

## Exercise Chapter 3 – Fast Fourier Transform (FFT)

In this exercise, you will visualize the (spatial) frequency response for some examples of images.

Before starting, load and unzip the file “*fft.zip*” which contains the scripts you will need throughout this exercise.

### Discrete Fourier Transform

1 – Add your own working folder paths to the path list in the path browser then open and analyze the script *fft2d\_sinus.m*. Use the Matlab help (command *helpwin*) to understand how the Matlab functions that we are using work. What is the kind of the 2D input signal?

2 – Modify the spatial period value (called *period* in the script): enter the values 4, 8, and 16 (orientation 0). For all these period values, note the magnitude and the normalized frequencies of the lines down. Change also the amplitude and the DC offset (*dc*) of the 2D signal. Interpret the results. Next, test the period value 17 and interpret again.

3 – Use the script *fft2d\_resolution.m* to compute a higher frequency resolution FFT. Change now the 2D signal orientation: test  $\pi/2$  then  $\pi/4$  (period = 16 and period =  $16*\sqrt{2}$ ).

4 – Run the scripts *fft2d\_square.m* and *fft2d\_checkerboard.m*. Change the parameters and interpret the results. By using the script *fft2d\_sinus.m*, write a script that displays a natural image and its frequency response. Run this script on several images.

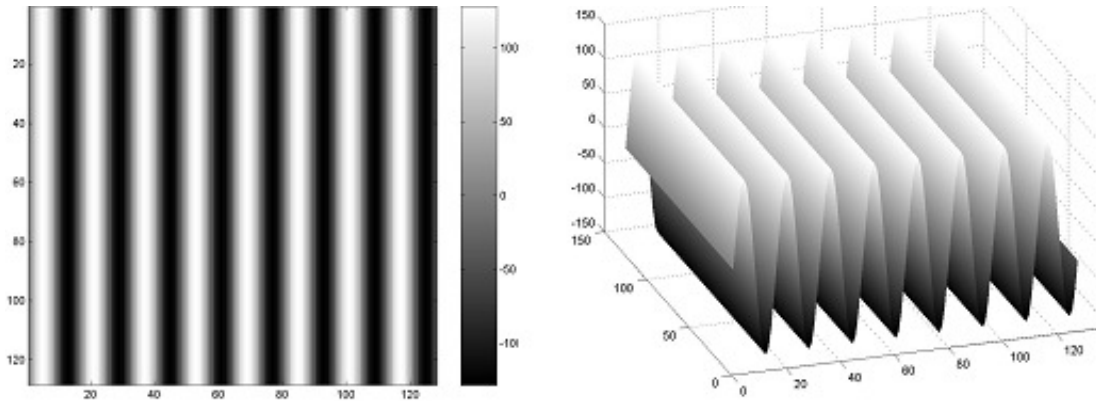
## Solution to the exercise on FFT

1 - The script `fft2d_sinus.m` allows you to visualize the discrete Fourier transform of a 2D sinusoidal signal.

First, we define the characteristics of the 2D signal: image size, spatial period, orientation, amplitude, and DC offset. Then the function `generation_sinus` generates a 2D signal from these characteristics (the function `generation_sinus` has been written for you).

```
Size1 = 128 % image size
period = 16 % spatial period
orientation = 0
amplitude = 128
dc = 0 % DC offset
im1 = generation_sinus(size1,period,orientation,amplitude,dc);
```

The image of the sinusoid is then visualized in two manners: in a view from the top (2D vision with the function `imagesc`) and in perspective view (3D vision with the function `surf`):



We note that the horizontal direction of the sinusoid involves the generation of vertical bands on the image (left). Once the sinusoid image is computed and stored in `im1`, we compute its fast Fourier transform (FFT) with the Matlab function `fft2`.

```
spectrum1 = fft2(im1)/(size1*size1);
```

Then we use the function `fftshift` to shift the zero-frequency component (0, 0) of the fast Fourier transform to the center of the spectrum.

```
spectrum1 = fftshift(spectrum1);
```

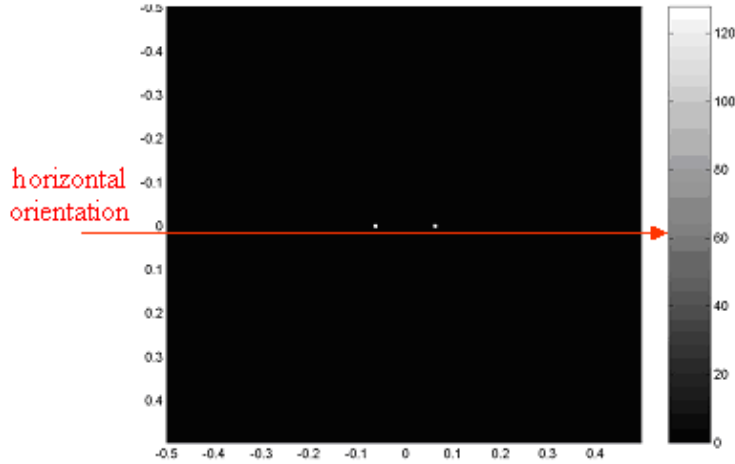
The normalized vectors represent the horizontal and vertical spatial frequencies. Here these vectors have the same length and the same components, giving:

```
vt = (- size1/2:1:( size1/2-1))/ size1;
```

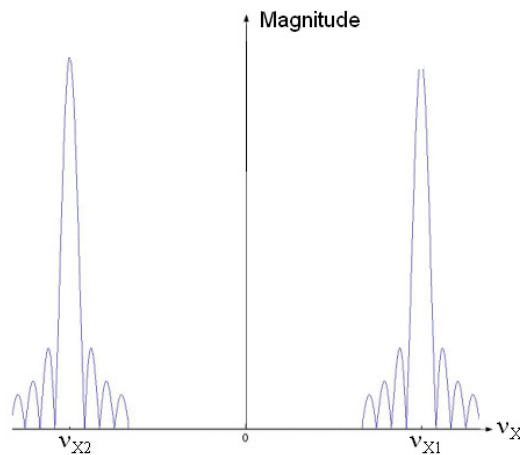
The absolute value (modulus) of the spectrum can be displayed with the command:

```
imagesc(vt,vt,(abs(spectrum1)));
```

Here is the image of the spectrum modulus:



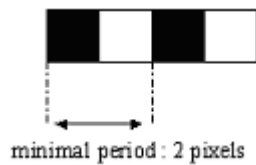
We see two white points. We will show later that these points are not two delta functions but they are the two main lobes of a sinus cardinal function because the 2D sinusoidal function is **bandlimited**:



Note that the orientation is horizontal along these two points (it is orthogonal to the vertical orientations of the bands in the original sinusoid image).

