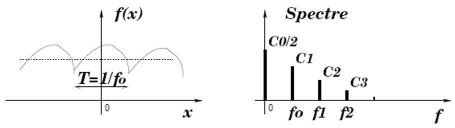
TRANSFORMÉE DE FOURIER
NOTION DE SPECTRE & SÉRIE DE FOURIER (2)

SÉRIE DE FOURIER

Soit $f(x)$, un signal périodique de période $T = 1/f_0$



$$f(x) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos(2\pi n f_0 x) + \sum_{n=1}^{\infty} B_n \sin(2\pi n f_0 x)$$

$$A_n = \frac{2}{T}\int_0^T f(x)\cos(2\pi n f_0 x)\,dx \quad \text{n>0}$$

$$B_n = \frac{2}{T}\int_0^T f(x)\sin(2\pi n f_0 x)\,dx \quad \text{n>0}$$

$$<=> f(x) = \sum_{n=1}^{\infty} C_n \cos(2\pi n f_0 x + \theta_n) + C_0/2$$

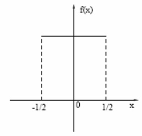
où $C_n = \sqrt{A_n^2 + B_n^2}$ et $\theta_n = \arctan\left(\frac{B_n}{An}\right) - \frac{\pi}{2}$

Spectre de raies

- $C_0/2$ ◄► Niveau moyen du signal
- f_0 ◄► Le fondamental
- nf_0 ◄► Les harmoniques

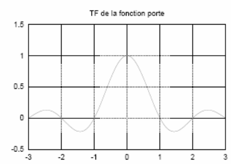
TRANSFORMÉE DE FOURIER
TRANSFORMÉE DE FOURIER -EXEMPLE- (3)

Fonction "Porte"



$$\begin{cases} \Pi(x) = 1 & \text{si} \quad |x| \leq \frac{1}{2} \\ \Pi(x) = 0 & \text{si} \quad |x| > \frac{1}{2} \end{cases}$$

$$\begin{aligned} F(\nu) &= \int_{-\infty}^{+\infty} f(x)\exp(-2\pi j\nu x)\,dx = \int_{-1/2}^{1/2}\exp(-2\pi j\nu x)\,dx \\ &= -\frac{1}{2\pi j\nu}\left[\exp(-2\pi j\nu x)\right]_{-1/2}^{1/2} \\ &= -\frac{1}{2\pi j\nu}\left[\exp(-\pi j\nu) - \exp(\pi j\nu)\right] \\ &= -\frac{1}{2\pi j\nu}\left[\cos(\pi\nu) - j\sin(\pi\nu) - \cos(\pi\nu) - j\sin(\pi\nu)\right] \\ F(\nu) &= \frac{\sin(\pi\nu)}{\pi\nu} = \text{sinc}(\nu) \end{aligned}$$

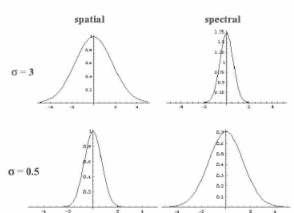


TRANSFORMÉE DE FOURIER
TRANSFORMÉE DE FOURIER -EXEMPLE- (5)

Fonction Gaussienne

Soit $f(x) = \exp(-\pi x^2)$

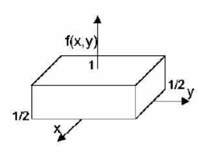
$$\begin{aligned} F(\nu) &= \int_{-\infty}^{+\infty}\exp(-\pi x^2)\exp(-2\pi j\nu x)\,dx \\ &= \exp(-\pi\nu^2)\exp(\pi\nu^2)\int_{-\infty}^{+\infty}\exp(-\pi x^2)\exp(-2\pi j\nu x)\,dx \\ &= \exp(-\pi\nu^2)\int_{-\infty}^{+\infty}\exp(-\pi(x+j\nu)^2)\,dx \\ &= \exp(-\pi\nu^2)\underbrace{\int_{-\infty}^{+\infty}\exp(-\pi w^2)\,dw}_{1} \qquad (w = x + j\nu) \\ F(\nu) &= \exp(-\pi\nu^2) \end{aligned}$$

