SEL 0449 - Processamento Digital de Imagens Médicas

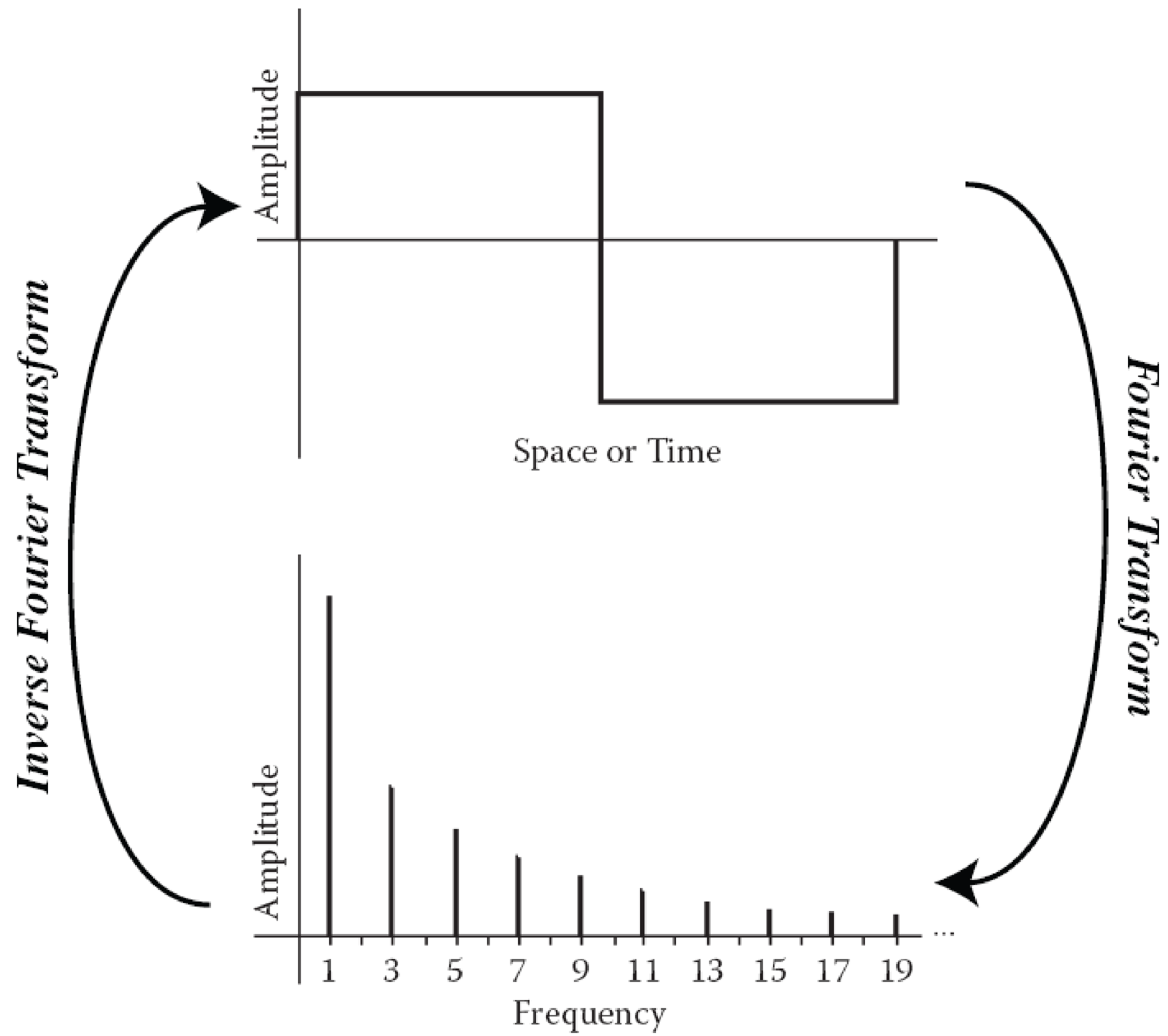
SEL 5895 – Introdução ao Processamento Digital de Imagens

Aula 6 – Propriedades da Transformada de Fourier

Prof. Dr. Marcelo Andrade da Costa Vieira

mvieira@sc.usp.br

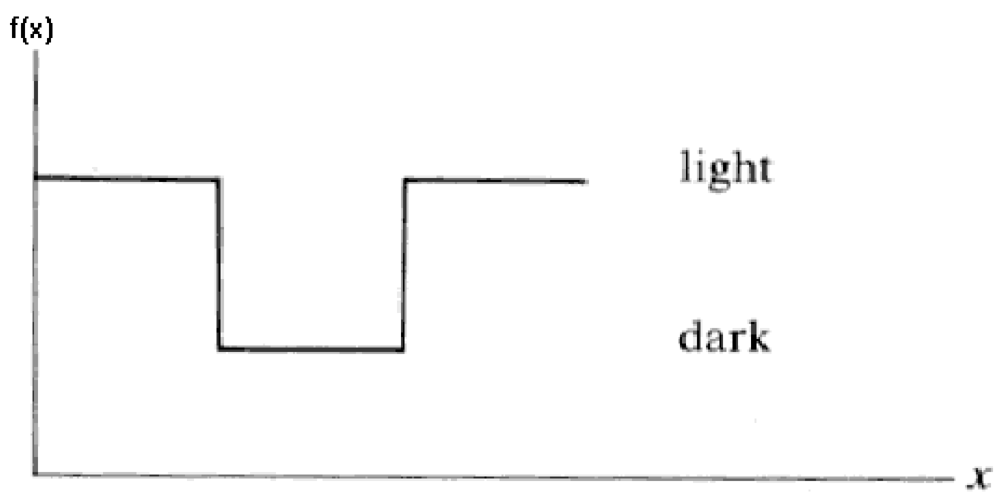
Transformada de Fourier



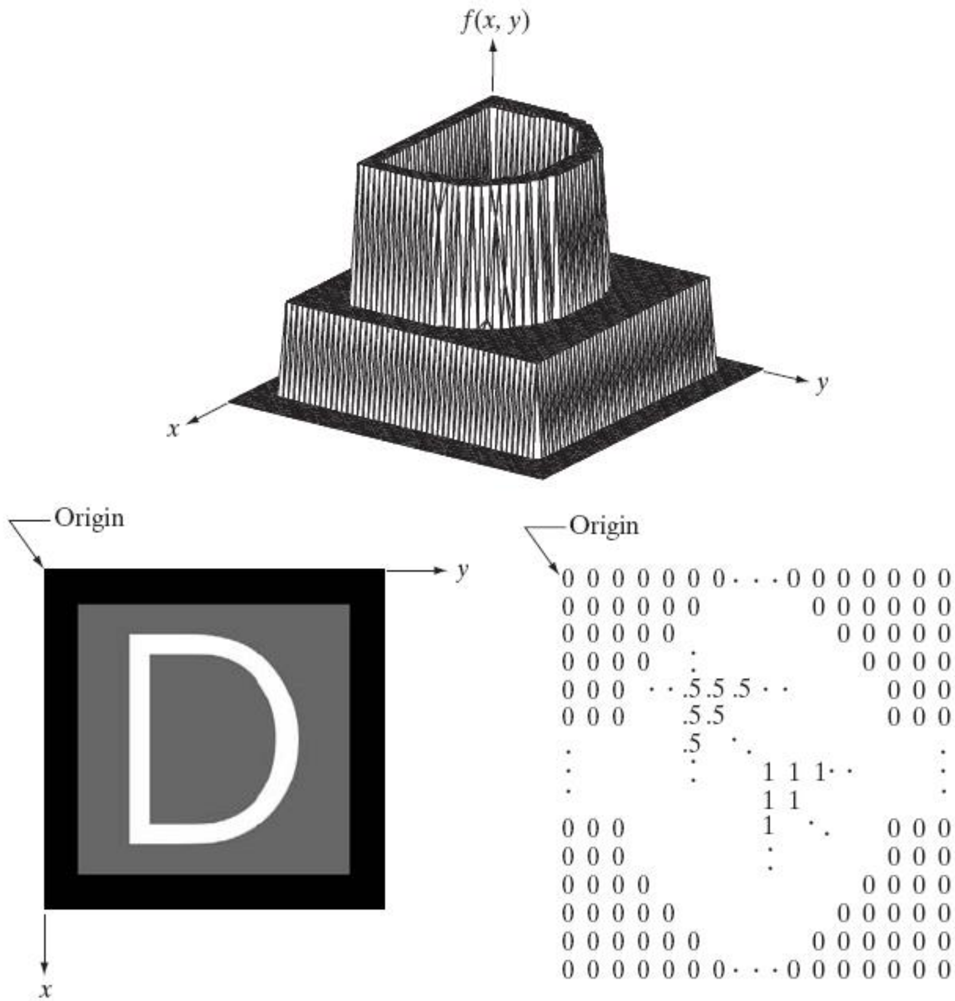
E em uma Imagem?