Note:

The spatial period is the minimal number of pixels between two identical patterns in a "periodic" image. The minimal spatial period in an image is thus two pixels:



The maximal normalized spatial frequency  $v_{max}$  is thus equal to:

$$v_{max} = \frac{1}{\text{minimal period}} = \frac{1}{2}$$

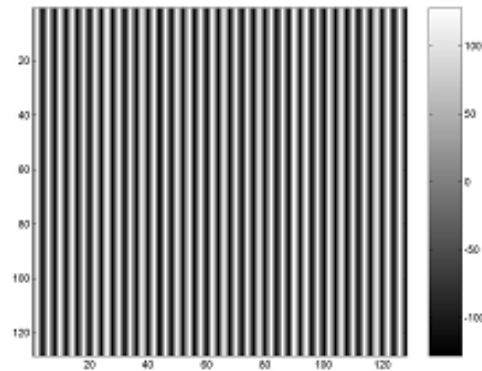
We can plot the image frequency response along the normalized spatial frequencies which belong to the range  $[-0.5, 0.5]$ .

Let  $T$  be the period. The corresponding normalized frequency  $v_{\text{norm}}$  is defined by:  $v_{\text{norm}} = 1/T$ .

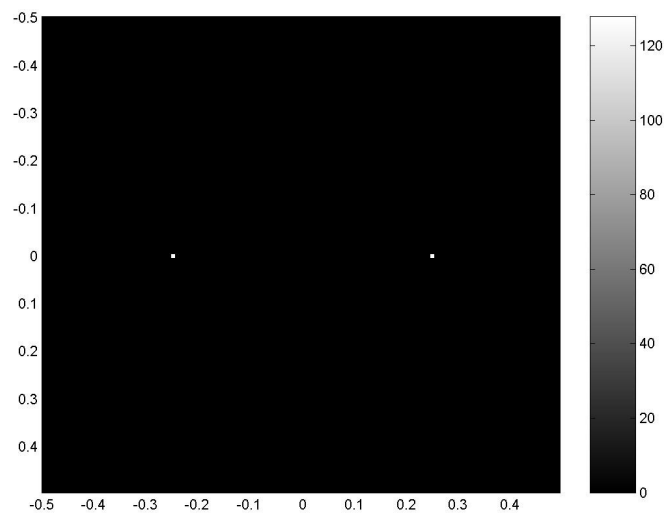
2 -

- Changing the period:

By decreasing the spatial period of the sinusoid (value = 4), we visualize the image below:



On the image, we observe one line per four pixels.  
The magnitude image is displayed below:



The two white points are located at the spatial frequency coordinates:

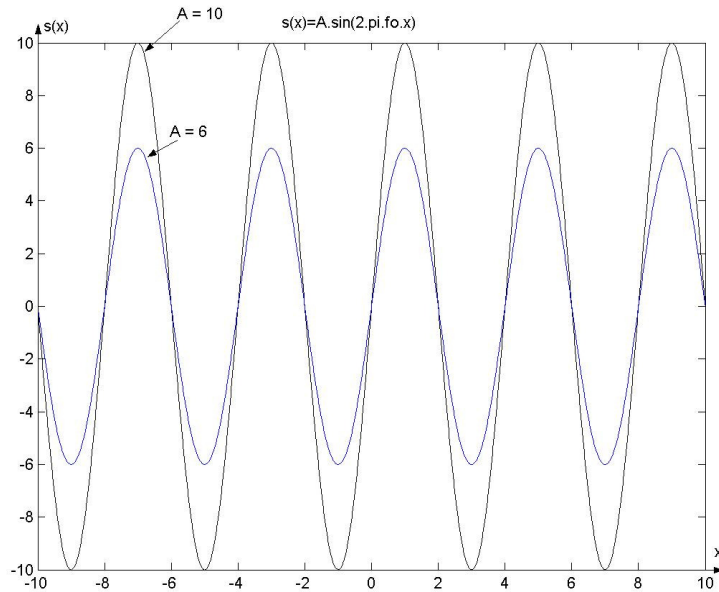
$$\{v_{1x} = -0.25, v_{1y} = 0\} \text{ et } \{v_{2x} = 0.25, v_{2y} = 0\}$$

When the orientation of the 2D sinusoid is zero, we can represent one line of this 2D signal by a 1D sinusoid signal which propagates along the horizontal axis ( $Ox$ ).

The absolute value of the horizontal spatial frequency  $v_{1x}$  is equal to the one of  $v_{2x}$ . In fact:  $v_{1x} = -v_{2x}$  thus  $|v_{1x}| = |-v_{2x}|$ . This absolute value is the inverse of the spatial period:  $v_{1x} = -1/T$  and  $v_{2x} = 1/T$ .

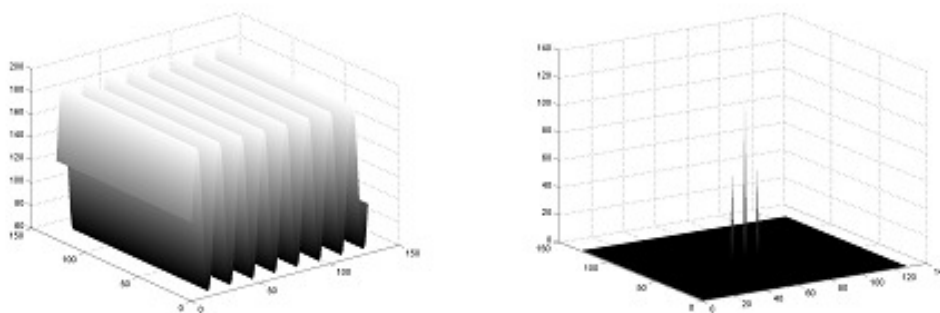
- Changing the amplitude

The sinusoid values belong to the range  $[-A, A]$ , where  $A$  stands for the amplitude. The change of the amplitude  $A$  influences the magnitude values of the two white points that we had previously noticed on the spectrum image.