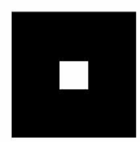


$$\mathcal{F}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})\right] = \exp -2\pi^2\sigma^2\nu^2$$

TRANSFORMÉE DE FOURIER
TRANSFORMÉE DE FOURIER 2D -EXEMPLE- (2)

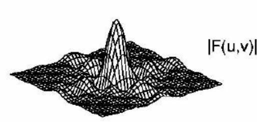
Fonction "Rectangle"

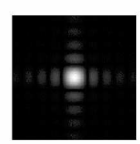




$$\begin{cases} f(x,y) = \Pi(x,y) = 1 & \text{si} \quad |x| \leq \frac{1}{2}, \quad |y| \leq \frac{1}{2} \\ f(x,y) = \Pi(x,y) = 0 & \text{sinon} \end{cases}$$

$$\begin{aligned} F(u,\nu) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp(-2\pi j(ux+\nu y))\,dxdy \\ &= \int_{-1/2}^{1/2}\exp(-2\pi jux)\,dx\int_{-1/2}^{1/2}\exp(-2\pi j\nu y)\,dy \\ &= \frac{\sin(\pi u)}{\pi u}\frac{\sin(\pi\nu)}{\pi\nu} \\ &= \text{sinc}(u,\nu) \end{aligned}$$





2-D Discrete Fourier Transform

If $f(m, n)$ is a function of two discrete spatial variables m and n, then we define the *two-dimensional Fourier transform* of $f(m, n)$ by the relationship

$$F(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) e^{-j\omega_1 m} e^{-j\omega_2 n}$$

The variables ω_1 and ω_2 are frequency variables; their units are radians per sample. $F(\omega_1, \omega_2)$ is often called the *frequency-domain* representation of $f(m, n)$. $F(\omega_1, \omega_2)$ is a complex-valued function that is periodic both in ω_1 and ω_2, with period 2π. Because of the periodicity, usually only the range $-\pi \le \omega_1, \omega_2 \le \pi$ is displayed. Note that $F(0, 0)$ is the sum of all the values of $f(m, n)$. For this reason, $F(0, 0)$ is often called the *constant component* or *DC*
The DFT is usually defined for a discrete function $f(m, n)$ that is nonzero only over the finite region $0 \le m \le M-1$ and $0 \le n \le N-1$. The two-dimensional M-by-N DFT and inverse M-by-N DFT relationships are given by

$$F(p, q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j(2\pi/M)pm} e^{-j(2\pi/N)qn} \qquad \begin{array}{l} p = 0, 1, \ldots, M-1 \\ q = 0, 1, \ldots, N-1 \end{array}$$

ATTENTION: in Matlab, as arrays cannot have null indices, the first term *F(0,0)*, representing the continuous component (DC), will be F(1,1).

Use the command **fft2(A)** to compute it. To speed up the process, if A size is not power of 2, it is possible to automatically perform a zero-padding up to power of 2 size by: **fft2(**A,Nrows, Ncolumns**)**

Caiani 30/03/2012

Even if f(x,y) is real, its transform is complex in general.
The principal method for analyzing a transform visually is to compute its **spectrum** (i.e., the magnitude of F(u,v), which is a real function) and display it as an image.
Letting R(u,v) and I(u,v) representing the real and imaginary components of F(u,v), the Fourier spectrum is defined as:

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

The **phase angle** of the transform is defined as:

$$|\phi(u,v)| = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$$

The **power spectrum** is defined as:

$$|F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

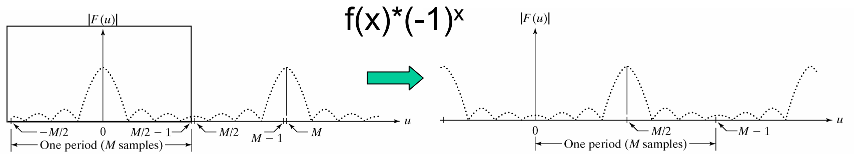
To compute magnitude and phase in Matlab:
mod=**abs**(**fft2**(f)); phase=**angle**(**fft2**(f)) or =**atan2**(**imag**(**fft2**(f),**real**(**fft2**(f))

