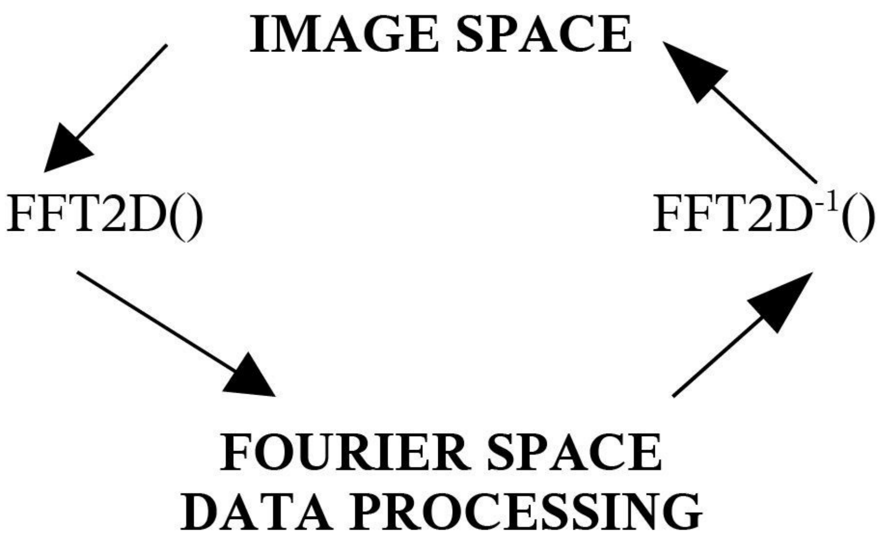
Fourier 2

Deepayan Bhowmik

Fourier: Filtering



Fourier: Filtering

Enables complex operations in image space to be computed as simple operations in Fourier Space such as:

- Convolution.
- Correlation.
- De-convolution (beyond scope of module).

Simpler operations in Fourier space:

- Computationally more efficient.
- Manipulation of spatial image content as coefficient in Fourier representation.

Fourier: Filtering

Recall Spatial Filtering lecture …

$$I_{output}(i,j)=\sum_{k=1}^{N}\sum_{l=1}^{M} I_{input}(i-(N-1)+k,\, j-(M-1)+l)\,m_{kl}$$

*

1	1	1
1	−8	1
1	1	1

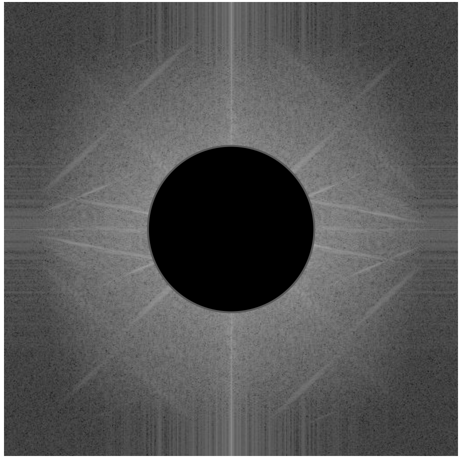
=

… where we had to perform spatial convolution (with a Laplacian kernel) to find the edges in the image. (what a lot of work!)

Fourier: Filtering

We just turn off all the low-frequency coefficients within the DFT representation of the image ….

DTF



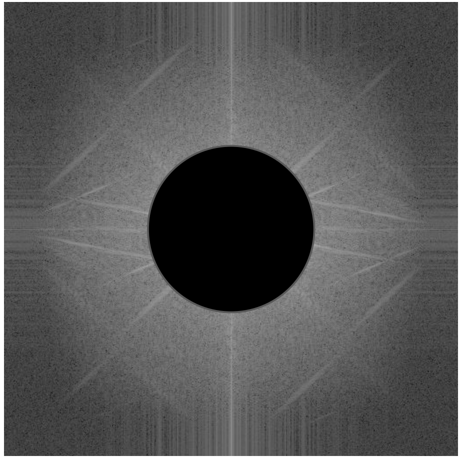
DTF^{-1}

… by multiplying them by 0, to keep the high frequency edges.