- Changing the DC offset:

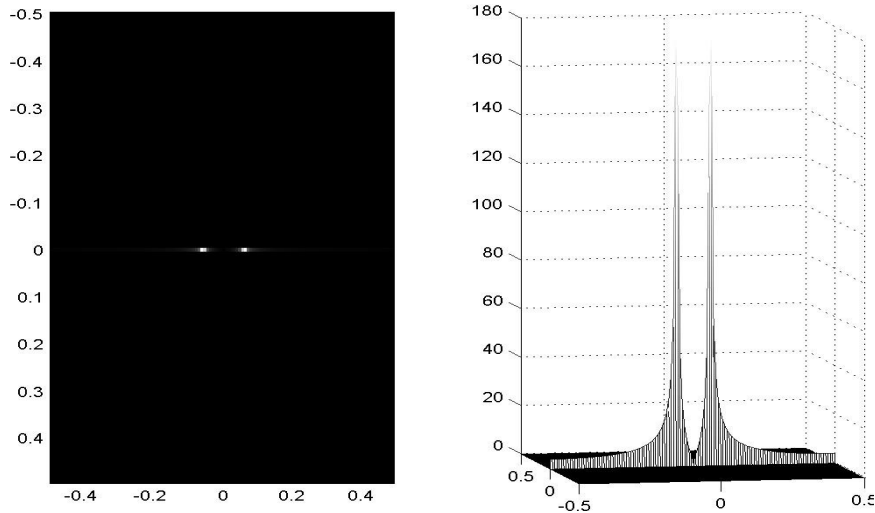
By increasing the DC offset  $V_M$ , we visualize the images below:



The spectrum modulus has now a new frequency component at the center (image on the right) adding to the two preceding frequency components. This component corresponds to the spectrum DC gain located at the spatial frequency coordinates  $\{v_x = 0, v_y = 0\}$ . This DC gain is equal to the DC offset of the sinusoid.

- Period T = 17:

We compute the fast Fourier transform with the spatial period 17:



On the Fourier plane (left), the two white points are "spread out". On the right image, note that the frequency response is not composed of only two impulses any more. The frequency response looks like the envelope of a sinus cardinal function.

Let us consider a 1D sinusoidal signal to explain this result. This signal is called  $f(x)$  and defined by:

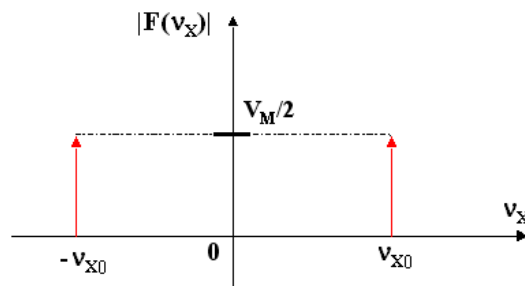
$$f(x) = V_M \cdot \sin(2 \cdot \pi \cdot f_0 \cdot x)$$

The spectrum  $F(v_x)$  of this sinusoidal signal is thus defined by:

$$F(v_x) = V_M \cdot \left( \frac{1}{2j} \cdot \delta(v_x - v_{x0}) + \frac{j}{2} \cdot \delta(v_x + v_{x0}) \right)$$

where "j" stands for the complex number such as  $j^2 = -1$ , and  $v_{x0} = 1/T$ .

The spectrum modulus  $|F(v_x)|$  is plotted on the figure below:

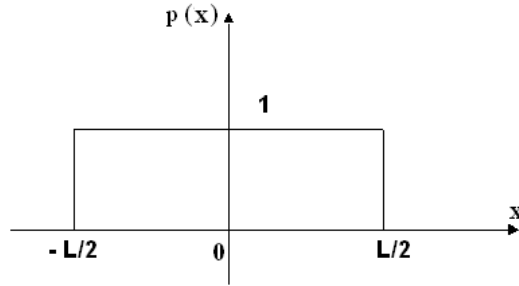


This spectrum is computed for a 1D sinusoidal signal of infinite support ( $x$  ranging from  $-\infty$  to  $+\infty$ ).

In practice we cannot represent an image on an infinite support, therefore the 1D sinusoidal signal is multiplied by a rectangular pulse  $p(x)$ . This rectangular pulse is defined by:

$$p(x) = \begin{cases} 1 & \text{if } x \in [-L/2; L/2] \\ 0 & \text{else} \end{cases}$$

Here is the graph of the function  $p(x)$ :



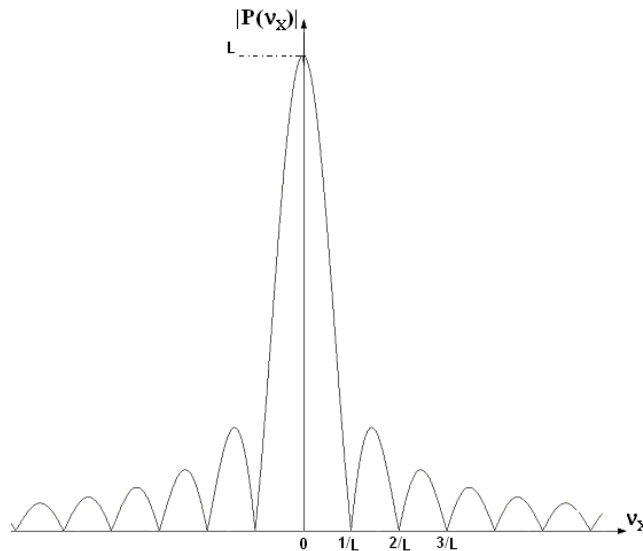
The spectrum  $P(v_x)$  of this rectangular pulse function is defined by:

$$P(v_x) = L \cdot \text{sinc}(L \cdot v_x)$$

Where "sinc" stands for sinus cardinal:  $\text{sinc}(x) = \sin(x)/x$ .

In the script `fft2d_sinus.m`, the size of the input image is  $128 \times 128$  pixels. The sinusoidal signal propagates along the axis (Ox), therefore  $L = 128$ .

