MRT Image Processing II Report

Suketu Patni, 23B1299

2 January, 2024

1 Fourier Transforms

1. It allows us to express any periodic function as a weighted sum of sinusoids of varying frequencies and phases. It takes a function in the spatial (or time) domain (x) and converts it into two functions (one of amplitude and other of phase) in the frequency domain (u).

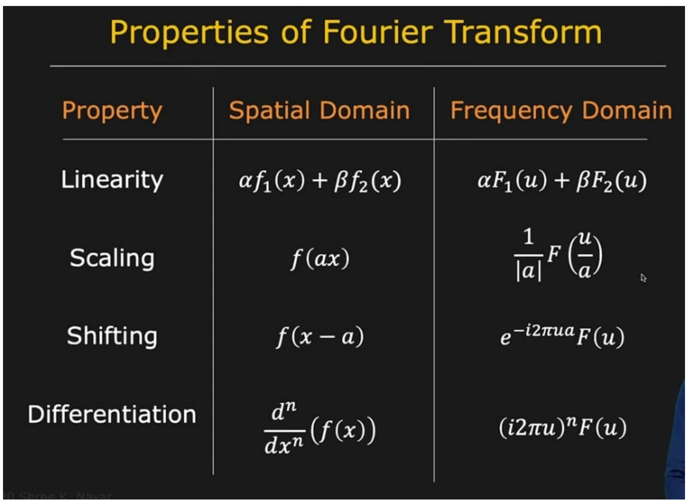
2. Fourier transform:
$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi iux}dx \tag{1}$$

3. Inverse Fourier transform:
$$f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi iux}du \tag{2}$$

4. We can recover the amplitude A and phase ψ of the sinusoid with frequency u from $F(u)$ as $A(u) = |F(u)|$ and $\psi(u) = arg(F(u))$

$f(x)$	$F(u)$
$cos(2\pi kx)$	$\frac{1}{2}(\delta(u+k) + \delta(u-k))$
$sin(2\pi kx)$	$\frac{i}{2}(\delta(u+k) - \delta(u-k))$
1	$\delta(u)$
$\delta(x)$	1
$Rect(\frac{x}{T})$	$Tsinc(Tu))$
e^{-ax^2}	$(\sqrt{\pi/a})e^{-\pi^2u^2/a}$

5. Properties:



Properties of Fourier Transform

Property	Spatial Domain	Frequency Domain
Linearity	$\alpha f_1(x) + \beta f_2(x)$	$\alpha F_1(u) + \beta F_2(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a\rvert}F\left(\frac{u}{a}\right)$
Shifting	$f(x-a)$	$e^{-i2\pi ua}F(u)$
Differentiation	$\frac{d^n}{dx^n}(f(x))$	$(i2\pi u)^nF(u)$

2 Convolution Theorem

1. $$g(x) = f(x) * h(x) \Longleftrightarrow G(u) = F(u)H(u) \tag{3}$$

2. $$G(u) = F(u) * H(u) \Longleftrightarrow g(x) = f(x)h(x) \tag{4}$$

3. Often it is computationally cheaper to take FTs of f and h then take IFT of that to get g instead of convolving it directly as there are very fast algorithms to perform both FTs and IFTs.

4. For instance, noise from a signal can be removed by convolving it with a Gaussian (a "fuzzy filter" for signals).

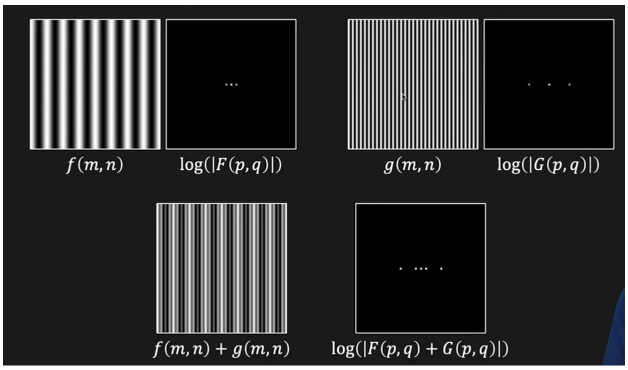
3 Image Filtering in Frequency Domain

1. 2D FTs: (u and v are frequencies along x and y respectively)
$$F(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)e^{-2\pi i(ux+vy)}dxdy \tag{5}$$

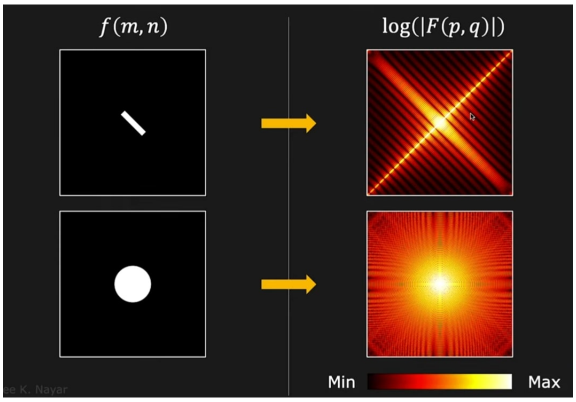
2. 2D IFTs:
$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u, v)e^{2\pi i(ux+vy)}dxdy \tag{6}$$