Uma linha de uma imagem formada por uma sequência de pixels de diferentes intensidades:

Pode ser representada no domínio do espaço como uma forma de onda:

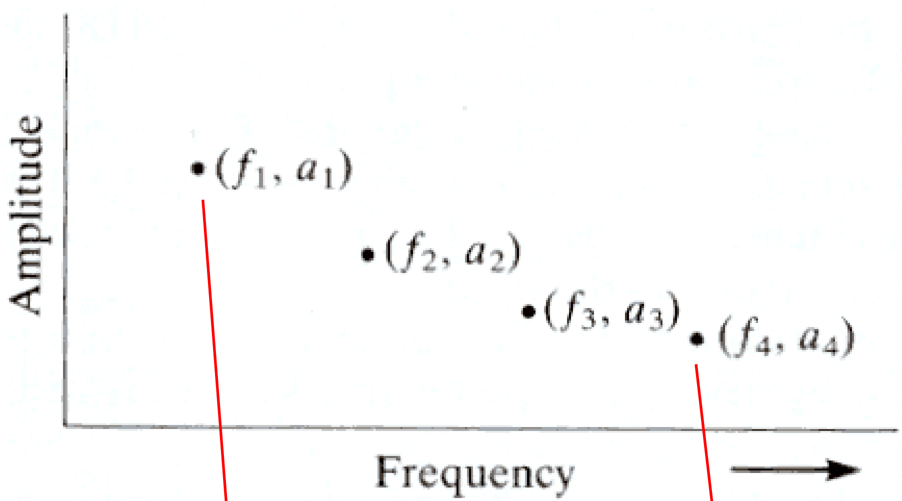


Representação de uma Imagem como uma função bidimensional



Domínio da frequência

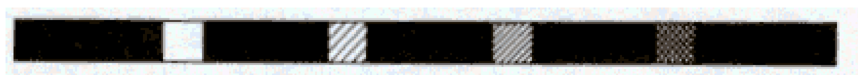
E no **Domínio da Frequência** pode ser representada por uma soma de senos e cossenos, através de suas frequências (f) e amplitudes (a):



Que podem ser colocadas no formato de uma imagem como uma linha de amplitudes em escala de cinza.

Transformada inversa

Diz-se então, que a imagem gerada através das amplitudes das frequências:



É a **Transformada no domínio da frequência** da imagem original dada no domínio do espaço:

É possível aplicar sobre a imagem no domínio da frequência, uma **Transformada Inversa**, obtendo a Imagem original.

Propriedades da DFT 2-D

1) Periodicidade e Simetria Conjugada

	Domínio do espaço*		Domínio da frequência*
1	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2	$f(x, y)$ imaginária	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ par; $I(u, v)$ ímpar
4	$f(x, y)$ imaginária	$\Leftrightarrow$	$R(u, v)$ ímpar; $I(u, v)$ par
5	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complexa
6	$f(-x, -y)$ complexa	$\Leftrightarrow$	$F(-u, -v)$ complexa
7	$f^*(x, y)$ complexa	$\Leftrightarrow$	$F^*(-u, -v)$ complexa
8	$f(x, y)$ real e par	$\Leftrightarrow$	$F(u, v)$ real e par
9	$f(x, y)$ real e ímpar	$\Leftrightarrow$	$F(u, v)$ imaginária e ímpar
10	$f(x, y)$ imaginária e par	$\Leftrightarrow$	$F(u, v)$ imaginária e par
11	$f(x, y)$ imaginária e ímpar	$\Leftrightarrow$	$F(u, v)$ real e ímpar
12	$f(x, y)$ complexa e par	$\Leftrightarrow$	$F(u, v)$ complexa e par
13	$f(x, y)$ complexa e ímpar	$\Leftrightarrow$	$F(u, v)$ complexa e ímpar

Propriedades da DFT 2-D

1) Periodicidade e Simetria Conjugada

A transformada discreta de Fourier (DFT) e sua inversa são periódicas:

$$\boxed{F(u,v) = F(u+N,v) = F(u,v+N) = F(u+N,v+N)}$$

Sendo N a dimensão da imagem.

Se f(x,y) for real, a DFT apresenta simetria conjugada:

$$\boxed{F(u,v) = F^*(-u,-v)} \quad \text{ou} \quad \boxed{|F(u,v)| = |F(-u,-v)|}$$