Fourier: Filtering

Multiplication in the **frequency domain** reduces to **convolution** in the **spatial domain**!

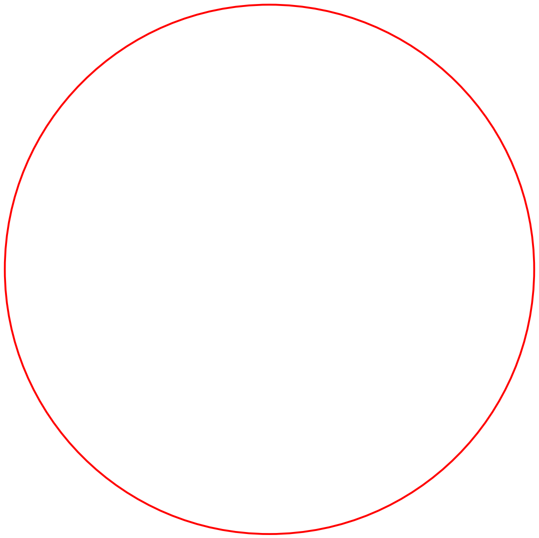
DTF



DTF^{-1}

Fourier: Filtering

Equation of a circle: $x^2 + y^2 = r^2$



Fourier: High Pass Filtering

Operation: remove low frequency components from the image

- Sets frequencies below a certain threshold to zero

Ideal high pass filter:

$$H(k_x, k_y) = \begin{cases} 0, & \sqrt{k_x^2 + k_y^2} \leq K \\ 1, & \sqrt{k_x^2 + k_y^2} > K \end{cases}$$

- *K* = cut-off distance (radius) from the fourier image origin.
- Defines a radius around the centre of the Fourier image.

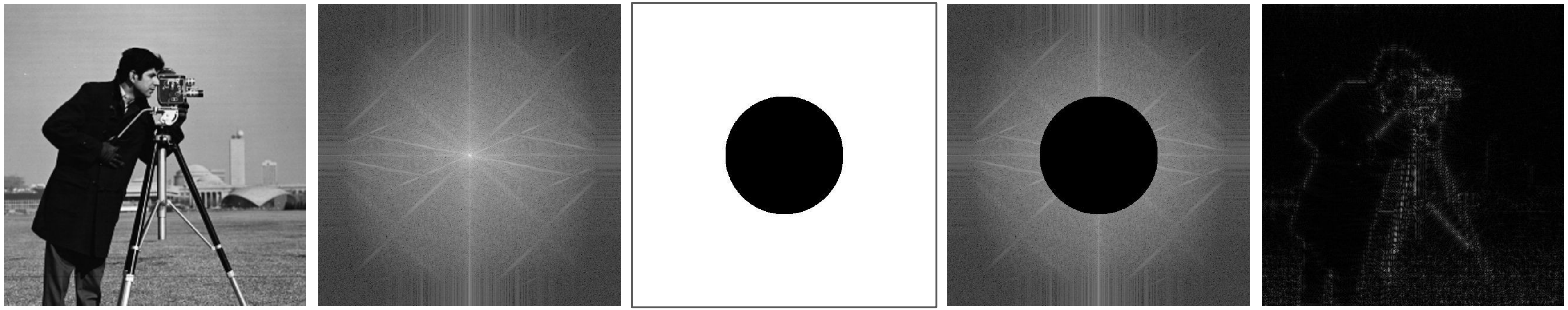
- Essentially multiplication of Fourier image *F()*, by $H(k_x,k_y)$:

$$S(k_x,k_y) = H(k_x,k_y)\ F(k_x,k_y)$$

Uses: detect edges / edge enhancement.