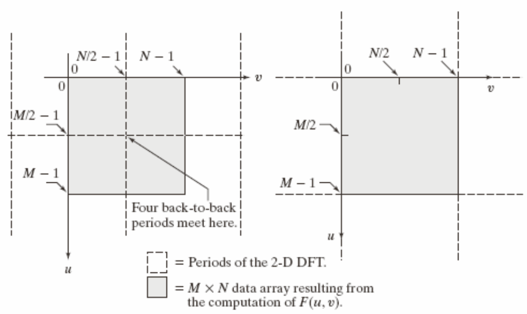
Load an image and compute its magnitude and phase.

Caiani 30/03/2012

If f(x,y) is real, its Fourier transform is conjugate symmetric about the origin:
F(u,v)=-F*(-u,-v)
This implies that the spectrum is symmetric about the origin too.

$f(x)*(-1)^x$

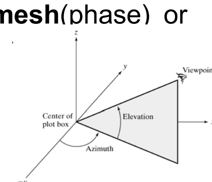




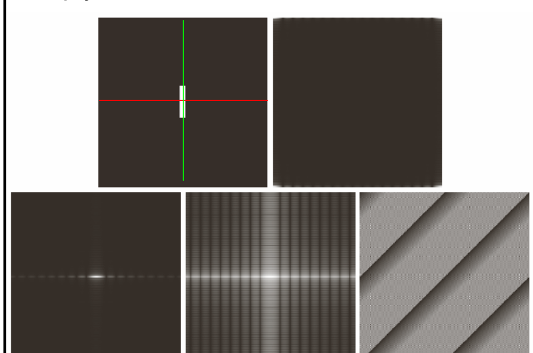
a b
FIGURE 4.2 (a) $M \times N$ Fourier spectrum (shaded), showing four back-to-back quarter periods. (b) Spectrum after multiplying $f(x, y)$ by $(-1)^{x+y}$ prior to computing the Fourier transform. The shaded period is the data that would be obtained by using the DFT.

Caiani 30/03/2012

To visualize the magnitude and phase: **mesh**(mod) and **mesh**(phase) or **imshow,** or **contour**(A). Using view(AZ,EL) it is possible observing point of the 3D mesh.



To visualize the DC component at the center of the spectrum:
modshift=**fftshift**(mod); **mesh**(modshift)
To better visualize values close to zeros, use **log**(1+modshift) instead than simply modshift.



a b
c d e
FIGURE 4.3 (a) Image. (b) Fourier spectrum. (c) Centered spectrum. (d) Spectrum visually enhanced by a log transformation. (e) Phase angle image.

The phase angle is as important in terms of information content as the magnitude, but not as intuitive. It carries infos on the displacement of the sinusoids of different frequency that generates the image with respect to their origin. Moving the rectangle in the image would reflect in the same spectrum but in a modified phase.

Verify on a similar image what written above, testing the various visualization commands.

Caiani 30/03/2012

Frequency content of a 2D rectangle function

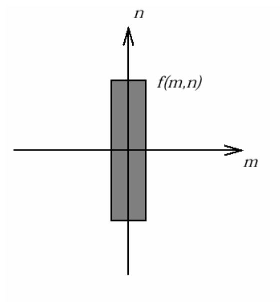


Figure 8-1: A Rectangular Function

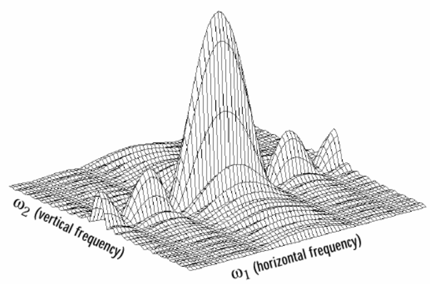
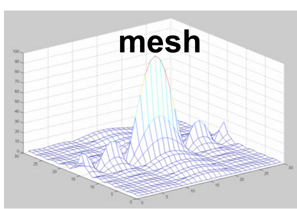