The spectrum modulus  $|P(v_x)|$  is displayed on the figure below:



Note that the length of the lobes is  $1/L$  (except for the main lobe whose length is  $2 \times 1/L$ ).

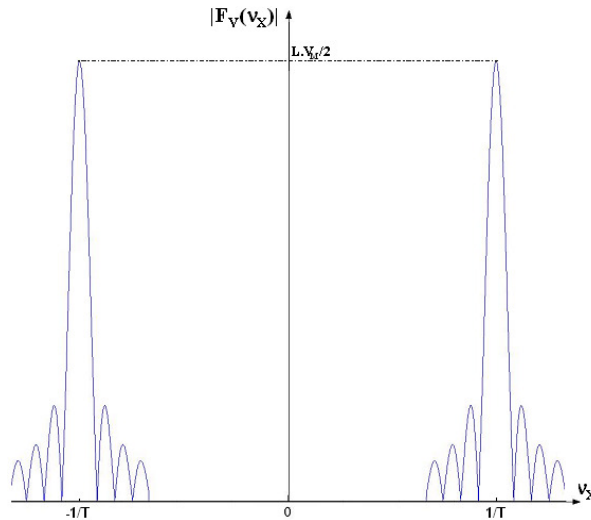
The visualized sinusoid  $f_v(x)$  is thus the function defined by:

$$f_v(x) = f(x) \cdot p(x)$$

Its spectrum  $F_v(v_x)$  is given by the relation:

$$F_v(v_x) = F(v_x) \otimes P(v_x) = \frac{1}{2} \cdot v_m \cdot \left( \frac{1}{j} \cdot P(v_x - v_{x0}) + j \cdot P(v_x + v_{x0}) \right)$$

Let us consider the T-periodic sinusoidal signal  $f_v(x)$  has an amplitude  $V_M$ , and is band-limited on a support whose length is L. Here, its frequency response is plotted:

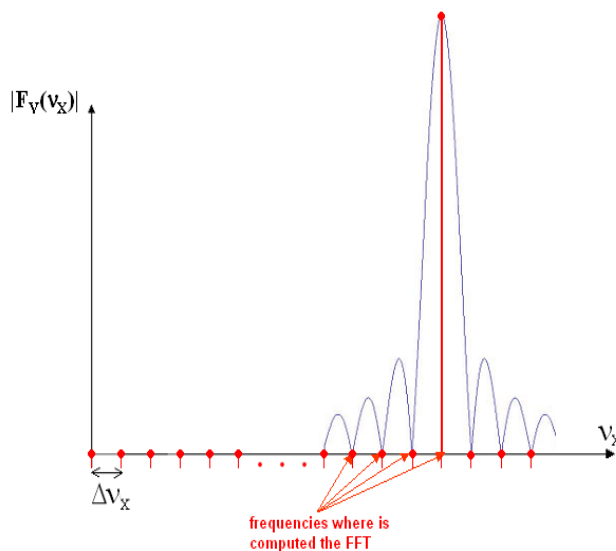


Moreover, to represent the spectrum of this sinusoid, a fast Fourier transform is computed. The spectrum is thus computed for a finite number of its points. By default, with the Matlab function `fft2`, this number of points is equal to the number of pixels in each direction (horizontal and vertical).

Here the number of pixels is the same one according to two directions. After FFT computation the spectrum modulus  $|F_{Vs}|$  is thus represented by 128 points of the spectrum modulus  $|F_V|$  according to the horizontal spatial frequencies. The horizontal frequency resolution  $\Delta v_x$  is then defined by:  $\Delta v_x = 1/128$ .

Note that  $1/L = \Delta v_x$ : on the spectrum of the sinusoidal signal; the length of any lobe (except the main lobe) is equal to the horizontal frequency resolution.

If the period T of the 1D sinusoidal signal is inversely proportional to the horizontal frequency resolution (i.e.  $1/T = k \cdot \Delta v_x$ , where "k" is a non-null natural integer), the spectrum modulus  $|F_{Vs}|$  represents only the two main lobes of the spectrum modulus  $|F_V|$  (the other lobes are set to 0):



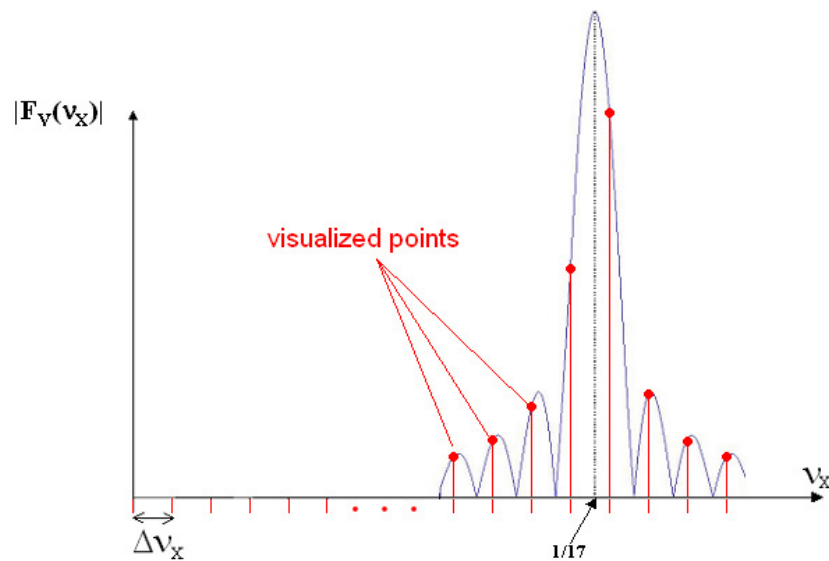


Then for the periods:

- $T = 4, \frac{1}{4} = 32 \times \frac{1}{128},$
- $T = 8, \frac{1}{8} = 16 \times \frac{1}{128},$
- $T = 16, \frac{1}{16} = 8 \times \frac{1}{128},$

The condition  $1/T = k \cdot \Delta v_x$  is thus true for these three periods. We visualize only the two maximal values of the main lobes. The other components are zeroing therefore the spectrum modulus looks like two delta functions (cf. 1).