Exemplo unidimensional

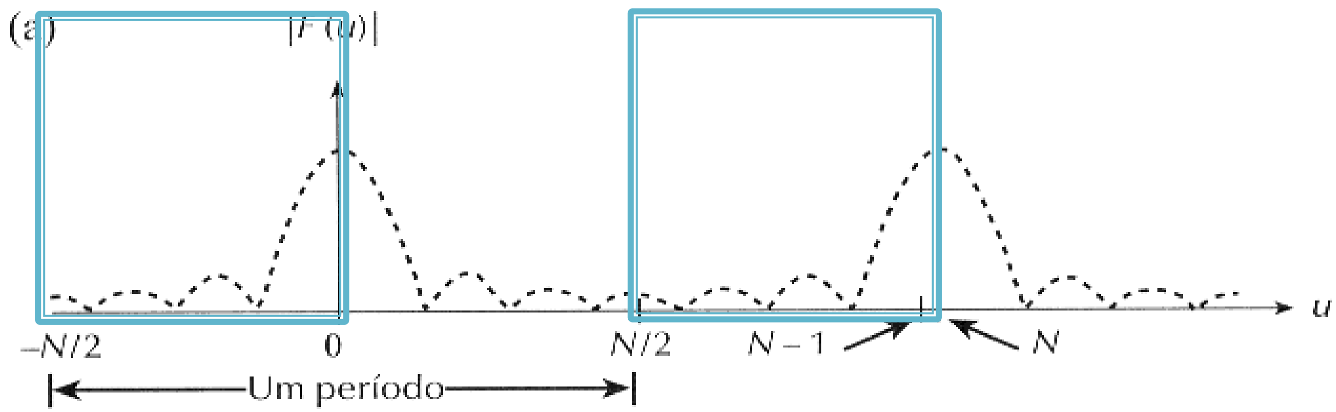
1) Periodicidade e Simetria Conjugada

$$F(u) = F(u+N)$$

$$|F(u)| = |F(-u)|$$

Magnitude centrada na origem

Reflexões



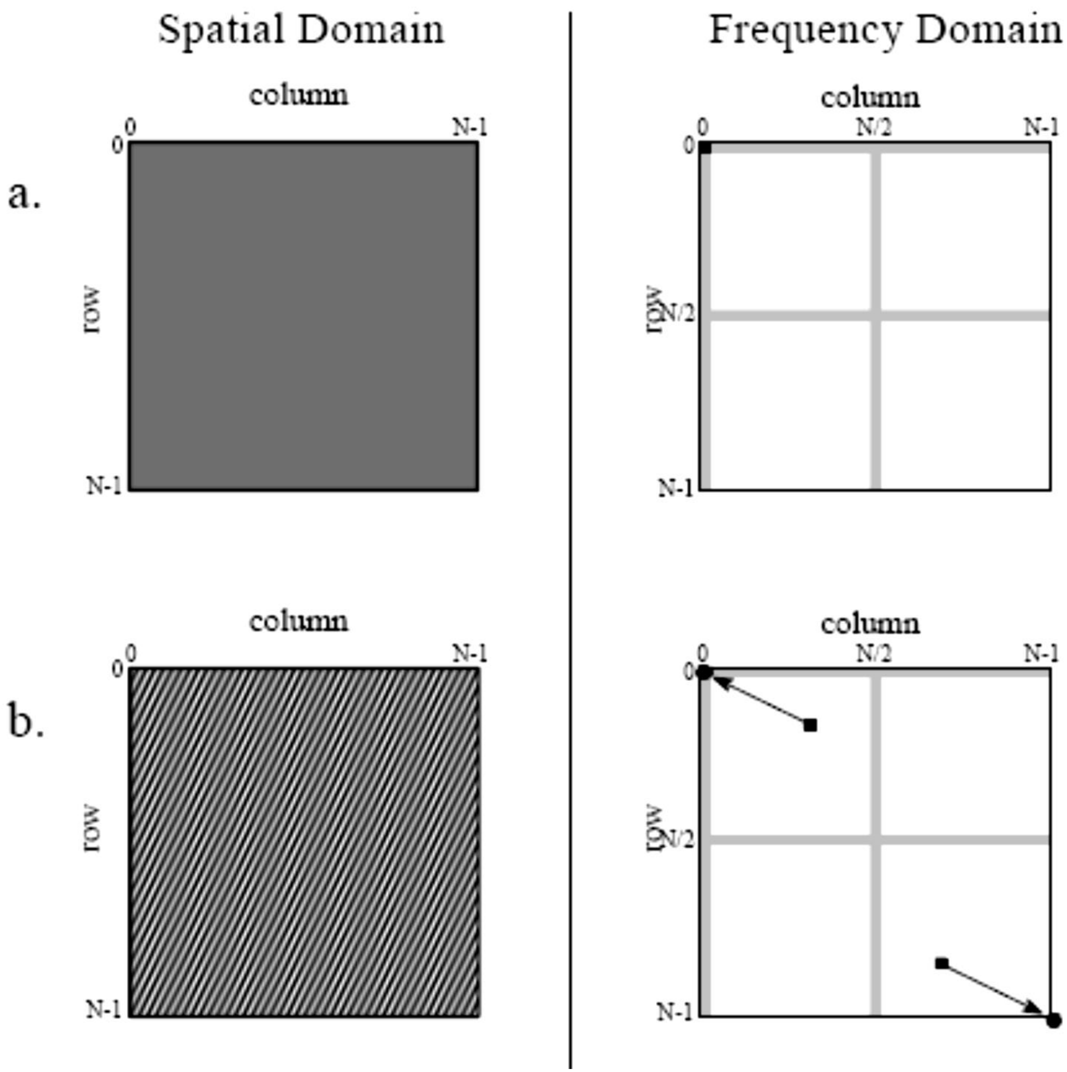
A transformada é formulada para valores de u no intervalo [0, N-1]

Espectro de Fourier 2-D (imagem)

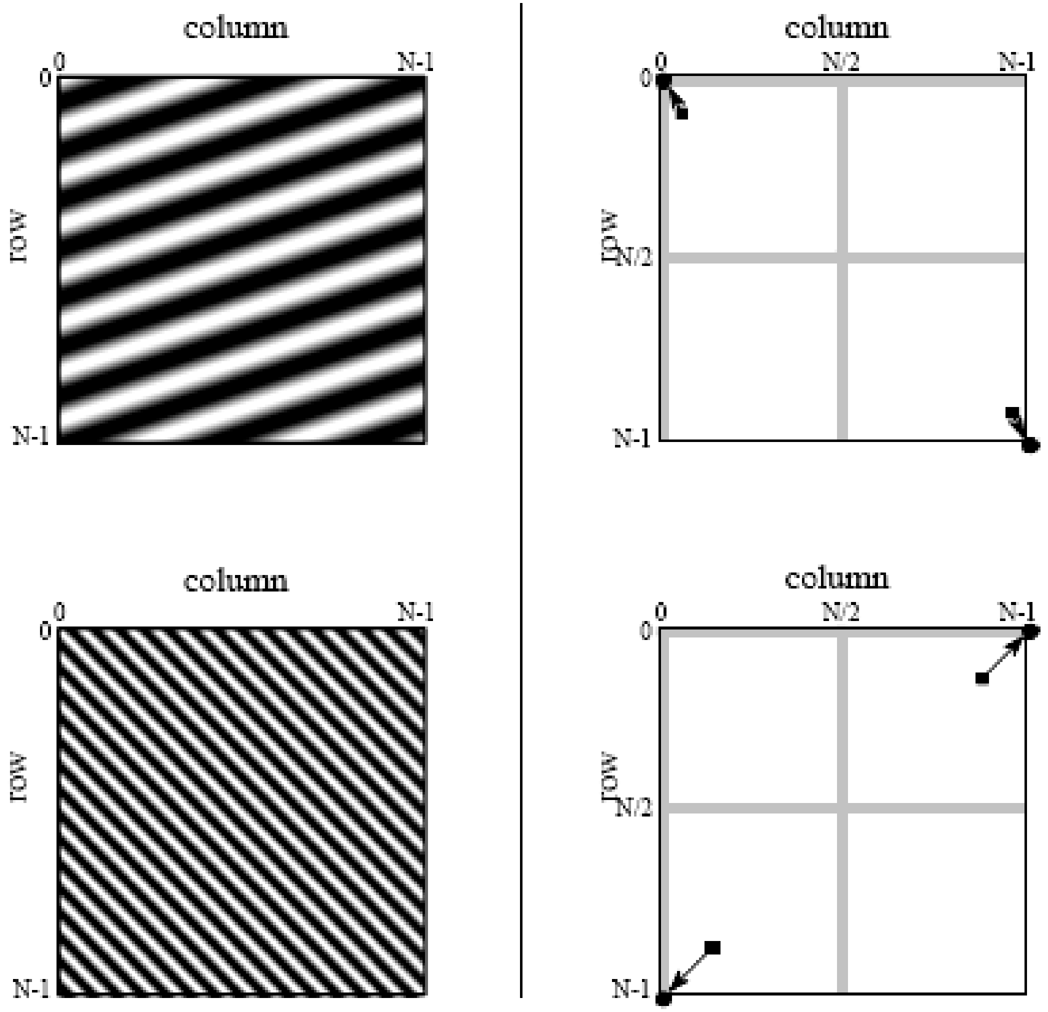
- Para uma função unidimensional, o espectro de Fourier fornece informação (frequência, amplitude e fase) sobre as senóides (1D) que devem ser somadas para formar a função desejada;