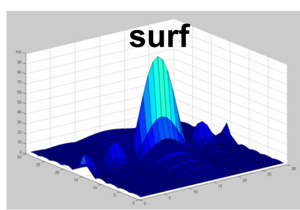
Figure 8-2: Magnitude Image of a Rectangular Function

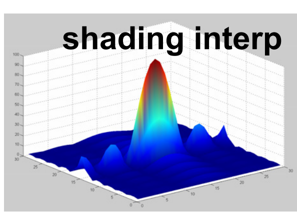
The peak at the center of the plot is $F(0, 0)$, which is the sum of all the values in $f(m, n)$. The plot also shows that $F(\omega_1, \omega_2)$ has more energy at high horizontal frequencies than at high vertical frequencies. This reflects the fact that horizontal cross sections of $f(m, n)$ are narrow pulses, while vertical cross sections are broad pulses. Narrow pulses have more high-frequency content than broad pulses.

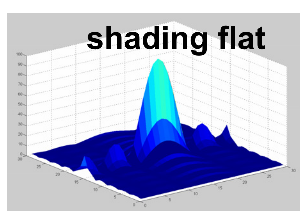
Caiani 30/03/2012

Visualization Commands









axis normal: remove the effects of square and image

Try the visualization commands on the previously computed spectrum.

Moreover, with **axis** it is possible to change the setting of axis visualization:

axis on/off

axis xy: cartesian

axis ij: matrix

axis image: same scale for the 3 axis

axis square: squared box

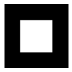
axis fill: filled box

axis([Xmin Xmax Ymin Ymax Zmin Zmax]): set the scale

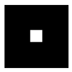
Caiani 30/03/2012

Impluse frequency content

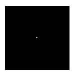
Create the following images, starting from a 64x64 matrix, in which a square of progressively reducing size is represented, up to a single pixel:



```
f=zeros(64,64);
f(17:48,17:48)=1;
```



```
g=zeros(64,64);
g(27:38,27:38)=1;
```



```
i=zeros(64,64);
i(32,32)=1;
```

Visualize magnitude and phase of FFT2 for these images, to understand the relation between the amplitude of the rect function and its frequency content.

Caiani 30/03/2012

GENERATE A TEST FUNCTION: COSINUSOIDAL

Consider a f(x,y) = f(x) invariant along the y direction: it is constituted by a cosinusoidal function as follows:

```
% N = dimensions of A
% TETA = tilt cosinusoid in respect to x axis in rad
% FREQ = spatial frequency in cycles/sample (1/FREQ=samples per cycle)
% FI = phase cosinusoid (e.g. : 0=cos; -pi/2=sin)

function A=immcos(amp,N,TETA,FREQ,FI);
WX=2*pi*cos(TETA)*FREQ; % pulsazione lungo l'asse x
WY=2*pi*sin(TETA)*FREQ; % pulsazione lungo l'asse y
for IX=1:N,
        for IY=1:N,
                ICOL=IX; % l'indice di colonna rappresenta la x
                IRIGA=IY; % l'indice di riga rappresenta la y
                A(IRIGA,ICOL)=amp*cos(WX*IX+WY*IY+FI);
        end
end
```

Using A=immcos(amp,N,TETA,FREQ,FI) you can generate a matrix A of NxN elements, with cosinusoid of magnitude amp, rotated by TETA (in rad), of spatial frequency FREQ and initial phase FI.

Caiani 30/03/2012

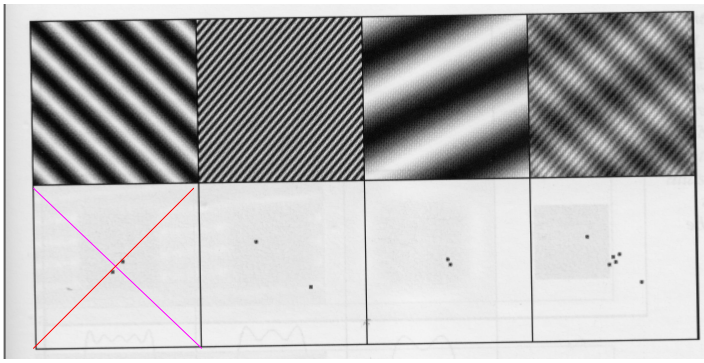


Figure 5. Three sinusoidal patterns, their frequency transforms, and their sum.