When the period is set to 17, the condition  $1/T = k \cdot \Delta v_x$  is not true. We visualize thus some samples of the other lobes:



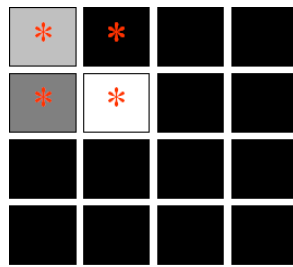
If you want to visualize the other lobes whatever the period is, you must increase the spatial frequency resolution to satisfy the condition:  $\Delta v_x < 1/L$ .

3 - The script ***fft2d-resolution.m*** allows you to increase the **frequency resolution**. We choose along one direction (the horizontal one then the vertical one, or conversely) the number of pixels  $N_2$  to compute the FFT. This number must be superior to the number  $N_1$  of points in the original image along the same direction (horizontal or vertical). The value of the padded points is zero.

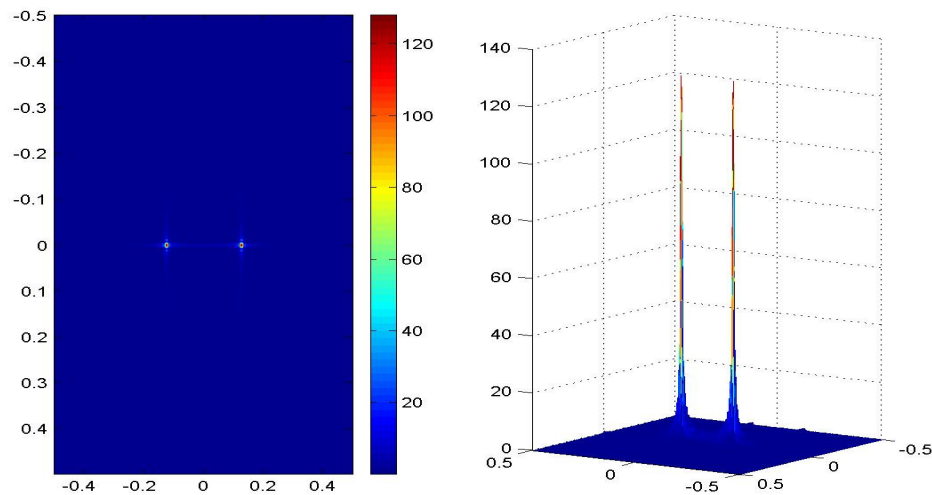
Let us consider a simple (2x2) grayscale image:



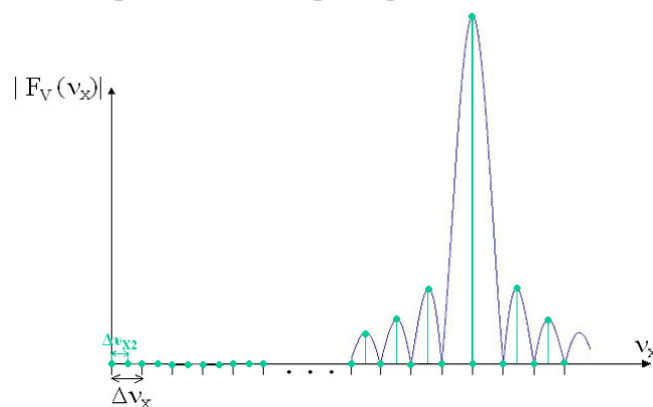
We want to increase the frequency resolution to compute the fast Fourier transform with 4 points along each direction instead of the two original points. The fast Fourier transform is thus computed on the following image ( $N_2=4$ ):



The four original pixels are marked by a red star. The other black pixels (zero padding) have been added to create a (4x4) image. When you want to compute the FFT in Matlab with a higher frequency resolution, you can pass the optional argument  $N_2$  to the Matlab function `fft2`: `fft2(im1,N2,N2)`. Here are the results obtained with a 2D 8-periodic sinusoid:



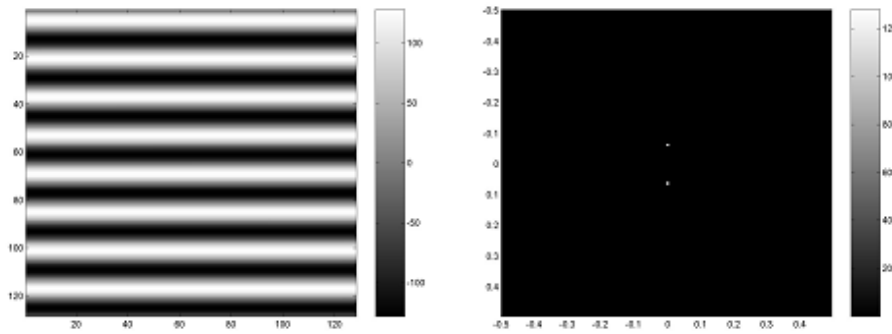
By increasing the frequency resolution, all the spectrum lobes are sampled even if the period is like  $1/T = k \cdot \Delta v_x$ . Let us consider for example a  $\times 2$  frequency resolution:  $\Delta v_{x2} = 2 \cdot \Delta v_x$ :



All the lobes are now represented by a non-zero value (maximal value here).

Changing the orientation:

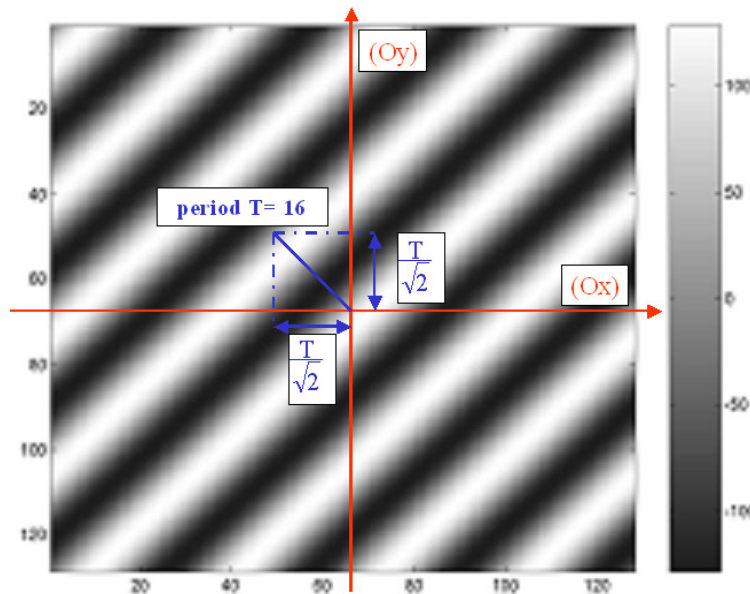
Here are the results obtained after having set the orientation to  $\pi/2$  (period = 16 pixels):



