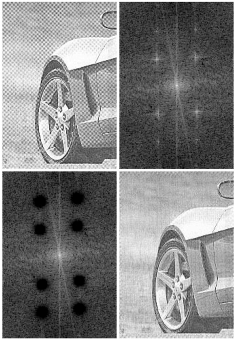
COMS20011 – Data-Driven Computer Science

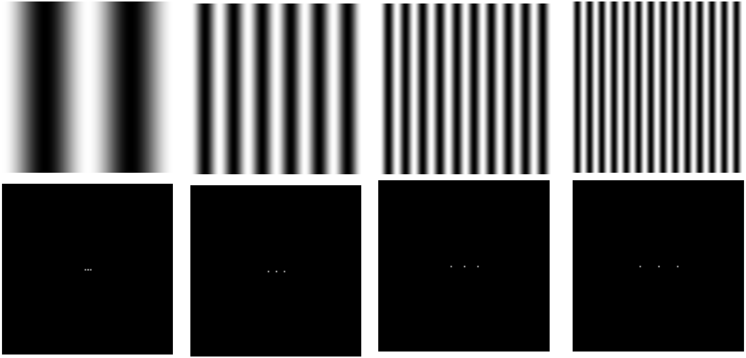


Frequency Domain Fundamentals
(and Frequencies as Features)

March 2023

Majid Mirmehdi

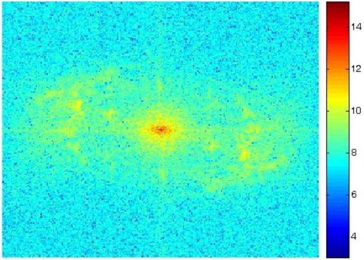
Next in DDCS



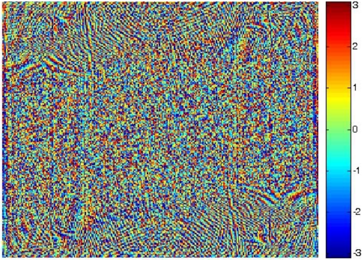
Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and **2D Fourier Transform**
- Another look at features
- Convolutions

Viewing Magnitude and Phase - *reminder*



$\log(|F(I)| + 1)$



$\varphi[\mathrm{F}(I)]$

Periodic Spectrum

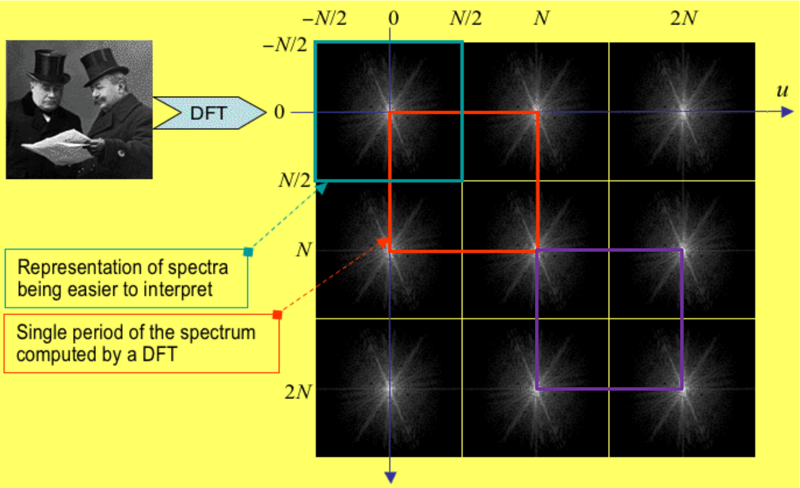


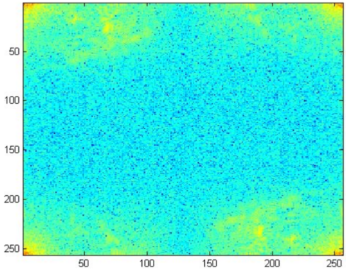
Illustration from Václav Hlaváč

Symmetry

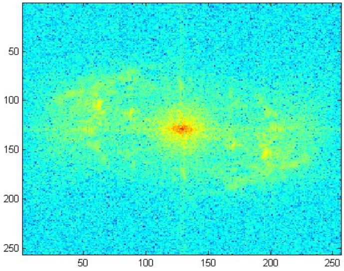
Important property of the FT: *Conjugate Symmetry*