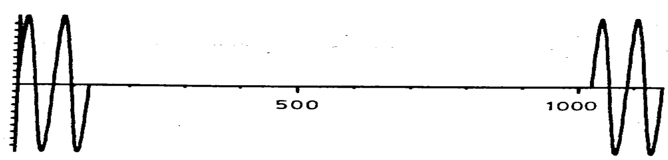
Generate a sinusoidal image of 256x256 elements, not rotated, of unit amplitude, 1/64 frequency,and visualize it.

Change the parameters generating different images, putting them in a struct array with fields names that reflect the sinusoid properties.

Compute fft2 of some previously generated images, visualizing magnitude and phase and understanding what the image content.

ZERO PADDING and FFT

The number N of samples in the signal is artificially increased adding zeros.



in figure: to the original signal of 128 samples, zeros has been added to obtain a total length of 1028 samples

Consequently, the number of N spectral lines is increased…

How this reflect on the spectral resolution?
It does not increase it, as adding zeros does not increas information!!!

ZERO PADDING and FFT

$$x(n) = \cos(0.48\pi n) + \cos(0.52\pi n)$$

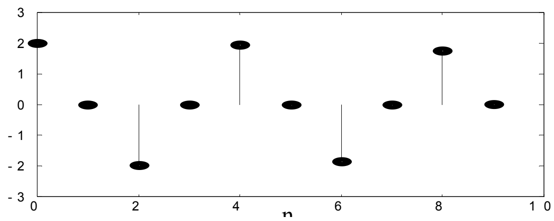
delta $\omega = 0.04\pi$

Compute x(n) spectrum on a finite number of samples:

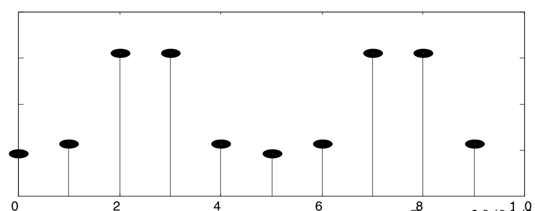
On <u>10 samples</u>,

$0 \le n \le 9$:

$x_{10}(n)$



$|X_{10}(k)|$



$0 \le k \le 9$:

Frequency resolution

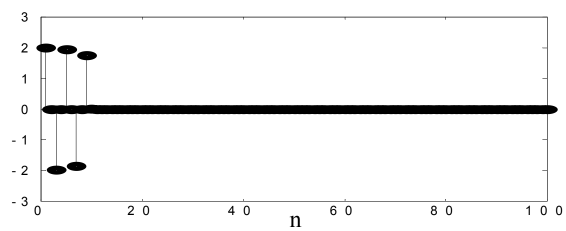
$2\pi/10 = 0.2\,\pi$

k

Caiani 30/03/2012

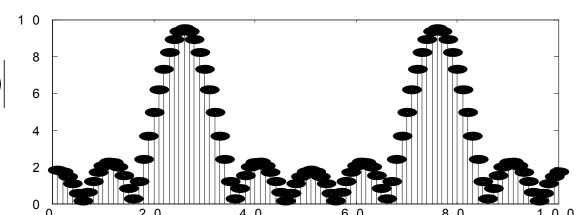
Apply zero-padding up to obtain 100 samples:

$\tilde{x}_{100}(n)$



n

$|\tilde{X}_{100}(k)|$



$0 \le k \le 99$:

Distance between samples

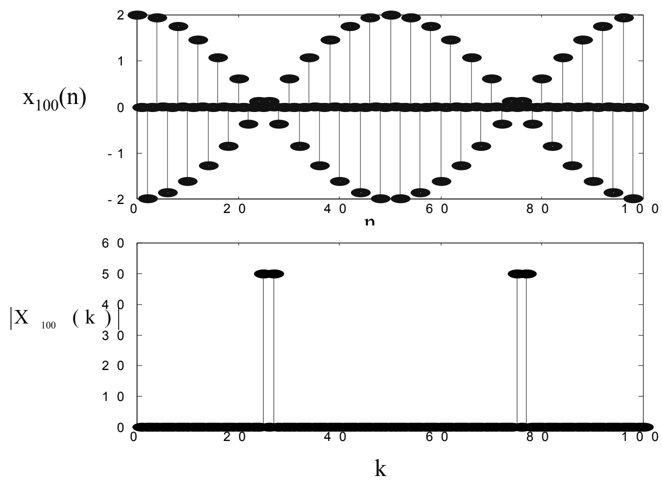
$2\pi/100$

k

- The spectrum is more densely sampled, but no new info on the signal frequency content is added.

Caiani 30/03/2012

Compute now the DFT on <u>100 real samples of the sequence</u>.



$0 \le k \le 99$:

Frequency resolution

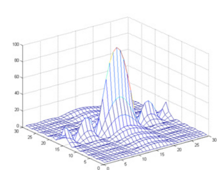
$2\pi/100 = 0.02\pi$

➢The spectrum computed on larger number of sample points has a greater resolution, thus being able to distinguish the two frequencies contained in the sequence.

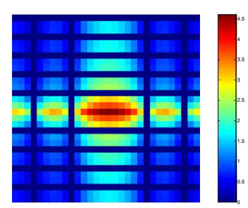
Caiani 30/03/2012

Zero-padding

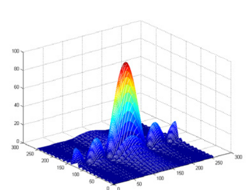
f = zeros(30,30);
f(5:24,13:17) = 1;
imshow(f,'notruesize')



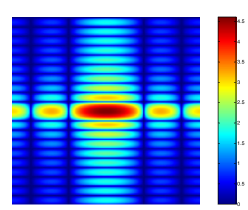
F = fft2(f);
mesh(fftshift(abs(F)))



Flog = log(1+abs(F));
figure,imshow(fftshift(Flog),[],'notruesize')
colormap(jet),colorbar



Fpad = fft2(f,256,256);
mesh(fftshift(abs(Fpad)))



Fpadlog = log(1+abs(Fpad));
figure,imshow(fftshift(Fpadlog),[],'notruesize')
colormap(jet),colorbar

Caiani 30/03/2012

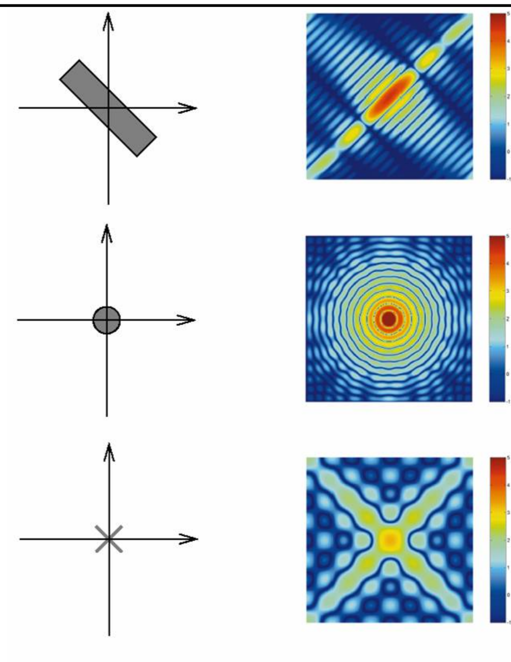


Figure 8-4: Fourier Transforms of Some Simple Shapes

Try to generate a binary image containing a simple geometrical form and compute its fft2, visualizing the spectrum with or without zero-padding.