As expected:

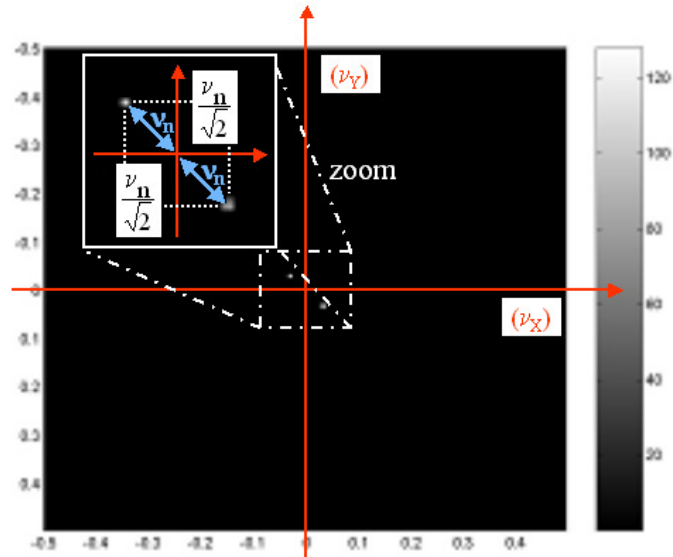
- The propagation is perpendicular to the original one (the bands are now horizontal in the image on the left);
- On the right, the two points of the spectrum define a direction which is perpendicular to the direction in the original spectrum (orientation = 0) and to the bands of the associated left image.

In the same way, by setting the orientation to  $\pi/4$  (period  $T = 16$  pixels), the propagation orientation of the sinusoid is rotated by  $-\pi/4$ . We visualize horizontal and vertical periodic patterns. These ones also create a diagonal periodic pattern whose period is equal to 16 pixels. The horizontal and vertical periods are thus equal to  $\frac{T}{\sqrt{2}}$  pixels:

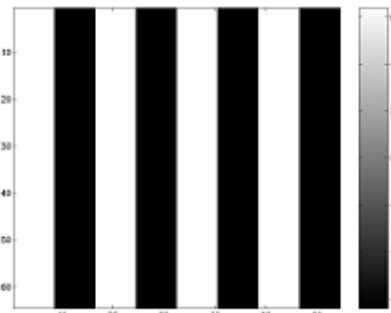


On the image below, the spectrum modulus is represented by two white points which are located at the coordinates:  $\left(-\frac{v_n}{\sqrt{2}}; \frac{v_n}{\sqrt{2}}\right)$  and  $\left(\frac{v_n}{\sqrt{2}}; -\frac{v_n}{\sqrt{2}}\right)$ , where  $v_n = \frac{1}{T}$ .

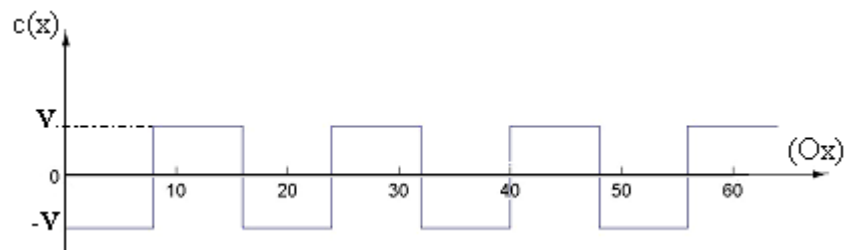
These two points define a direction which is perpendicular to the bands in the associated original image. The 2D image of the frequency response gives several pieces of information about the spatial orientation of the original signal.



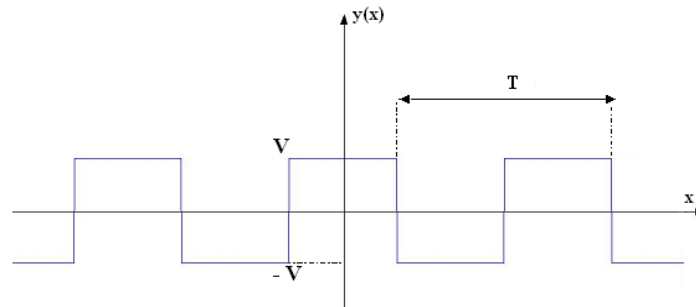
4 - By launching the script `ff2d_square.m`, you visualize the following image:



The 2D signal is a square signal which propagates along the horizontal axis ( $Ox$ ). We can thus represent one line of this 2D signal by a 1D square signal which propagates along the horizontal axis ( $Ox$ ):

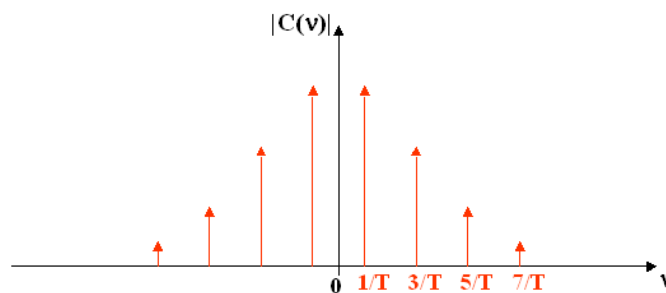


Let us consider the following square wave:



It is established that its 2D Fourier transform  $Y(\nu)$  is defined by the relation:  $Y(\nu) = \sum_n V \cdot \text{sinc}(n/2) \cdot \delta(\nu - n/T)$ .