Fourier: High Pass Filtering

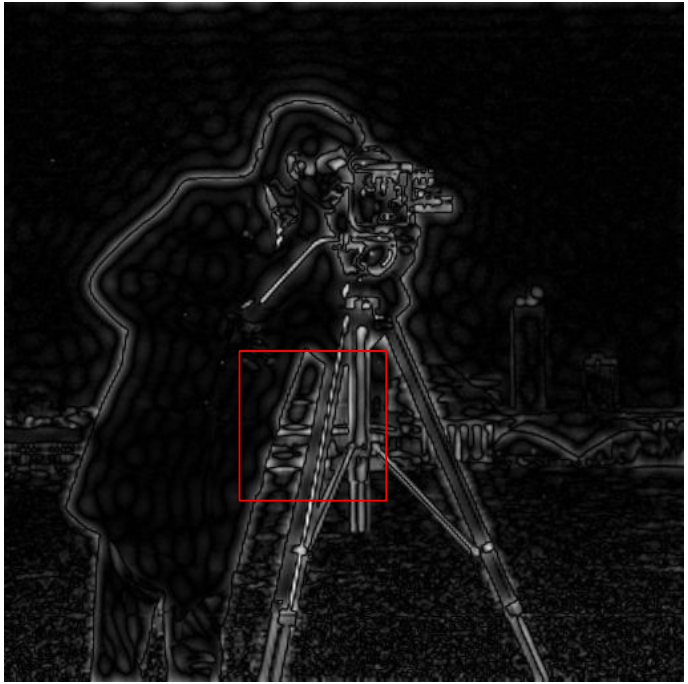
DFT() $H(k_x,k_y)$ x $F(k_x,k_y)$ *DFT*$^{-1}$*()*



$$H(k_x, k_y) = \begin{cases} 0, & \sqrt{k_x^2 + k_y^2} \leq K \\ 1, & \sqrt{k_x^2 + k_y^2} > K \end{cases}$$

Fourier: High Pass Filtering

We observe also high frequencies where they do not exist in the corresponding spatial images …..



Known as “ringing” (i.e. processing noise), introduced by the sharp {0 → 1} cut-off of frequencies in the Fourier domain.

Fourier: Butterworth High Pass Filtering

Operation: approximation to the ideal high pass filter:

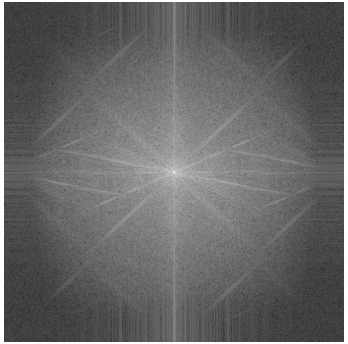
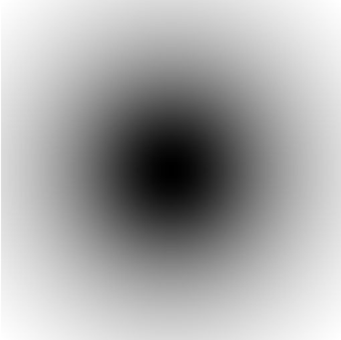
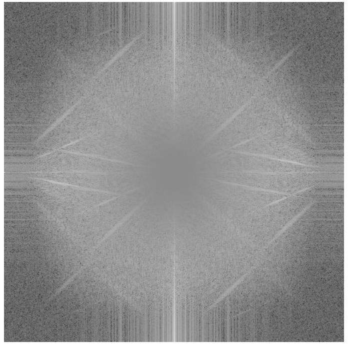
- Why approximate ? - to avoid “ringing” effects.
- A piecewise continuous circularly symmetric filter given by:

$$B(k_x, k_y) = \frac{1}{1 + \left(\frac{K}{\sqrt{k_x^2 + k_y^2}}\right)^{2n}}$$

- Cut-off value K (as before).
- n = user-defined positive integer called the order of the filter:
  - As n increases, the filter approaches the ideal filter.
  - (defines the gradient of the cut-off).