- Para uma função bidimensional, o espectro de Fourier fornece informação (frequência, amplitude, fase e direção) sobre as ondas senoidais (2D) que devem ser somadas para formar a função desejada;

Espectro de Fourier Bidimensional (imagem)



Espectro de Fourier Bidimensional (imagem)



Propriedades da Transformada 2-D

2) Translação

Multiplicar f(x,y) pelo termo exponencial, conforme abaixo, e fazer a transformada deste produto, resulta em um deslocamento da origem do plano das frequências para o ponto (u_0,v_0).

$$f(x,y)\exp\left[j2\pi\left(u_0x+v_0y\right)/N\right]\Leftrightarrow F(u-u_0,v-v_0)$$

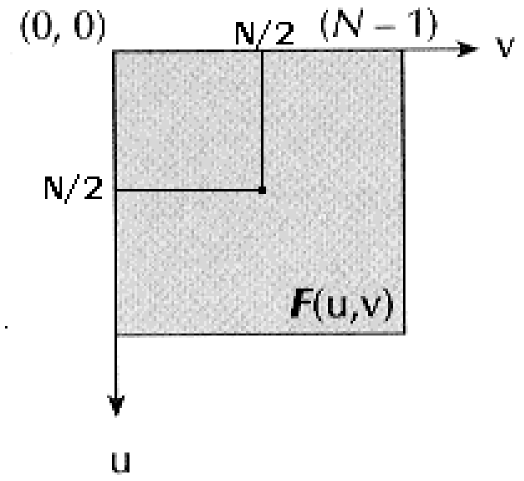
Multiplicar F(u,v) pelo termo exponencial, conforme abaixo, e fazer a Transformada Inversa deste produto, resulta em um deslocamento da origem do plano espacial para o ponto (x_0,y_0).

$$f(x-x_0,y-y_0)\Leftrightarrow F(u,v)\exp\left[-j2\pi(ux_0+vy_0)/N\right]$$

Propriedades da Transformada 2-D

Fazendo $u_0 = v_0 = N/2$ a origem da transformada de Fourier de f(x,y) pode ser movida para o centro do quadrado de frequências N x N.

$$\exp\left[j2\pi (u_0 x + v_0 y) / N \right] = e^{j\pi (x+y)} = \cos \pi + jsen\pi = (-1)^{x+y}$$



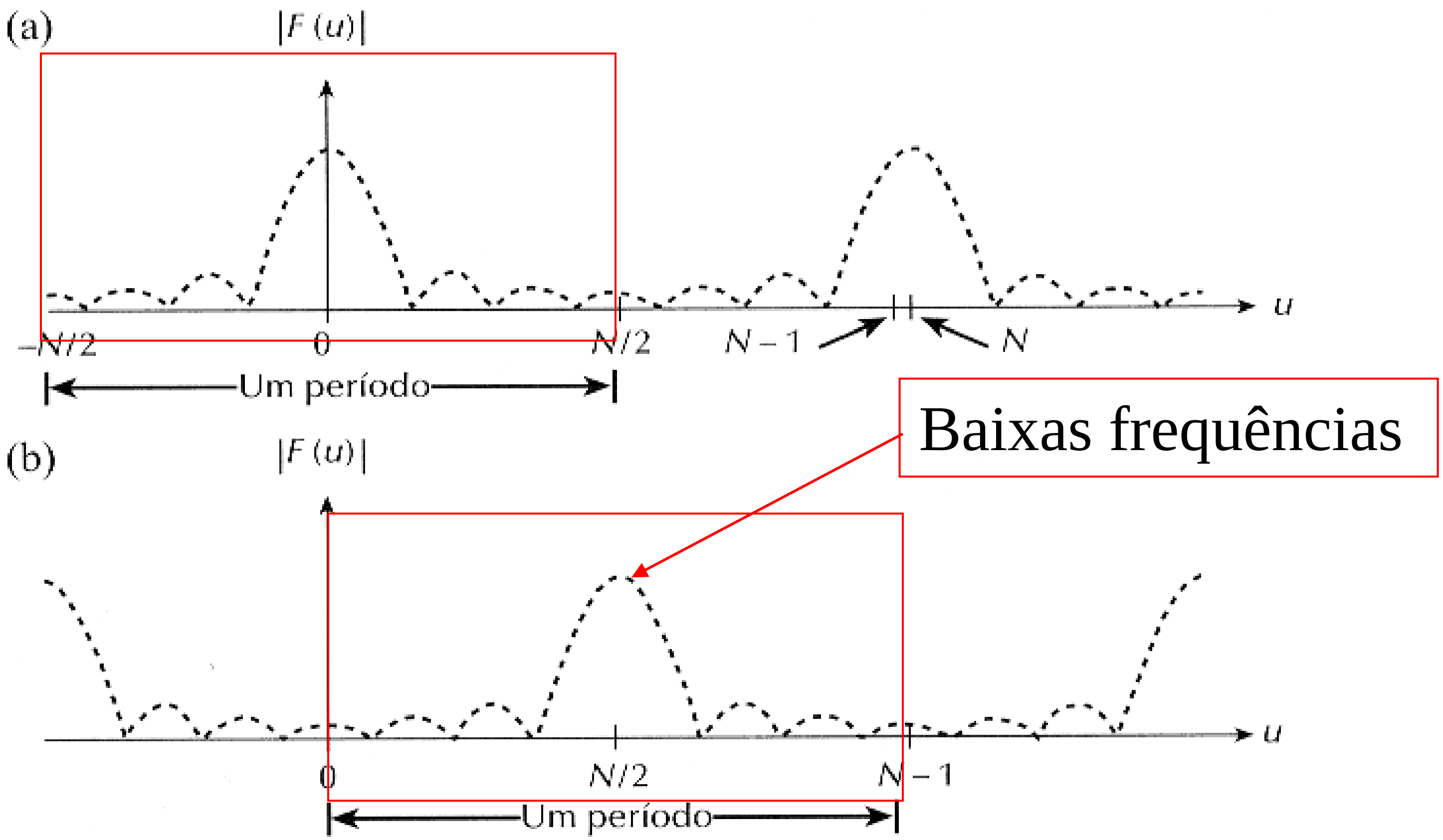
Ou seja:
Multiplicar *f(x,y)* por $(-1)^{x+y}$ e realizar a transformada de Fourier, simplesmente muda a origem das frequências para o centro do quadrado.

A magnitude da Ttansformada não é afetada:

$$\left| F(u,v) \exp\left[-j2\pi (ux_0 + vy_0) / N \right] \right| = \left| F(u,v) \right|$$

