFT plot of check pattern. As the pattern is quite far from the camera, the frequency domain peaks are spread in the plot. This is because the pattern frequency is high. The check pattern produce box pattern in frequency domain. The box pattern as a whole has the phase of about $\frac{\pi}{2}$.

5 Conclusion

All in all, in this project we successfully implemented the live frame grab and 2 dimensional FFT plot. Additionally, we experimented with different patterns and confirmed the relationship between the frequency and spacial representation of images.

Moreover, if the pattern is taken far from the screen, the frequency spectrum spreads outwards near the π frequency value. On the other hand, if the pattern is brought closer to the screen the frequency comes near the center as the frequency is low. If the pattern is rotated, it reflects in the spectrum by phase shift.

References

- [1] Parallelizing Two-dimensional FFTs, *The Sequential 2D FFT Algorithm*, http://dsp-book.narod.ru/FFTBB/0270_PDF_C23.pdf, 2000 [9 June 2014].
- [2] Fourier Series and Transform, *Why Fourier transform*, <http://www.engr.uconn.edu/~lanbo/G377FFTYC.pdf> [9 June 2014].
- [3] What is a Matrix?, *A Video Screen is One Type of Matrix*, <http://cycling74.com/docs/max5/tutorials/jit-tut/jitterwhatisamatrix.html> [9 June 2014].