The FT of a real function f(x,y) gives:

$$F(u,v) = F^*(-u,-v) \Rightarrow |F(u,v)| = |F^*(-u,-v)|$$



Before shift

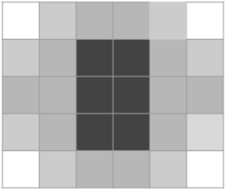


After shift

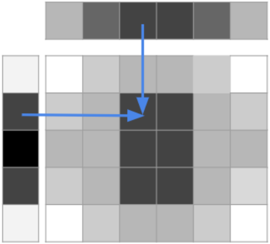
Separability

Important property of the FT: *Separability*

If a 2D transform is separable, the result can be found by successive application of two 1D transforms.



Original matrix: MxN



Separable: Mx1 * 1xN

Faster Computation: convolving an $M\ x\ N$ image with an $m\ x\ n$ filter would require $\mathcal{O}(M.N.m.n)$ operations.
In 1D separable form, only $\Rightarrow \mathcal{O}(M.N.(m+n))$

Separability

Important property of the FT: *Separability*

If a 2D transform is separable, the result can be found by successive application of two 1D transforms. This is a principle aspect of the Fast Fourier Transform (FFT).

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v) \quad e^{\frac{-j2\pi ux}{N}} \quad \text{where} \quad F(x,v) = \frac{1}{N}\sum_{y=0}^{N-1} f(x,y) \quad e^{\frac{-j2\pi vy}{N}}$$

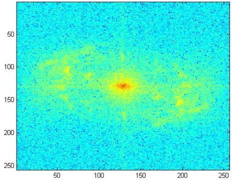
$$f(x,y) \xrightarrow[\text{1-D row transforms}]{} F(x,v) \xrightarrow[\text{1-D column transforms}]{} F(u,v)$$

Rotation

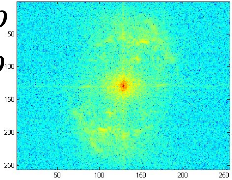
Important property of the FT: *Rotation*

Rotate the image and the Fourier space rotates.

$$x = r\cos\theta$$
$$y = r\sin\theta$$



$$u = \omega\cos\varphi$$
$$v = \omega\sin\varphi$$

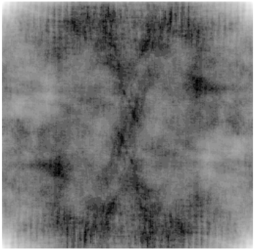


$$f(r, \theta + \rho_0) \quad \Longrightarrow \quad F(\omega, \varphi + \rho_0)$$

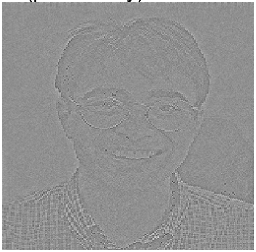
Importance of Phase

Andrew

ifft(mag only)



ifft(phase only)