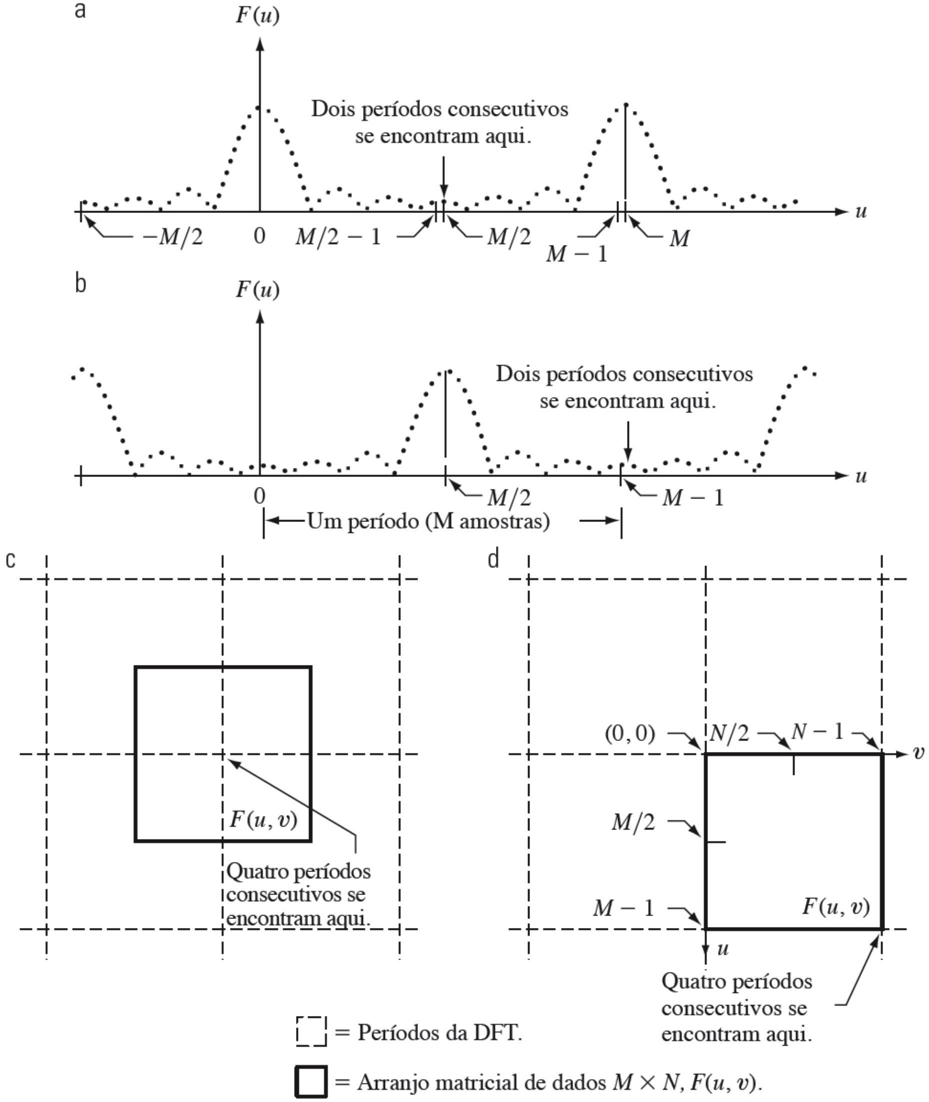
Exemplo unidimensional

Para exibir um período inteiro, basta mover a origem da transformada para o ponto $u = N/2$



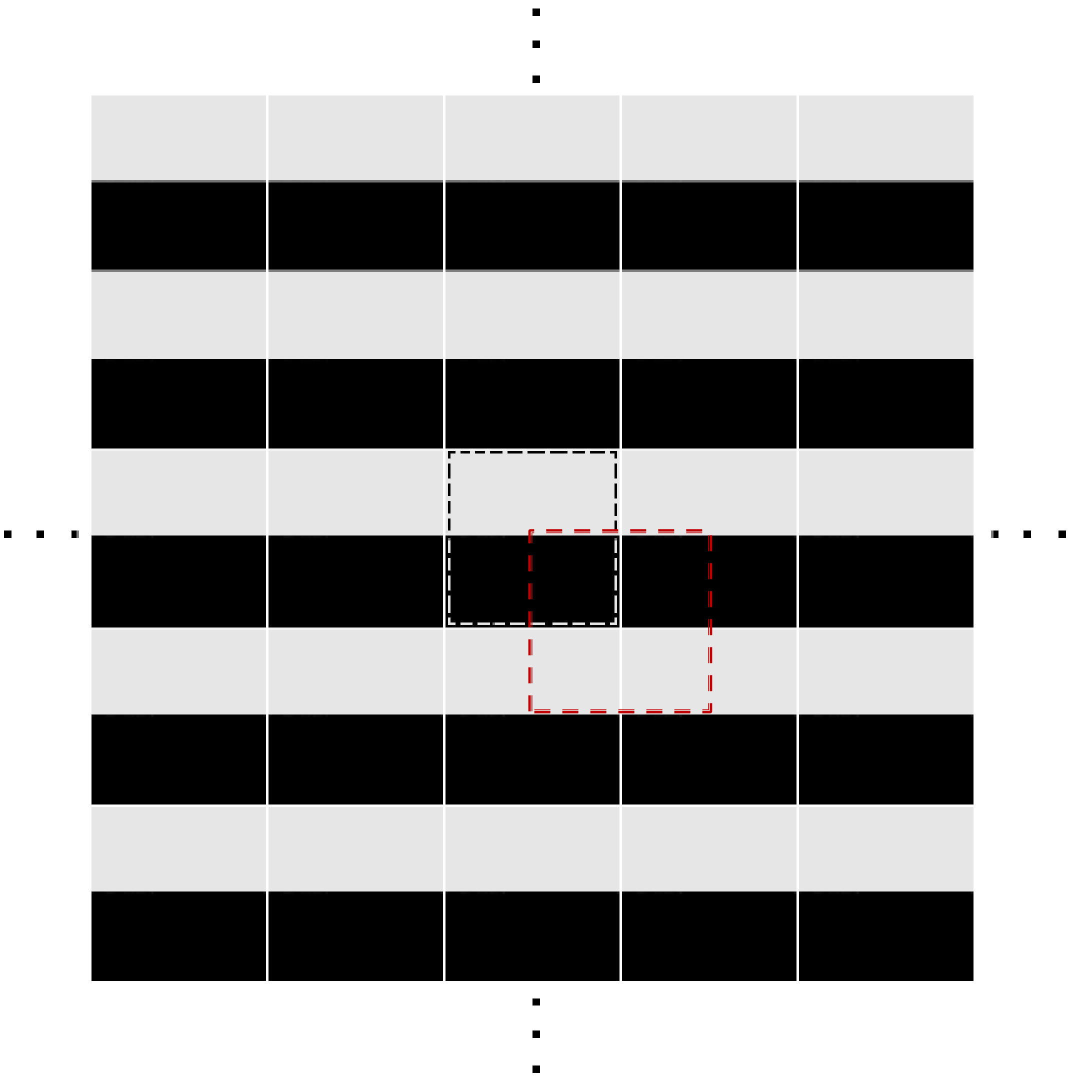
Espectro de Fourier Bidimensional (imagem)

Frequência Zero deslocada para o centro



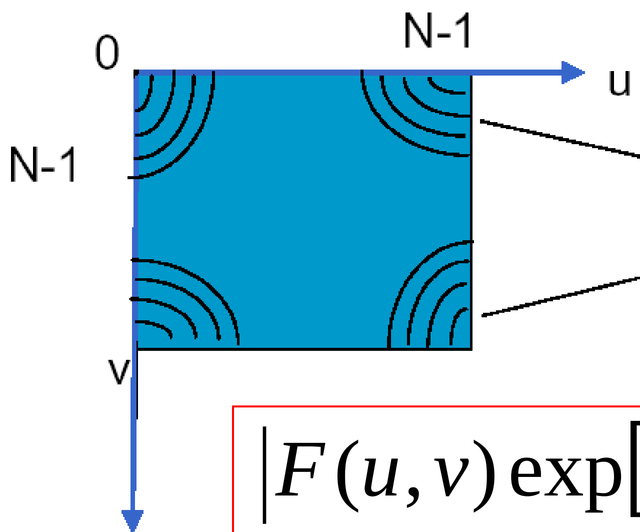
Espectro de Fourier Bidimensional (imagem)

