University of Brescia
Faculty of Engineering

Communication Technologies and Multimedia

Multimedia Information Coding and Description

Laboratory Experience, No.2

Students:

Arrigoni Simone
Bozzetti Eridia
Leali Alberto
Melkamsew Tenaw
Rocco Davide
Simoni Nicola
Wasyhun Asefa

2013/2014

Abstract

In the first part of this experience we implement, in efficient way, the one-dimensional Fourier Transform using Matlab. After that, we develop a Matlab function that computes the two-dimensional Fourier Transform of an image. In the second part of the laboratory, we use this function with a synthetic image generate by a mathematical formula. We repeat the experiment with some signals comparing the obtained results. In the end, we compute the Fourier Transforms of real images (luminance component only) and we compare the spectrum of natural image with the ones of periodic pattern texture images.

1 2D Fourier Transform

In this first part we develop a Matlab function **MCS_FT** that computes the Fourier Transform of a one-dimensional signal (Equation 1); given as inputs the vector of signal values, the vector of their associated time instants, and the vector of frequency values at which the transform is to be evaluated.

$$U(f) = \int_{-\infty}^{\infty} u(t) \cdot e^{-2\pi i t f}\, dt \tag{1}$$

where $u(t) = [..., u(-1), u(0), u(1), ...]$ are sampling points of $u(t)$ associated to t, vector of time instants; while $U(f) = [..., U(-1), U(0), U(1), ...]$ are sampling points of $U(f)$ associated to f, vector of frequencies.

To improve the performance of our code we write this function using the vectorial operations available in Matlab. The function is implemented as follows:

MCS_FT.m

```
% --- 1D Fourier Tranform --- %

function [ U ] = MCS_FT( u, t, f, delta_t )

  U = u * exp(-2*pi*i*f'*t)' * delta_t;

end
```

In order to verify that our Fourier Transform function works well, we use the *rect* signal as input into our code. The result obtained is, as expected, a *sinc* function (see Figure 1).

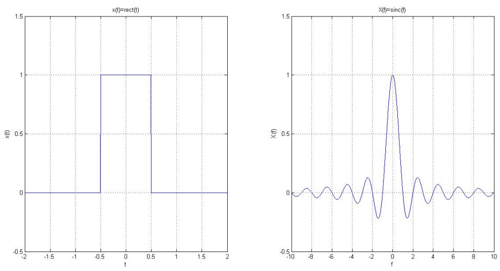

Figure 1: The $rect$ signal and its Fourier transform.

We also write a Matlab function **MCS_FT2** that computes the two-dimensional Fourier Transform of an image (Equation 2). Given as input a matrix of pixel values (gray scale image), the vectors of their associated to spatial-dimensions (in this case: rows and columns), and the vectors of spacial-frequencies values at which the transform is to be evaluated.

$$U(f_x, f_y) = \int \int_{-\infty}^{\infty} u(x,y) \cdot e^{-2\pi i (x \cdot f_x + y \cdot f_y)} \, dx dy \tag{2}$$

It is possible to extend to the two-dimensional case the vectorial operation used in the previous one-dimensional case. The function is implemented as follows:

MCS_FT2.m

```
% --- 2D Fourier Tranform --- %

function [ U ] = MCS_FT2( u, x, y, fx, fy )

  wx = exp(-2*pi*i*(x'*fx))*(dx(2)-dx(1));
  wy = exp(-2*pi*i*(y'*fy))*(dy(2)-dy(1));

  U = wx.' * u * wy;

end
```

In the next section we use this function to compute the Fourier Transform of two-dimensional signals.

2 Deformations in Space

After having implemented the two-dimensional Fourier Transform in the first part, in this sections we use this function to compute and display the Fourier Transform of synthetic images; which are generated by mathematical formulas.

Using the set $[-6 : 0.05 : 6]$ for both the x and y domains, and the set $[-4 : 0.05 : 4]$ for both the fx and fy domains, we generate the first signal using the following function: $s1(x, y) = rect(x) \cdot rect(y)$ (see Figure 2).