Note that  $c(x) = y(x + T_1/4) = y(x) \otimes \delta(x + T_1/4)$ , where  $T_1$  stands for the spatial period of  $c(x)$ , then:  $C(\nu) = Y(\nu) \cdot \exp(2j\pi\nu \cdot T_1/4)$ . The spectrum moduli of  $C(\nu)$  and  $Y(\nu)$  are equal. Here is their representation:

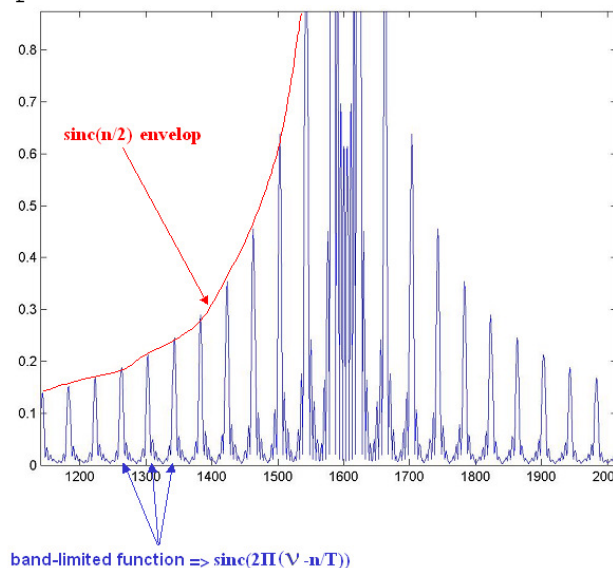


Here the support is assumed to be infinite. In practice, the signal is band-limited, therefore we must again use the rectangular pulse  $p(t)$ :

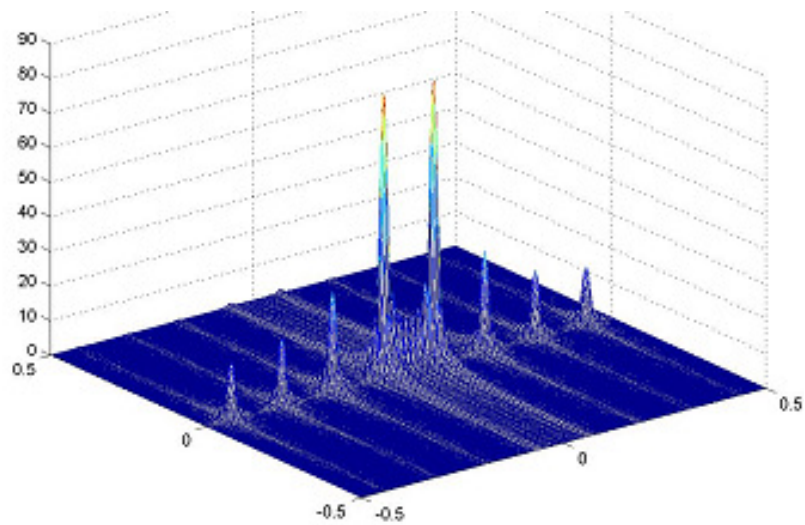
- $|C(\nu)| = \left| \sum_n V \cdot \text{sinc}(n/2) \cdot \delta(\nu - n/T) \otimes L \cdot \text{sinc}(2\pi\nu) \right|$ ,
- Then:  $|C(\nu)| = V \cdot L \cdot \sum_n |\text{sinc}(n/2) \cdot \text{sinc}(2\pi(\nu - n/T))|$

The spectrum modulus  $|C(\nu)|$  is thus equal to the multiplication between two sinus cardinal functions: the first one corresponds to the FFT of the window  $p(t)$  and the second one corresponds to the FFT of the square wave. The spectrum modulus is thus obtained by switching the delta functions for sinus cardinal functions in the figure above.

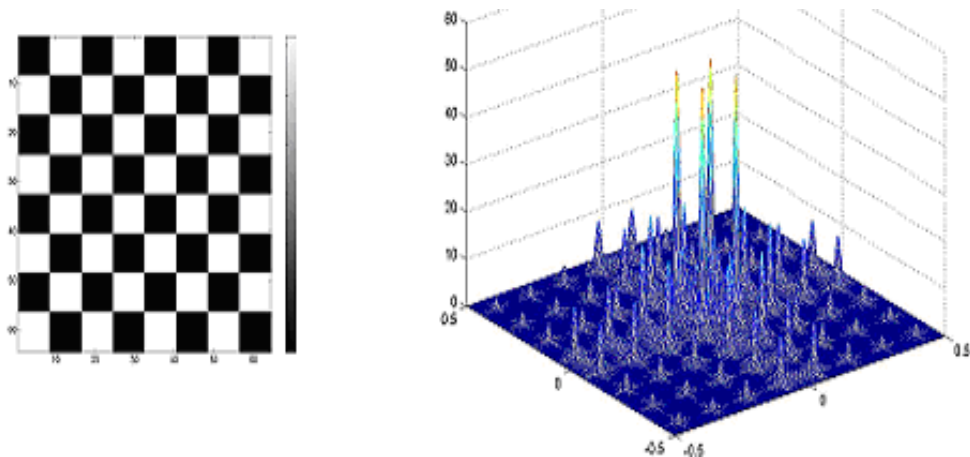
Note that the DC offset is zero, therefore the DC gain is also zero. Here is the form of the spectrum modulus:



Here is the frequency response of the 2D band-limited square signal. As expected the representation of the spectrum modulus is given by the product of two sinus cardinal functions:



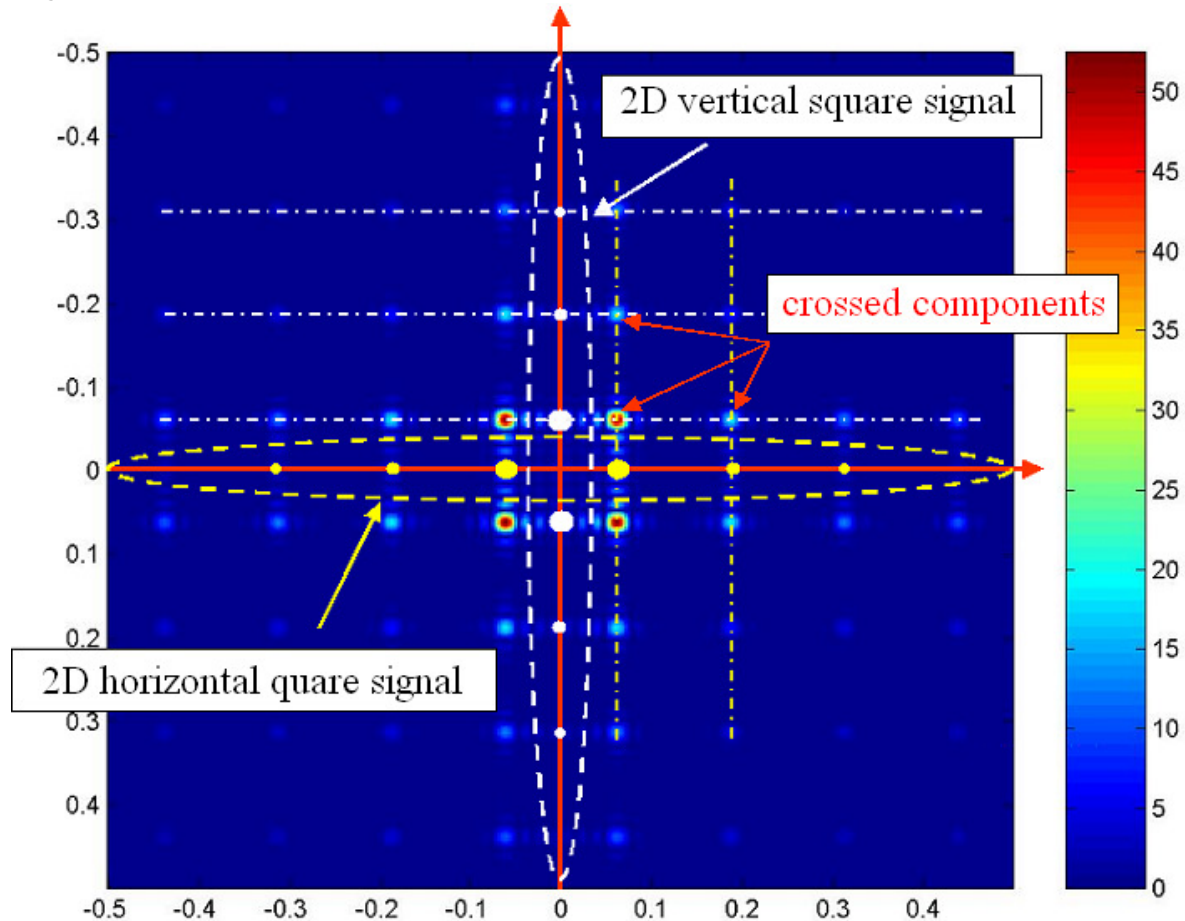
By launching the script `ff2d_checkerboard.m`, we obtain the following results (period = 16):



We perform the XOR-operator between two orthogonal square waves to create the square checkerboard (you can also use the Matlab function `checkerboard`) When you visualize the frequency response of this image, you can notice crossed magnitudes which are similar to the one obtained with a single square wave.

In addition, note that the checkerboard patterns (on the left) are periodic along the diagonal directions of the image.

The frequency response of the checkerboard image is displayed on the image below (period = 16). Here we plot the spectrum modulus of the two basic square signals: the horizontal one and the vertical one (warning: they do not appear when you display the frequency response). The spectrum components of the checkerboard image are equal to the crossed components of the two frequency responses which are associated with the two basic square signals.



5 - Here is an example of a script for displaying the frequency response of the image *CLOWN\_LUMI*:

```

im1=imread('CLOWN_LUMI.BMP') ;
% colormap for displaying in gray levels
figure(1);
imagesc(im1);
map = 0:1/255:1;
map = [map',map',map'];
colormap(map);
% FFT
[size1, size2]=size(im1(:,:,1)) ;
nb_point1 = 2*size1;
nb_point2 = 2*size2;
spectrum1 = fft2(im1,nb_point1,nb_point2) / (size1*size2);
spectrum1 = fftshift(spectrum1);
% To reduce the DC component
spectrum1 = 2*spectrum1;
spectrum1(size1/2+1,size2/2+1)=spectrum1(size1/2+1,size2/2+1)/2;
% normalized frequency vectors
vt1=(-size1/2:size1/nb_point1:(size1/2-size1/nb_point1))/size1;
vt2=(-size2/2:size2/nb_point2:(size2/2-size2/nb_point2))/size2;

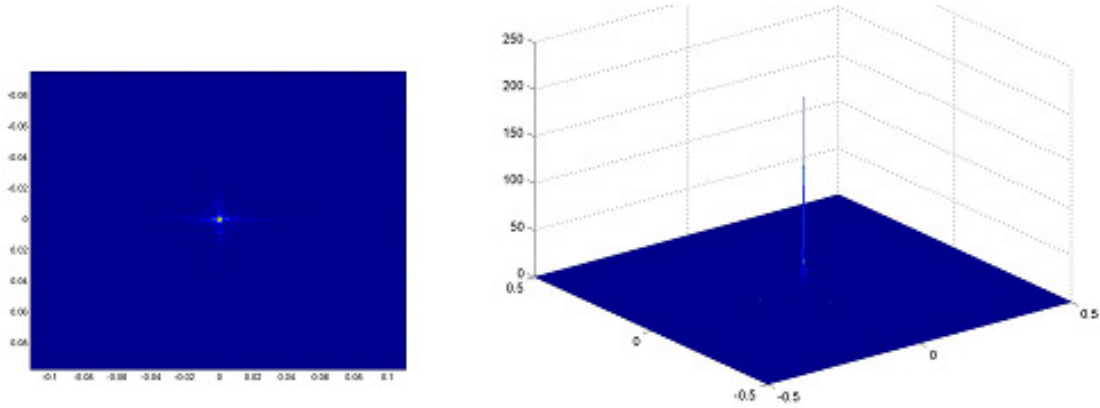
```

```

% To display the spectrum modulus
figure(2);
imagesc(vt1,vt2,(abs(spectrum1)));
% 3-D representation
figure(3)
[X,Y]=meshgrid(vt1,vt2);
mesh(X,Y,abs(spectrum1));

```

Here is the result:



In natural images the DC component is very often higher than the other frequency components (middle and high frequency components). We prefer generally to display the natural logarithm of the spectrum modulus instead of the spectrum modulus itself. Here are the commands to type to display the natural logarithm of the "Clown\_lumi.bmp" image spectrum modulus:

```

% We display the spectrum modulus
figure(2);
imagesc(vt1,vt2,(log10(abs(spectrum1))));
colormap(map) ;

```

Here is the result:

