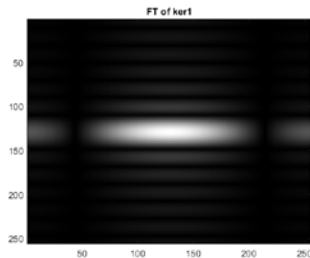
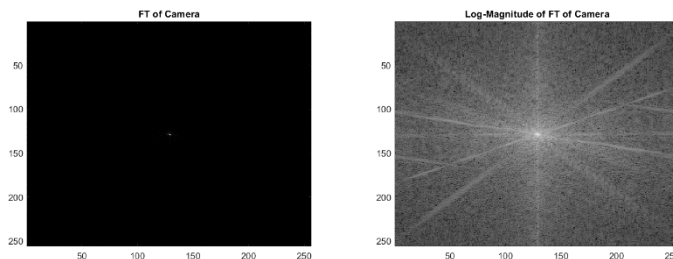


Project 3

1. The magnitude of the FFT of a uniform 13x3 kernel is shown below. The kernel changes value faster in the horizontal direction than it does in the vertical direction, so its FT contains more energy in the high frequency regions in the horizontal direction. It is non-zero length in both directions, so the FT contains low frequency content in both. This implies that an image convolved with this kernel would maintain low to medium frequency content horizontally and only low frequency content vertically.

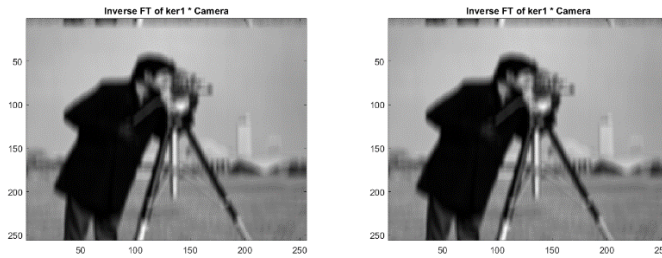


2. The FT magnitude of a 'normal' image such as the classic cameraman image, shown below, appears to show only low-frequency content, with effectively all of the energy clustered around the origin. Some line features become visible after dynamic range compression, shown below. The diagonal features correspond to things such as the legs of the camera and the edges of the cameraman's coat against the background, the vertical features to the horizon and edge of the grass, and the horizontal features to the central post of the tripod, buildings against the sky, and the cameraman's pant legs against the background.



3. The inverse FT of elementwise multiplication of the FT of the kernel and the FT of the image results in an image that appears to be smeared vertically, corresponding to averaging vertical sections of the original image. This is replicated by convolving the kernel with the original image, with a key difference being the convolution displaying a dark border due to zero padding and

the inverse FT shows smearing from periodic copies. Both are shown below.



4. On average, the FFT of a 1024x1024 image takes 0.017 seconds as measured by the attached code. As an FFT requires $M*N*\log(M*N)$ multiplications, my computer does approximately $8.55e08$ floating point operations per second. A DFT, requiring $M*M*N*N$ multiplications, would take $1.3e03$ seconds.

```
5. tic
6. for i = 1:100
7.     img = rand(1024, 1024);
8.     fft2(img);
9. end
10.    toc
```