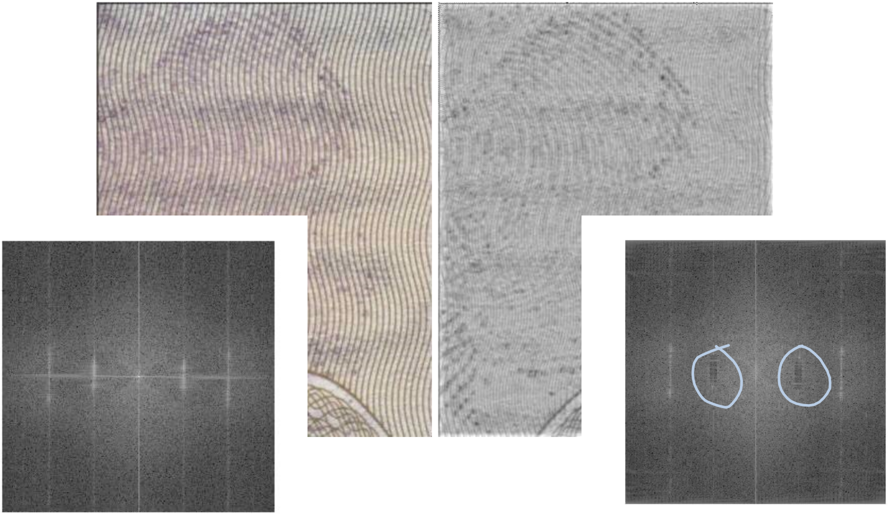


Peter

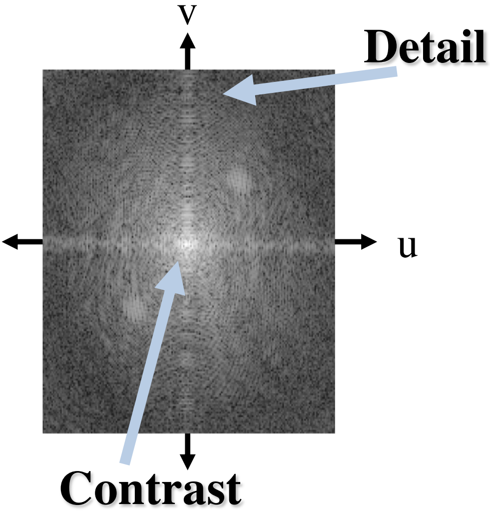
ifft(mag(Peter) and phase(Andrew))

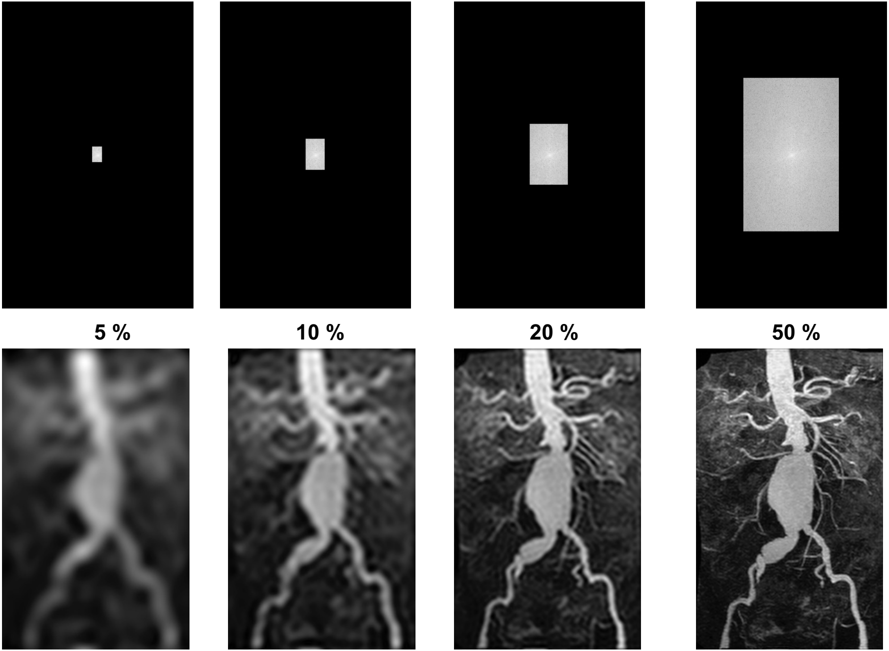
ifft(mag(Andrew) and phase(Peter))

Changing the frequency values! By hand!



Manipulating the Fourier Frequencies

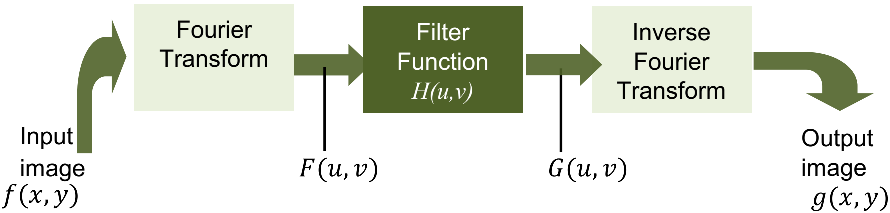



5 %
10 %
20 %
50 %

Filtering the Fourier Frequencies

Filtering → to manipulate the (signal/image/etc) data.