3. 2D DFTs: (p and q are frequencies along m and n respectively)
$$F[p, q] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m, n]e^{-2\pi i(pm/M+qn/N)} \tag{7}$$
where $p = 0, 1, 2, ..., M - 1$ and $q = 0, 1, 2, ..., N - 1$

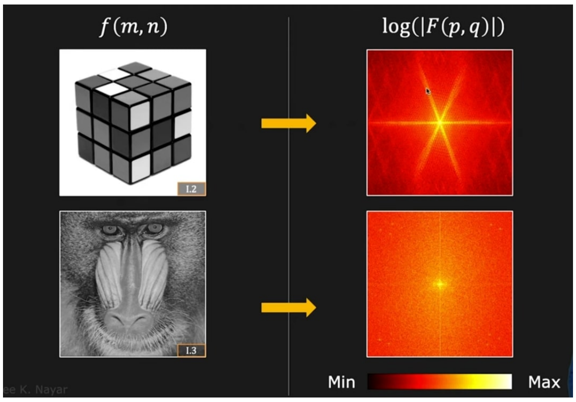
4. 2D IDFTs:
$$f[m, n] = \frac{1}{MN}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} F[p, q]e^{2\pi i(pm/M+qn/N)} \tag{8}$$
where $m = 0, 1, 2, ..., M - 1$ and $n = 0, 1, 2, ..., N - 1$



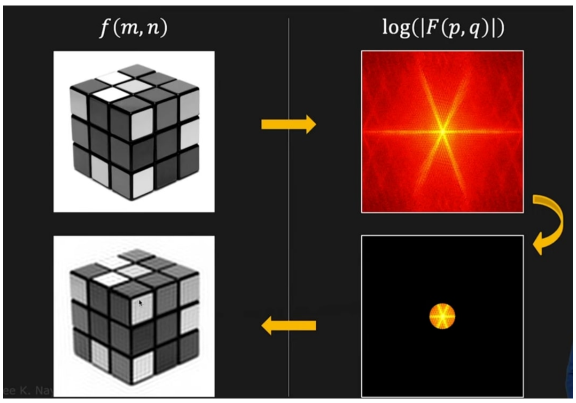
5. Example 1
For now, consider plotting only the magnitude of the FT of the image (log for bringing values down to required range).
First 2 images are simple cosines on x axis, leading to 2 frequencies on either side of the central dot (which is formed due to there being a DC component in all pixel values, say, 128, as they range from 0 to 255). Their addition leads to 5 dots as there are now 2 frequencies.



6. Example 2
   There are very strong frequencies along the $y = x$ diagonal (to recreate the sharp edges in the bar perpendicular to that diagonal) and some high frequencies along the other diagonal (as there the edges are less sharp).
   For the disc, the FT is rotationally symmetric and there are some high frequencies along every direction to create the edge of the disc.
7. Example 3
   For the cube, high frequencies are along 3 main directions as there are edges along 3 main directions. For the mandarin, and any other image in general, it is very difficult to extract any information from the FT visually.

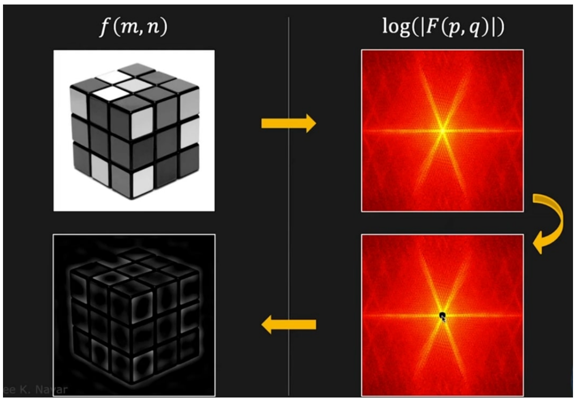


8. Noise usually resides in the higher frequencies.
9. Low Pass filtering
   We truncate the FT, allowing only the high frequencies in the center to remain, and then take the



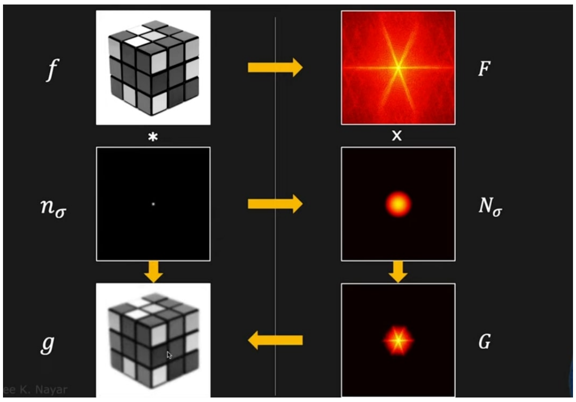
IFT. Blocky artifacts and blurriness increase as the filter becomes harsher and harsher.