Fourier: Butterworth High Pass Filtering

DFT() *$B(k_x,k_y)$* x *$F(k_x,k_y)$* *$DFT^{-1}()$*


$$B(k_x, k_y) = \frac{1}{1 + \left(\frac{K}{\sqrt{k_x^2 + k_y^2}}\right)^{2n}}$$

Fourier: Low Pass Filtering

Operation: remove high frequency components from the image:

- Dets frequencies above a certain threshold to zero:

Ideal low pass filter:

$$H(k_x, k_y) = \begin{cases} 1, & \sqrt{k_x^2 + k_y^2} \leq K \\ 0, & \sqrt{k_x^2 + k_y^2} > K \end{cases}$$

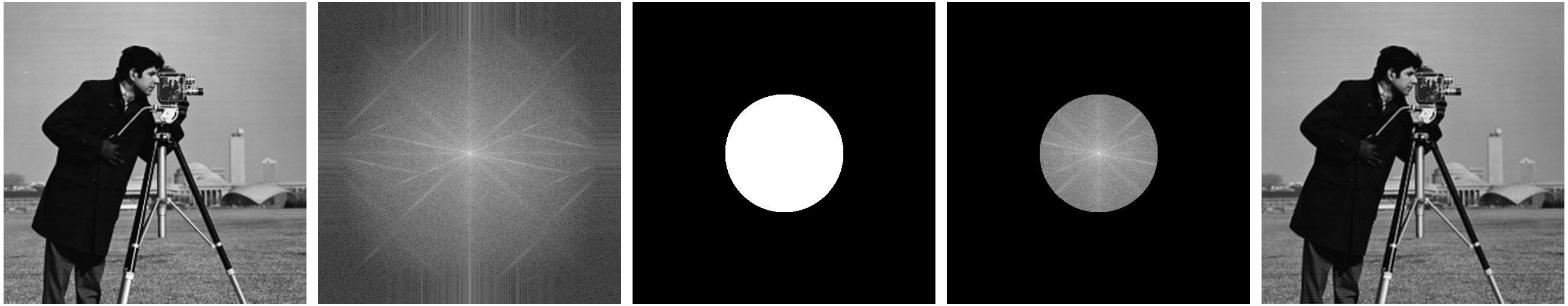
- K = cut-off distance (radius) from the image origin
    - Defines a radius around the centre of the Fourier image

- Multiplication of Fourier image *F()*, by $H(k_x,k_y)$:

$$S(k_x,k_y) = H(k_x,k_y)\, F(k_x,k_y)$$

Uses: noise removal / image smoothing.

Fourier: Low Pass Filtering

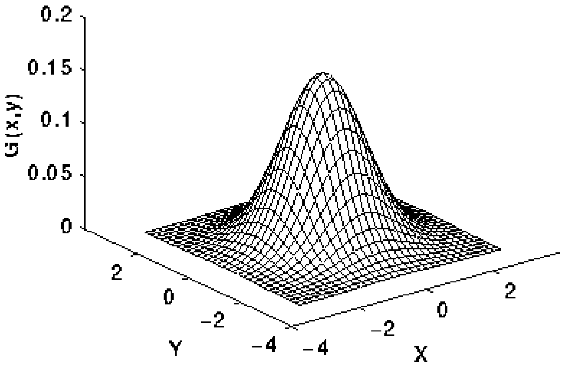


$$H(k_x, k_y) = \begin{cases} 1, & \sqrt{k_x^2 + k_y^2} \leq K \\ 0, & \sqrt{k_x^2 + k_y^2} > K \end{cases}$$

Fourier: Gaussian Low Pass Filtering

$$G(k_x, k_y) = \exp\left[-(k_x^2 + k_y^2)/2\sigma^2\right]$$

- Where σ is the width of the filter at 1/e.
- Adjusting σ adjusts the bandwidth of this filter.
  - i.e. the range of frequencies allowed through.