Frequência Zero deslocada para o centro



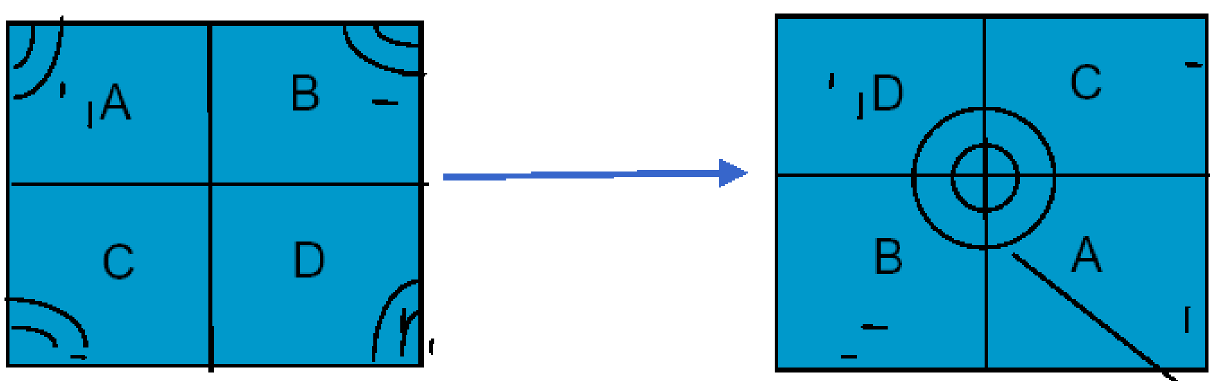
Espectro de Fourier Bidimensional (imagem)