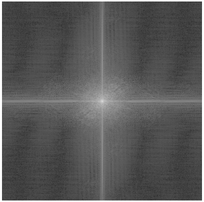
1D: $G(u) = F(u)H(u)$ 2D: $G(u,v) = F(u,v)H(u,v)$

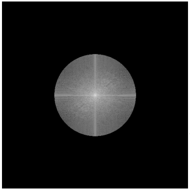


Low Pass Filtering

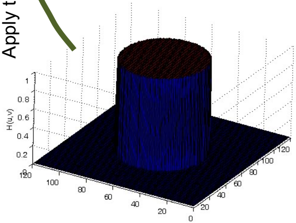
1D: turning the “treble” down on audio equipment!

2D: smooth image





Apply to freq. domain

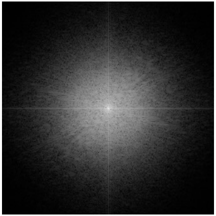


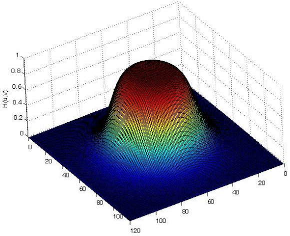
$$H(u,v) = \begin{cases} 1 & r(u,v) \le r_0 \\ 0 & r(u,v) > r_0 \end{cases}$$

$r(u,v) = \sqrt{u^2 + v^2}$, r_0 is the filter radius

Butterworth's Low Pass Filter

After applying filter to freq. domain



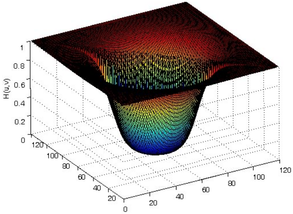


$$H(u,v) = \frac{1}{1+[r(u,v)/r_0]^{2n}} \quad \text{of order } n$$

Butterworth's High Pass Filter

1D: turning the bass down on audio equipment!

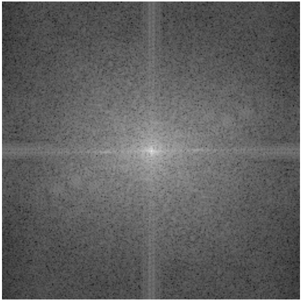
2D: sharpen image



Order of $n=3$

$$H(u,v) = \frac{1}{1 + [r_0/r(u,v)]^{2n}}$$ of order n

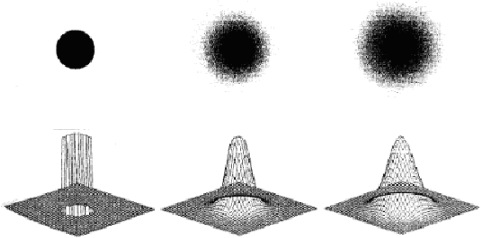
Butterworth’s Low and High Pass Filtering Examples