<u>High frequency, low frequency</u>

Generate the following images:

A=immcos(1,256,0,1/128,-pi/2)
B=immcos(1,256,0,1/16,-pi/2)
C=immcos(1,256,0,1/64,-pi/2)

Order them based on increasing spatial frequency

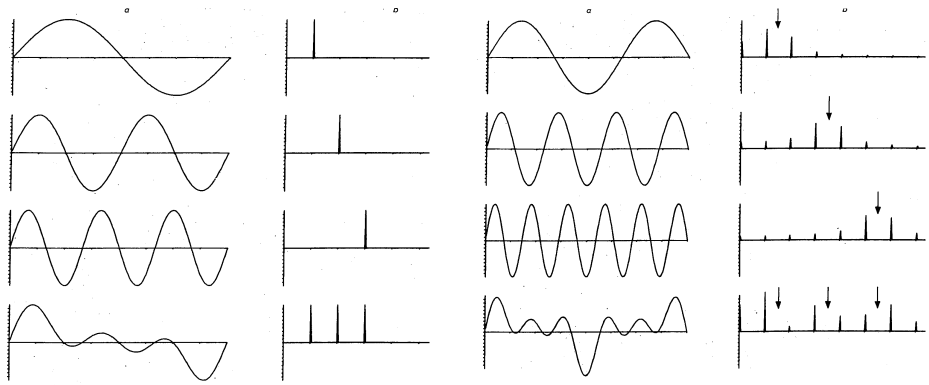
OSBERVATION: The frequency content of an image reflects information on the size of the objects in it. Low frequency implies big objects, high frequency implies small objects.

Generate the following images:

A16=immcos(1,256,0,1/16,-pi/2)
B16=immcos(1,255,0,1/16,-pi/2)

Are the spectrums different? How do you explain this result?

Spectral leakage



Signal length=number of points in the sinusoid period.
The frequency of the 3 sinusoids corresponds to the frequency of the spectral lines.

The length of the signal is different from the number of points inside the sinusoidal period. The frequency of the 3 sinusoids does not correspond to the frequency of the spectral lines: we cannot obtain from the spectra the correct information.

Caiani 30/03/2012

INVERSE TRANSFORM

$$f(m,n) = \frac{1}{MN}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} F(p,q)e^{j(2\pi/M)pm}e^{j(2\pi/N)qn} \qquad \begin{array}{l} m = 0, 1, \ldots, M-1 \\ n = 0, 1, \ldots, N-1 \end{array}$$

F=**fft2**(A);
B=**ifft2**(F)
If the antitransform is applied to F, the exact starting image is obtained (in reality, A=**real**(B)).

Fpad=**fft2**(A,256,256);
B=**ifft2**(Fpad)
If ifft is applied to the Fpad (obtained with padding), an image B including the starting one, but surrounded by zeros is obtained.

Fpad=**fft2**(A,256,256);
Fpads=**fftshift**(Fpad);
B=**ifft2**(Fpads)

ATTENTION: if ifft is applied to Fpads (with fftshift) a different image is obtained, in respect of the expected one.
If a shift has been introduced, an inverse operation has to be computed with **ifftshift** , and then apply the ifft:
Fpad=**ifftshift**(Fpads);
B=**ifft2**(Fpad)

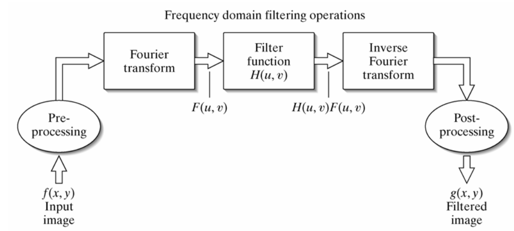
Try the previous commands.

Caiani 30/03/2012

SPATIAL FILTERING IN FREQUENCY

Given the Fourier transform H(u,v)of the linear filter h(x,y) and the Fourier transform F(u,v) of the image f(x,y), the Fourier transform of the filtered image is G(u,v)=H(u,v)F(u,v)

N.B.: the product implies that H(u,v) and F(u,v) have the same dimensions.



The filtered image is obtained by:
g(x,y)=**ifft2**(G(u,v))

Working in the frequency domain allows to reduce the computational burder associated with filtering (convolution)

Caiani 30/03/2012