10. High Pass filtering
We puncture the FT, allowing only the low frequencies in the non-center to remain, and then take



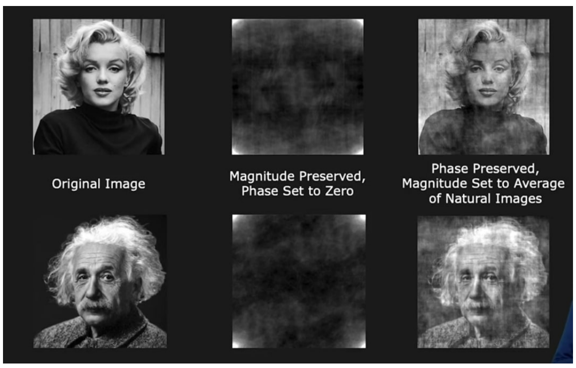
the IFT. As the filter becomes harsher and harsher, only the edges and corners and regions where a drastic transition occurs remain. This may be useful in edge detection.

11. Gaussian Smoothing



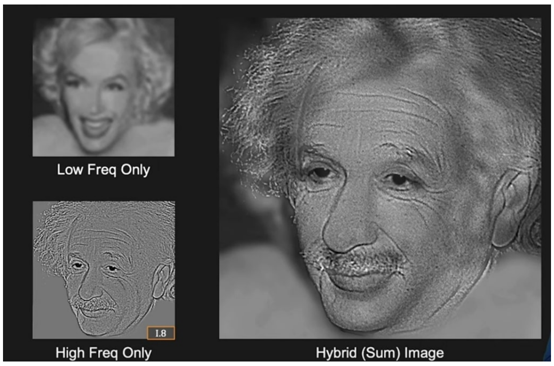
Instead of convolving it with the kernel directly, we may follow the above process. Indeed, as the kernel gets broader, the end result gets smoother and blurrier.

12. Phase of the FT



The phase is often more important to the information in an image than the magnitude of the FT.

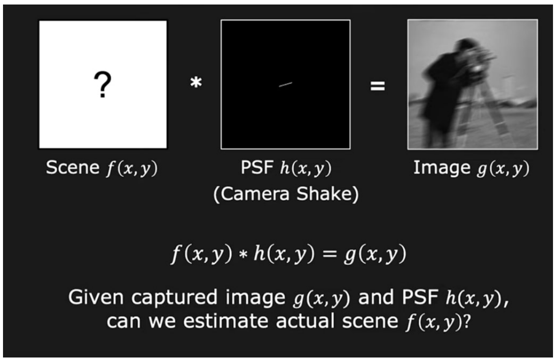
13. Hybrid Images



From close, we are able to see Einstein because we can see the all the details and edges (high frequencies). But from afar, because the eye has its own PSF, we only see the low frequencies, seeing Monroe instead.

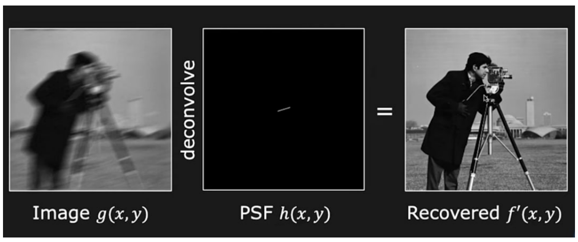
4 Deconvolution

1. Sometimes, the image might get convolved inadvertently by a function (for instance, camera is shaken during a photo). Deconvolution helps us undo that convolution.



2. If the recovered scene is $f'(x, y)$, then

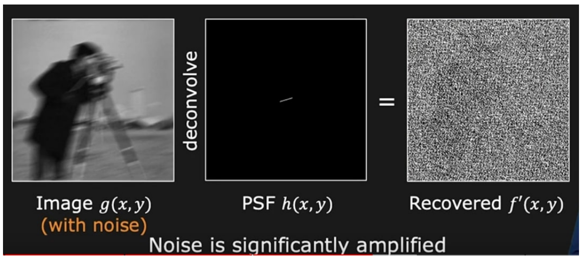
$$f'(x, y) = IFT\left(\frac{G(u, v)}{H(u, v)}\right) \tag{9}$$



3. Issues:
   - If $H(u, v) = 0$, $F'(u, v)$ is not recoverable.
   - Noise:

     $$g(x, y) = f(x, y) * h(x, y) + \eta(x, y)$$

     The motion blur filter $H(u, v)$ is a low pass filter as it blurs the edges and the image. So, for high frequencies u and v, the noise (if any) will be high (as noise usually comes from high frequencies) and $H(u, v) \approx 0 \Rightarrow$ the noise will be amplified heavily.



4. Weiner Deconvolution (Noise Suppression)

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \left(\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right) \tag{10}$$

where NSR is the ratio of the power of the noise and the signal at (u, v) i.e.

$$NSR(u,v) = \frac{|N(u,v)|^2}{|F(u,v)|^2} \tag{11}$$

5. If either the noise is high or $H(u, v)$ is low, the Weiner factor becomes very small, and $F(u, v)$ is attenuated.
6. However, we don't actually know the noise or the original scene. it suffices very well to set $NSR(u, v) = \lambda$ for a lot of purposes.