Frequência Zero deslocada para o centro



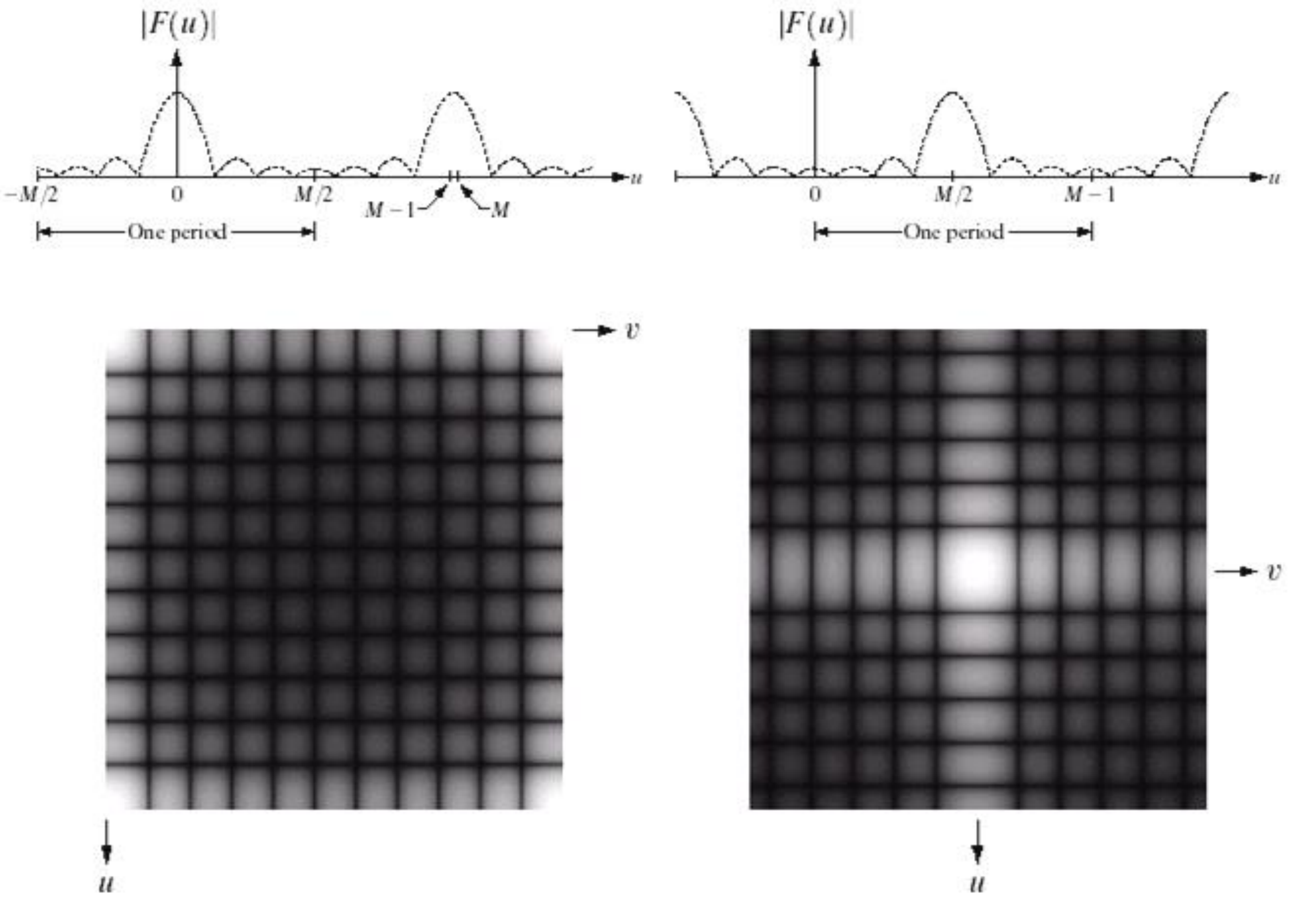
Baixas Frequências

$$\left|F(u,v)\exp\left[-j2\pi(ux_0+vy_0)/N\right]\right|=\left|F(u,v)\right|$$

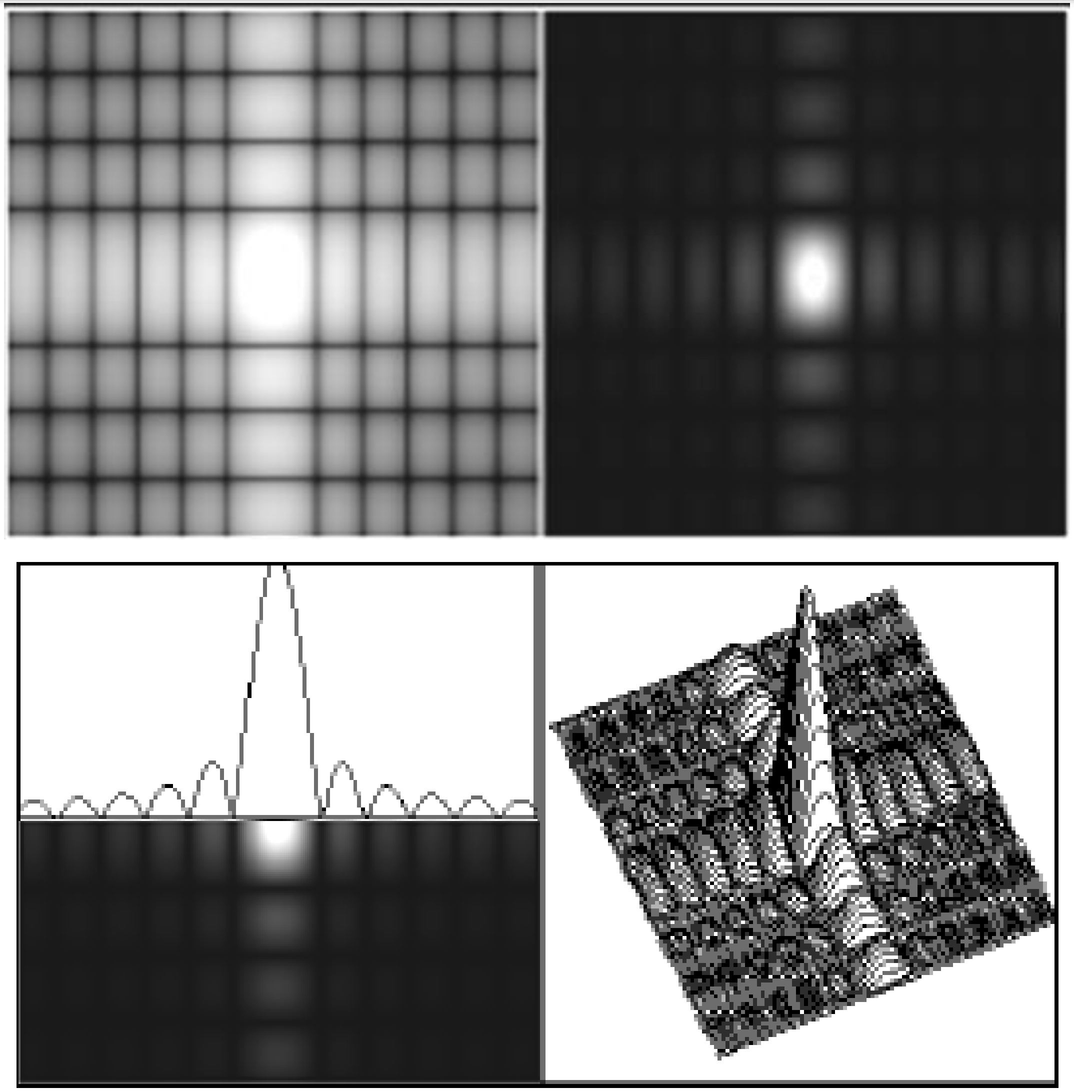


Baixas Frequências

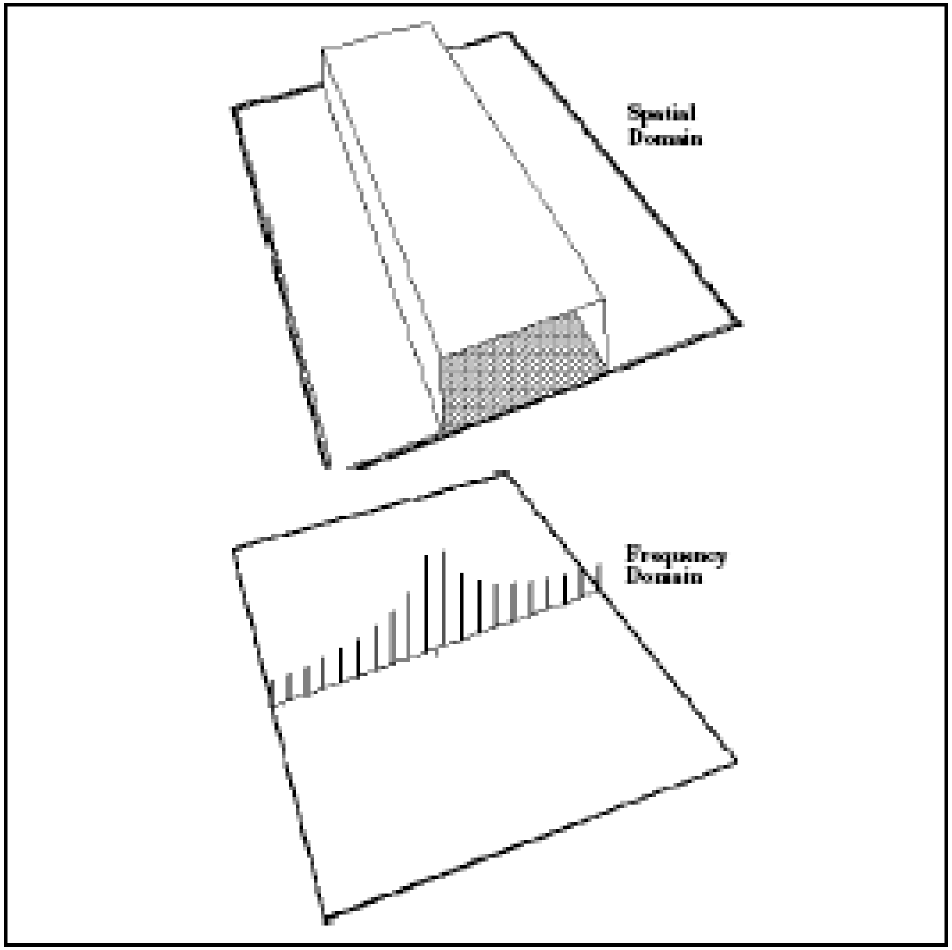
Espectro Unidimensional e Bidimensional



Espectro Unidimensional e Bidimensional



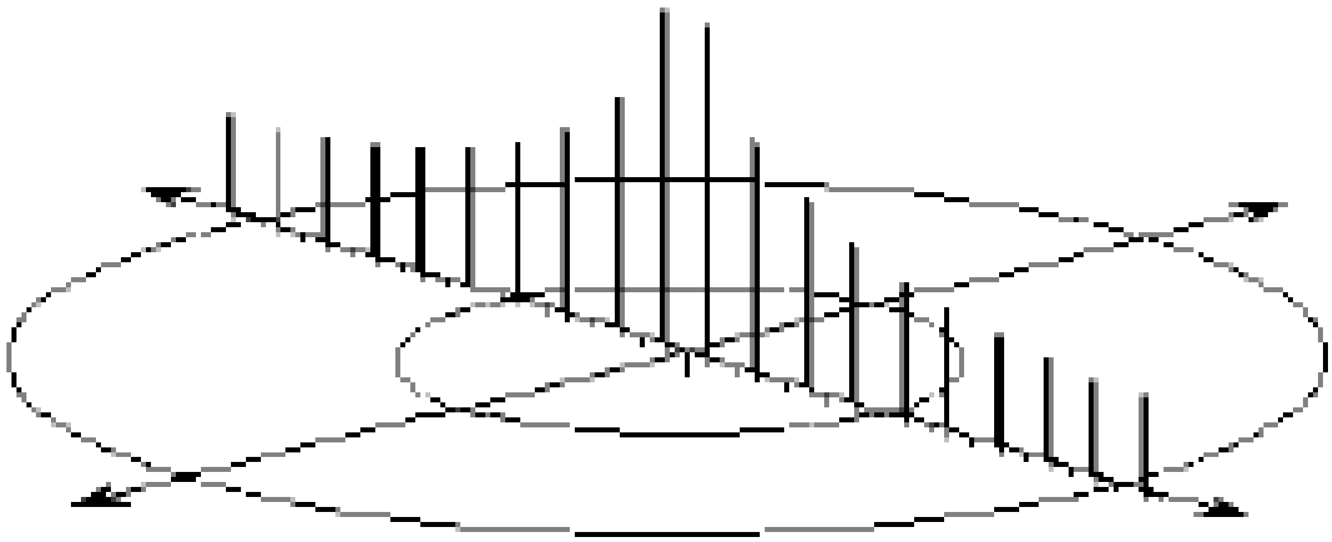
Exemplos:



Padrão com variação de freqüência em apenas uma direção (x). Nas outras direções a frequência é zero.