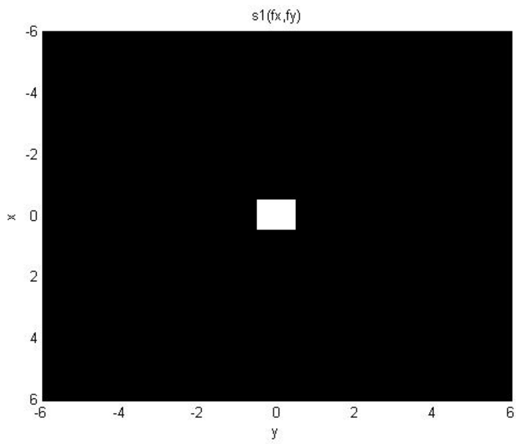

Figure 2: The $2D - rect$ signal.

The obtained results are show in Figure 3 and Figure 4.

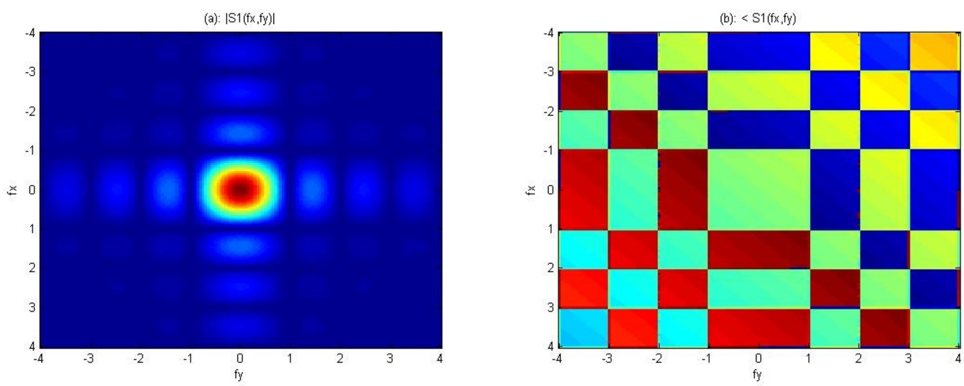

Figure 3: The Fourier Transform of 2D-$rect$ signal (top view); (a) module and (b) phase.

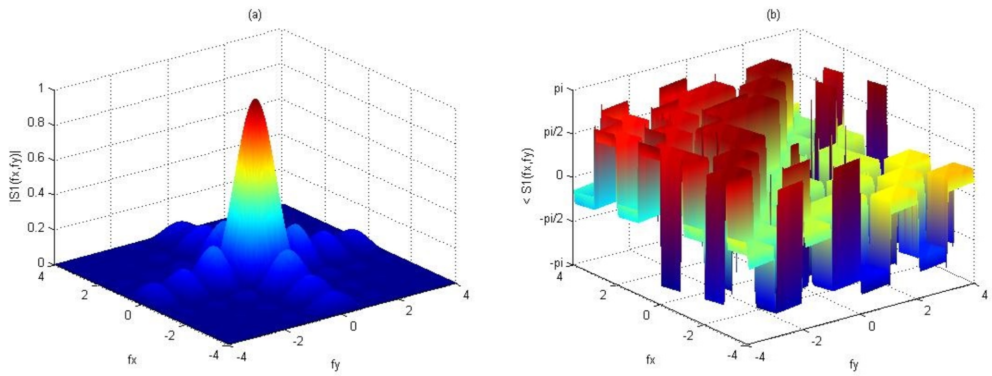

Figure 4: The Fourier Transform of 2D-$rect$ signal; (a) module and (b) phase.

A Fourier Transform of a 2D-$rect$ function is a product of two $sinc$ functions. In Figure 3 is also possible see how the components of low frequency lie in the center of the plotted image, while the high frequency components lie on the edge of image.

In order to understand better the property of two-dimensional Fourier Transform, we try both the scaling and deforming properties on the 2D-$rect$ signal.

So, we generate the new signals: one is the scaled version of the $s1(x, y)$ signal and we call it $s2(x, y) = rect(x/2) \cdot rect(y)$ and another one is the deformed version of the $s2(x, y)$ signal and we call it $s3(x, y) = rect(x/2) \cdot rect(y - x/4)$ (see Figure 5).

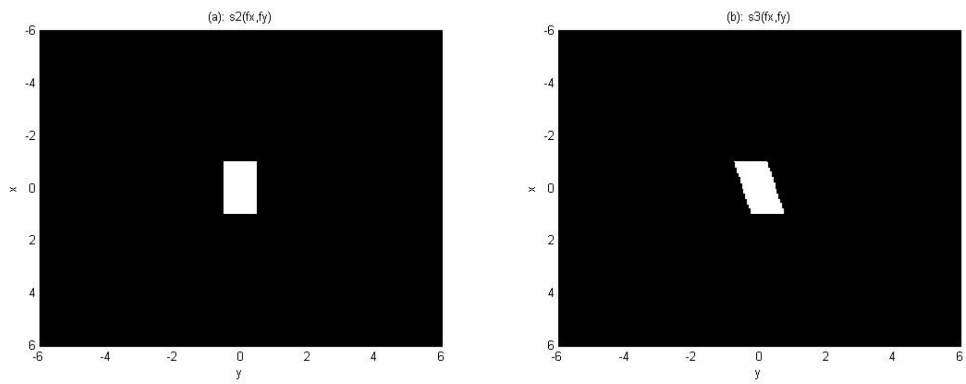

Figure 5: The signals: (a) $s2(x, y)$ and (b) $s3(x, y)$.

The obtained results are show in Figure 6 and Figure 7.

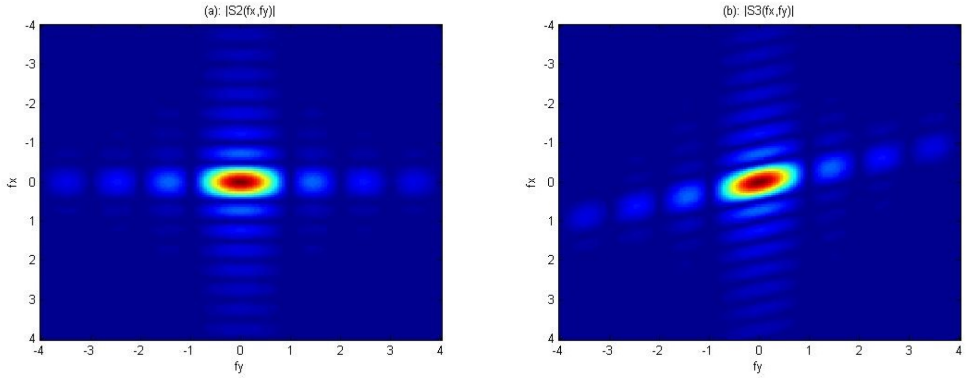

Figure 6: The Fourier Transform of $s2(x, y)$ and $s3(x, y)$ signals (top view); (a) $|S2(f_x, f_y)|$ and (b) $|S3(f_x, f_y)|$.

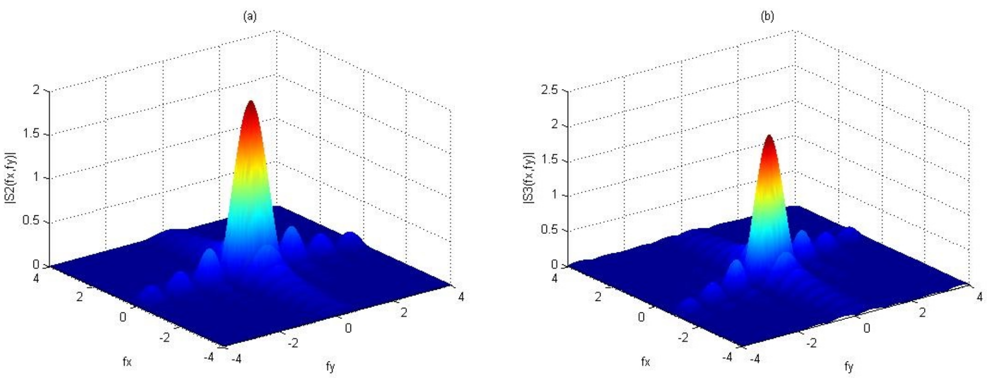

Figure 7: The Fourier Transform of $s2(x, y)$ and $s3(x, y)$ signals; (a) $|S2(f_x, f_y)|$ and (b) $|S3(f_x, f_y)|$.

Compare now the module of the Fourier Transform of the signals $s1(x, y)$ (Figure 3 and Figure 4), $s2(x, y)$ and $s3(x, y)$ (Figure 6 and Figure 7).

We can see that the spectrum of $s2(x, y)$ is tighter than the spectrum of $s1(x, y)$ on f_x direction because the spatial-support of the signal $s1(x, y)$ is wider than the one of the signal $s2(x, y)$ on x direction.

While the spectrum of $s3(x, y)$ is deformed by an affine transformation (is a function between affine spaces which preserves points, straight lines and planes. Also, sets of parallel lines remain parallel after an affine transformation. An affine transformation does not necessarily preserve angles between lines or distances between points, though it does preserve ratios of distances between points lying on a straight line).

It is theoretically possible to demonstrate this concept by introducing the matrix D of the change of the reference system (Equation 3).

$$\begin{vmatrix} x \\ y \end{vmatrix} = \underline{\underline{D}} \cdot \begin{vmatrix} x' \\ y' \end{vmatrix} \tag{3}$$

Since Equation 3 is a linear relationship, the relationship between the coordinate system in frequency domain must be linear as well, like in Equation 4.

$$U(f_x, f_y) = \frac{1}{|\underline{\underline{F}}|} \cdot \tilde{U}(f'_x, f'_y) \tag{4}$$

where $\underline{\underline{F}}^T \cdot \underline{\underline{D}} = \underline{\underline{I}}$ identity matrix.

Now follow other examples of the two-dimensional Fourier Transform of the signals: $s4(x, y) = sin(\pi \cdot x)$, $s5(x, y) = sin(\pi \cdot y)$ and $s6(x, y) = sin(\pi \cdot x - 2\pi \cdot y)$ (Figure 8).

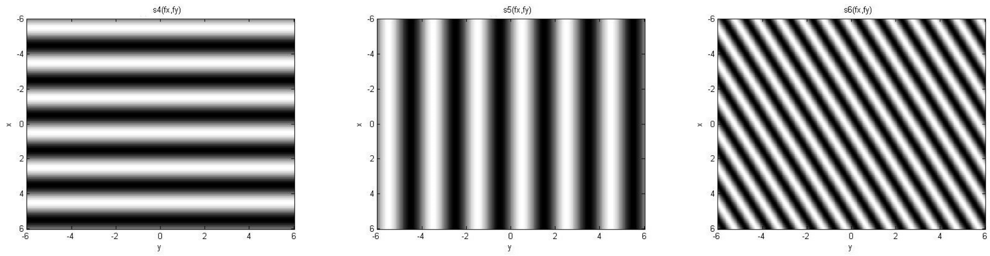

Figure 8: The signals: $s4(x, y)$, $s5(x, y)$, and $s6(x, y)$.

The obtained results are show in Figure 9.

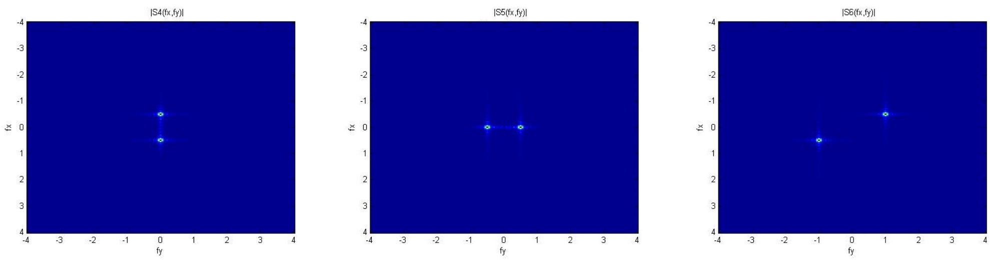

Figure 9: The Fourier Transform of $s4(x, y)$, $s5(x, y)$, and $s6(x, y)$ signals.

3 Real Images

In this last section we compute the Fourier Transforms of real images using the function implemented in the previous point; after that we compare the spectrum of a natural image with the ones of periodic pattern texture images.

Before applying the Fourier Transform we must convert the color space of image from RGB to YC_bC_r; where Y is the luminance component while CB and CR are the blue-difference and red-difference chroma components. The original image and the Y component are shown in Figure 10.

Figure 10: The real (natural) reference image and its Y luminance component.

The Fourier Transform is used to access the geometric characteristics of an image into its spatial domain. This because the image in the Fourier domain is decomposed into its sinusoidal components; it is easy to examine or process certain frequencies of the image, thus influencing the geometric structure in the spatial domain.

We start off by applying the Fourier Transform to Y, the luminance component. The obtained result are show in Figure 11.

Figure 11: The Fourier Transform of Y luminance component of the image.

We can see that the DC-value is by far the largest component of the image. However, the dynamic range of the our spectrum (i.e. the intensity values in the image of spectrum) is too large to be displayed on the screen, therefore all other values appear as black.

If instead, we apply a logarithmic transformation[1] to the image we obtain the result in the Figure 12.

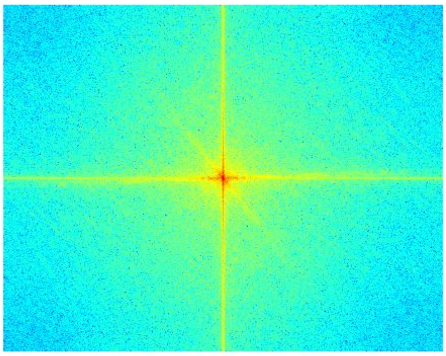

Figure 12: The Fourier Transform of Y luminance component of the image (logarithmic scale).

The result shows that the spectrum (magnitude) contains components of all frequencies, but that their magnitude decreases as the inverse of frequency. Hence, low frequencies contain more image information than the higher ones.

It is possible to see that there are two dominating directions in the spectrum, one passing vertically and one horizontally through the center.

While the magnitude is for us important, the phase does not yield much new information about the structure of image into the spatial domain; so, in this and the following examples, we restrict ourselves to displaying only the magnitude of the Fourier Transform.

Now we compute the Fourier Transform of some simple images to better understand the nature of the transform, in particular we are interested in the Fourier Transform to periodic patterns in the spatial domain images. This can be seen very easily comparing the spectrum of the previous natural image with ones of periodic pattern texture images, called i_1, i_2 and i_3. These are show in the Figure 13.

Like in the previous case to view all the frequency components we

[1] $Q(i,j) = c \cdot log(1 + |P(i,j)|)$ where $c = \frac{255}{log(1+max(P(i,j))}$

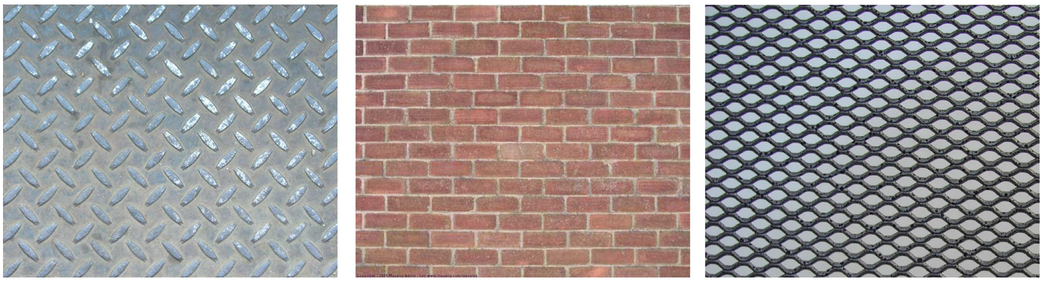

Figure 13: The periodic pattern texture images: i_1, i_2 and i_3.

apply a logarithmic transformation to the magnitude of the Fourier Transform. The obtained result are show in Figure 14.

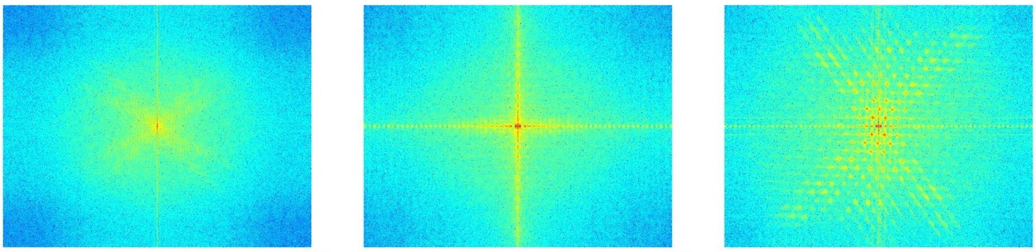

Figure 14: The Fourier Transform of periodic pattern texture images: I_1, I_2 and I_3.

From the theory we know that the periodicity in the spatial domain implies the sampling in the frequencies domain; in fact we can see how the most of the energy is localized into some components in the frequencies domain. Obviously our images are not perfectly periodic images so there are some other frequencies in the spectrum.

In the Figure 15 and the Figure 16, we propose another example in order to clarify better this concept.

In this case is more evident that the spectrum of a periodic pattern image is the sampled version of the spectrum of the original image.

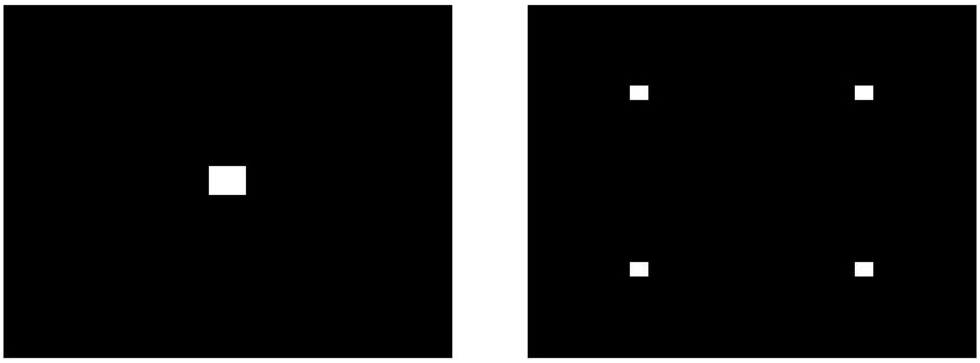

Figure 15: The synthetic images: the 2D-$rect$ signal and its periodic version.

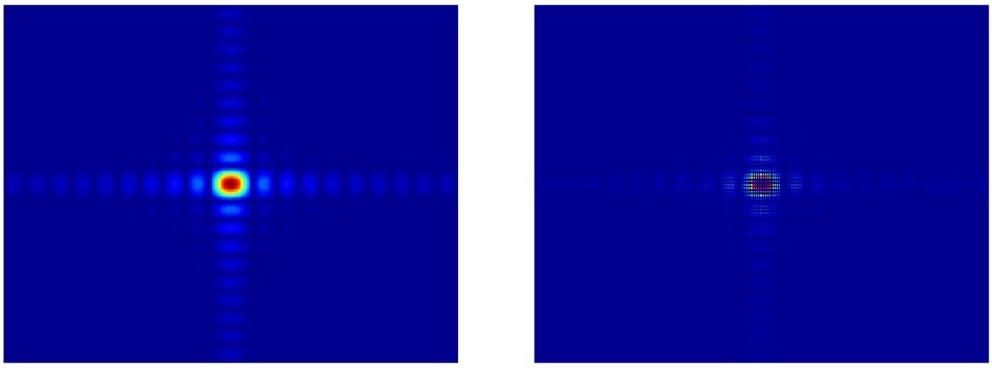

Figure 16: The Fourier Transform of the previous synthetic images.