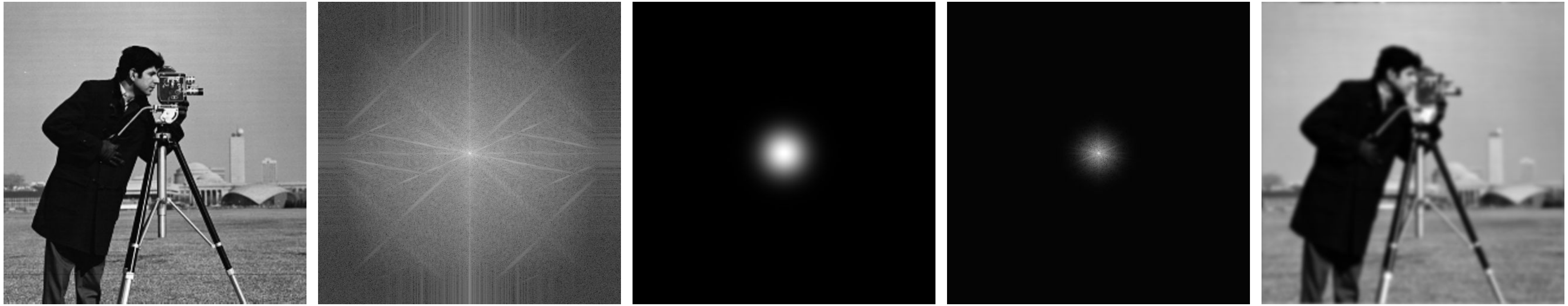


$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

- Piecewise continuous function in the spectral domain.

Fourier: Gaussian Low Pass Filtering

DFT() — $G(k_x,k_y)$ x $F(k_x,k_y)$ — $DFT^{-1}()$



$$G(k_x, k_y) = \exp\left[-(k_x^2 + k_y^2)/2\sigma^2\right]$$

Fourier: Butterworth Low Pass Filtering

Alternative to Gaussian

Operation: approximation to the ideal low pass filter:

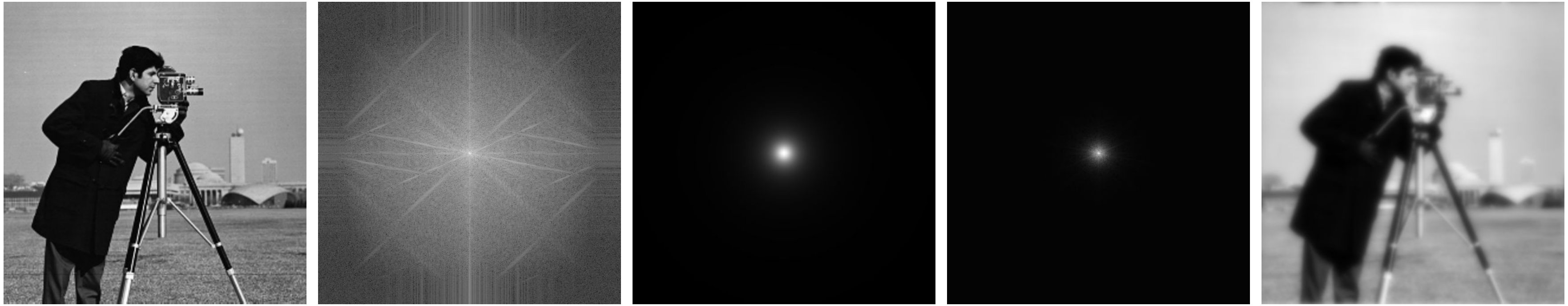
- Piecewise continuous function in the spectral domain:

$$B(k_x, k_y) = \frac{1}{1 + \left(\frac{\sqrt{k_x^2 + k_y^2}}{K}\right)^{2n}}$$

- cut-off value K (as before).
- n = user-defined positive integer called the order of the filter.
    - as n increases, the filter approaches the ideal filter but the computational cost similarly increases!

Fourier: Butterworth High Pass Filtering

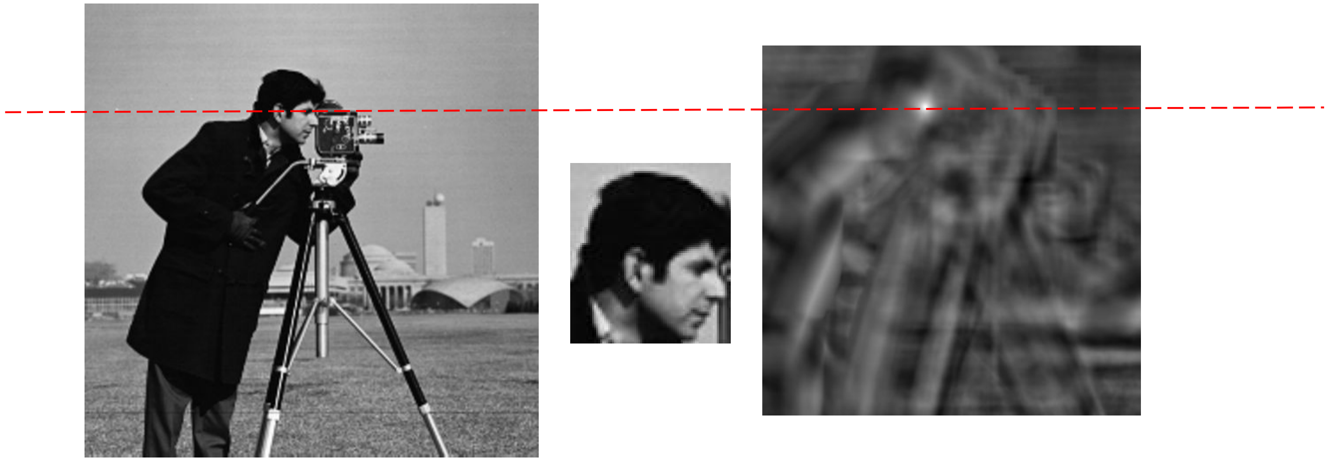
DFT() *$B(k_x,k_y)$* x *$F(k_x,k_y)$* *$DFT^{-1}()$*



$$B(k_x, k_y) = \frac{1}{1 + \left(\frac{\sqrt{k_x^2 + k_y^2}}{K}\right)^{2n}}$$

Fourier: Correlation

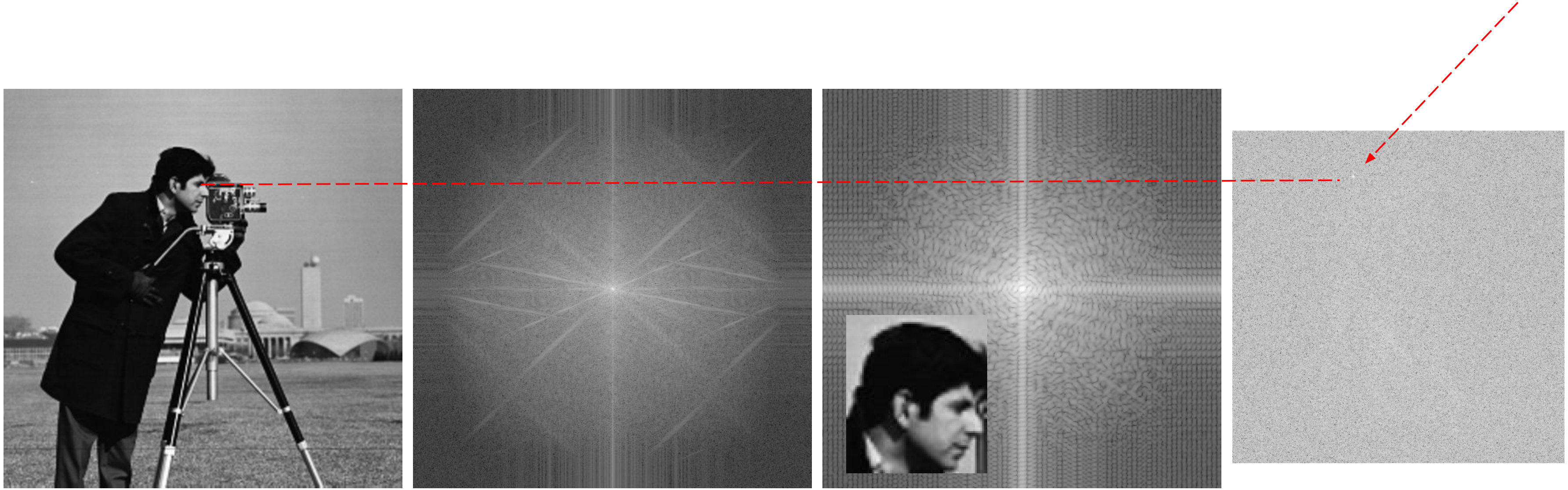
Operation: find all the regions with certain properties or patterns.



Fourier: Correlation

Process (in Fourier): transform image and mask into Fourier space and multiply them together:

- Frequencies in mask remain (amplified), whilst others set zero(attenuated).



Fourier: Deconvolution

De-convolution: remove the effects of a given filter function

Dividing the image by the filter mask in Fourier Space to perform deconvolution

- Inverse of convolution, used as basis for image deblurring.

$$I_{dec} = DFT^{-1}(\ DFT(I) \ / \ DTF(K)\ )$$

Although inverse filtering is efficient, it is usually problematic when the values of *DFT(K)* are small, i.e. the convolution kernel has zeros in the Fourier domain. Divisions by zero or values close to zero will severely amplify noise.

Fourier: Wiener Deconvolution

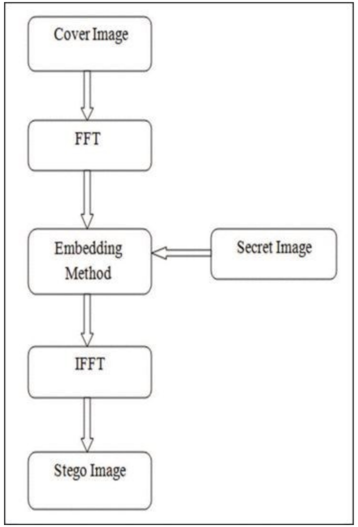
Wiener filtering applied to the deconvolution problem adds a damping factor.

$$I_{dec} = DFT^{-1}\left(\left(|DFT(K)|^2 / \left(|DFT(K)|^2 + 1 / SNR\right)\right) * DFT(I) / DFT(K) \right)$$



Applications

A new image steganography technique using Fast Fourier Transform on the cover image. The secret image is embedded into the two components of the cover image formed after applying the FFT on the cover image using the Least Significant Bit substitution method.



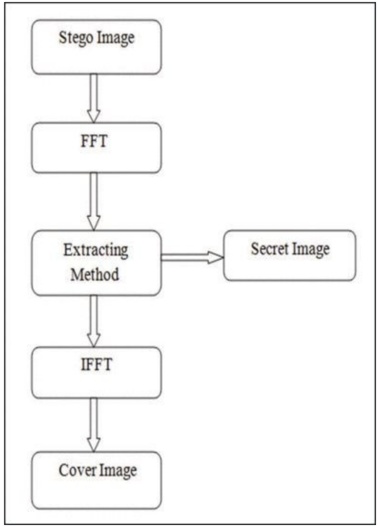
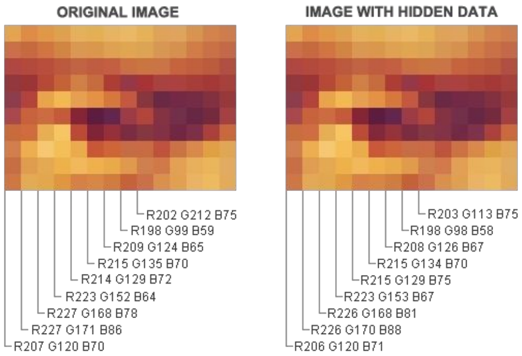


Image	Image Size	PVD	GLM	PMM	LSB	Proposed Method
Lena	128*128	**	2048	2393	2048	4096
	256*256	**	8192	10012	8192	16384
	512*512	50960	32768	45340	32768	65536
Pepper	128*128	**	2048	2860	2048	4096
	256*256	**	8192	11694	8192	16384
	512*512	50685	32768	46592	32768	65536

(**All the images used for PVD are of the size 512*512)



A. Shaukat, M. Chaurasia and G. Sanyal, "A novel image steganographic technique using fast fourier transform," 2016 International Conference on Recent Trends in Information Technology, Chennai, 2016, 1-6.