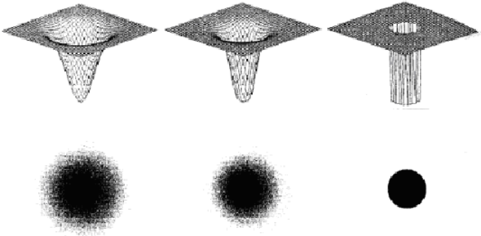
Low Pass

High Pass

Low and High Pass Filtering

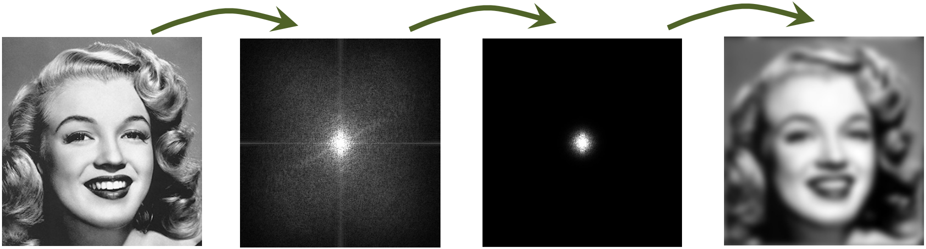


Basic Butterworth Gaussian

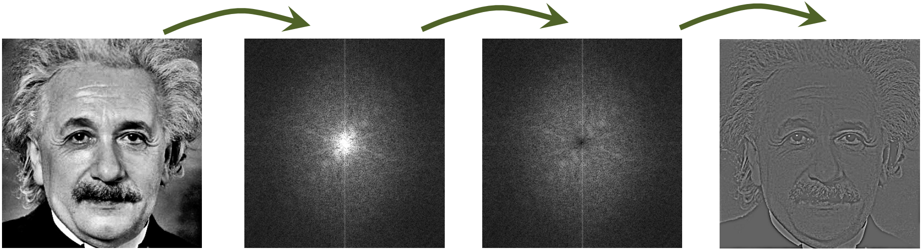


Gaussian Butterworth Basic

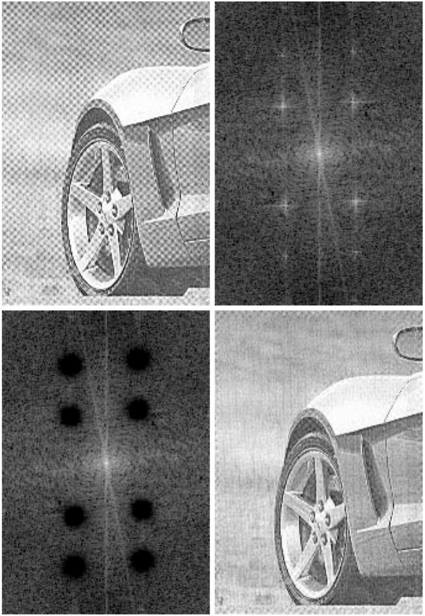
Filtering Examples



$log(|F(I)| + 1)$ Filtered



Filtering to Remove Periodic Noise

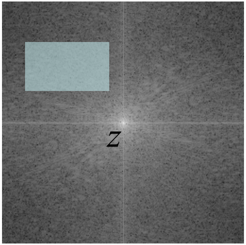


$log(|F(I)| + 1)$

Butterworth's Notch Reject Filtered

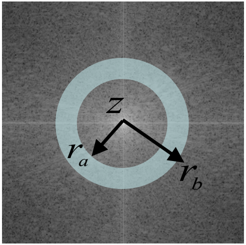
Spectral Features from Spectral Regions

- Fourier space, with origin at $z=(u=0,v=0)$.



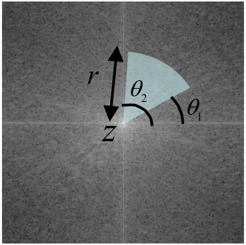
$$a \le u \le b$$
$$c \le v \le d$$

box



$$-r_b \le u \le r_b$$
$$\pm\sqrt{r_a^2 - u^2} \le v \le \pm\sqrt{r_b^2 - u^2}$$

ring

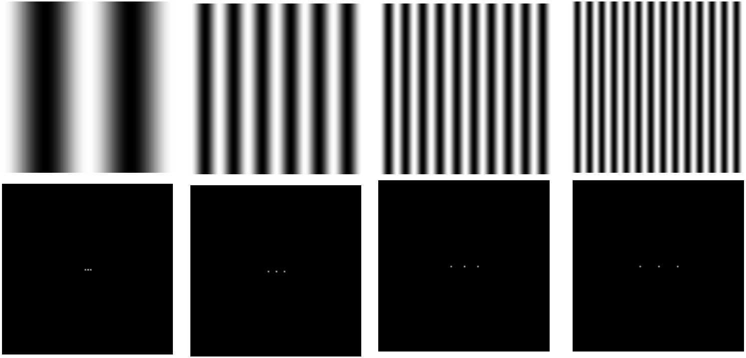


$$u^2 + v^2 = r^2$$
$$\theta_1 \le \tan^{-1}\frac{v}{u} \le \theta_2$$

sector

Sum the magnitudes for $u, v \in \Re$

Next in DDCS



Feature Selection and Extraction

- Signal basics and Fourier Series
- 1D and 2D Fourier Transform
- **Another look at features**
- Convolutions