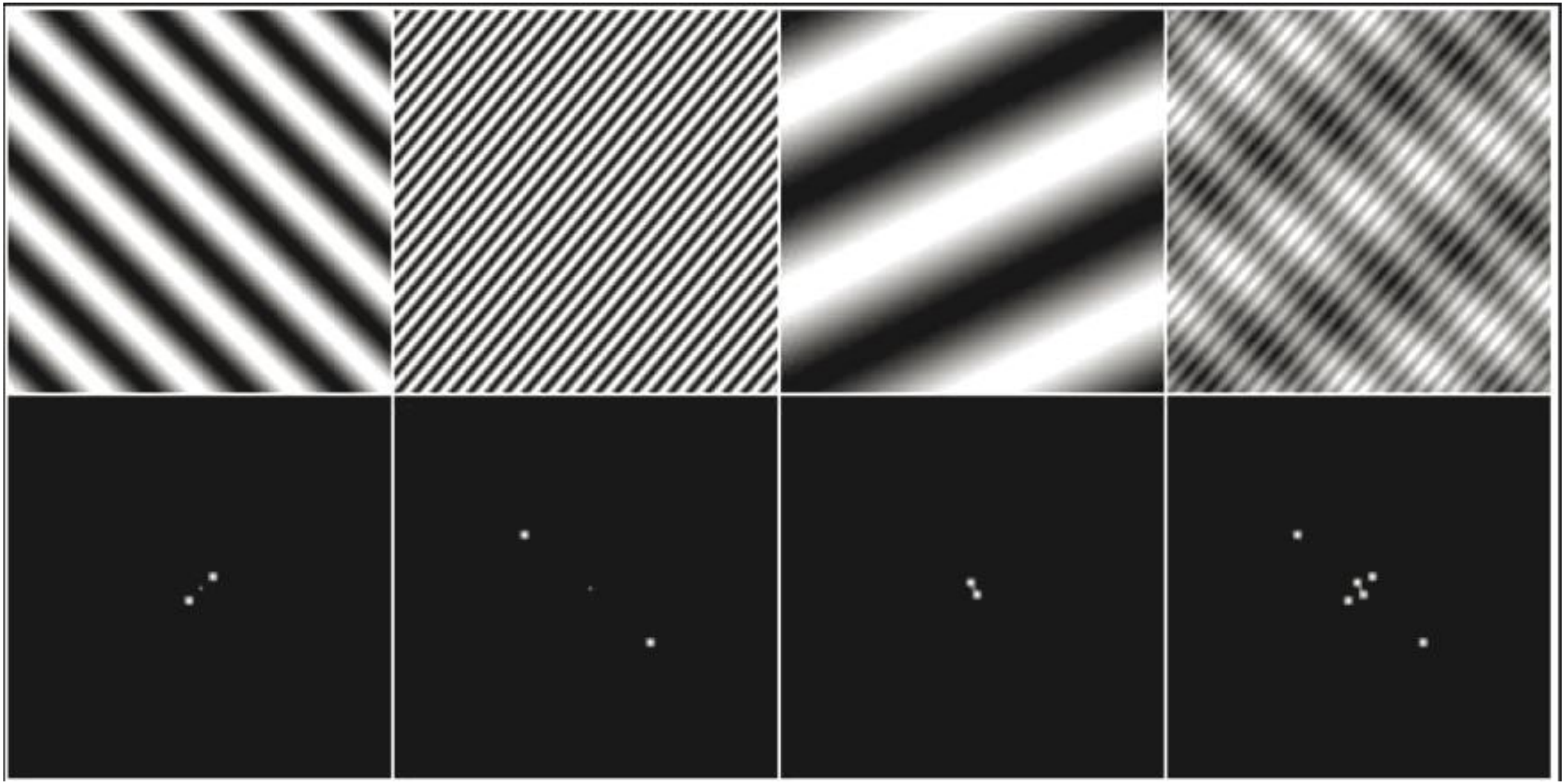
Espectro de Fourier Bidimensional (imagem)

Frequência Zero deslocada para o centro



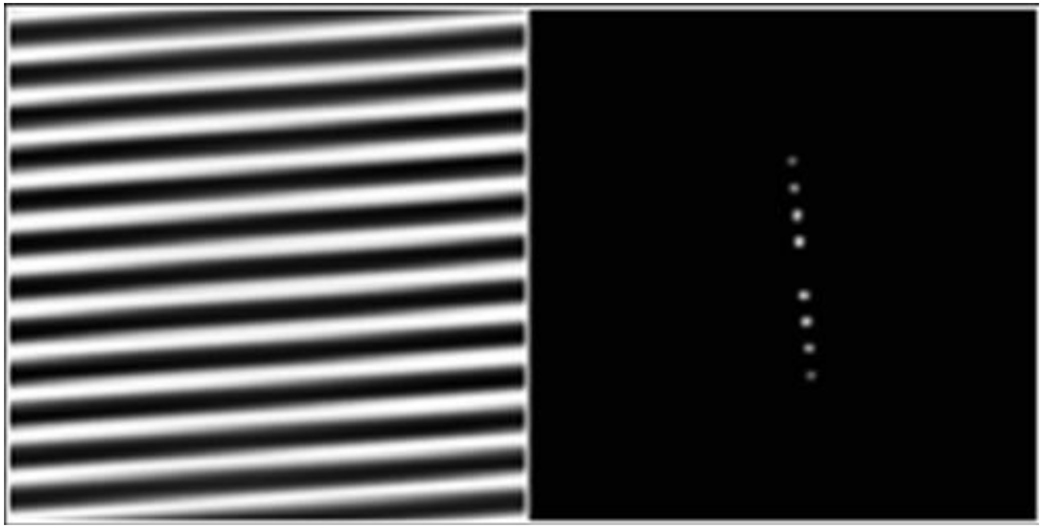
Espectro de Fourier Bidimensional (imagem)

Frequência Zero deslocada para o centro



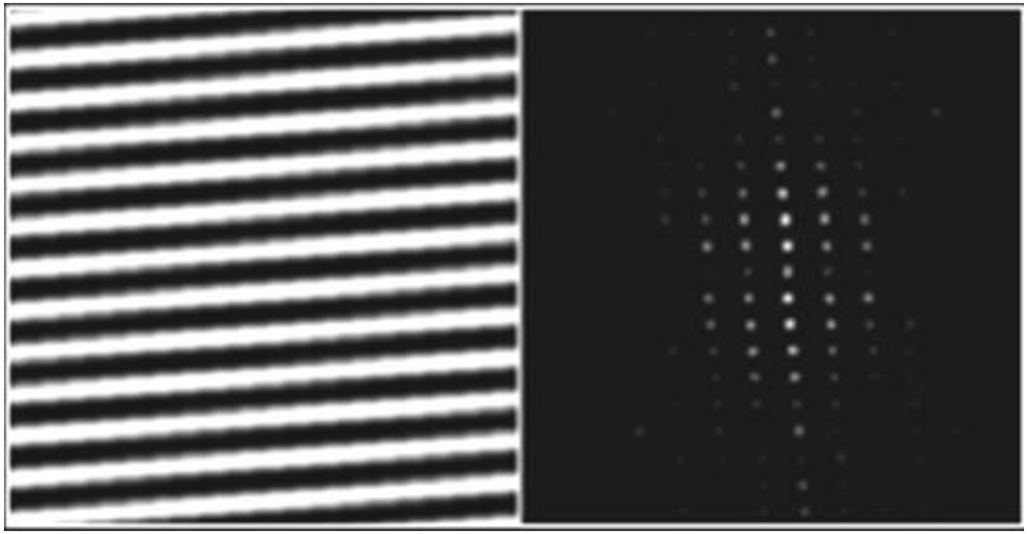
Exemplos de Texturas:

Padrão Senoidal

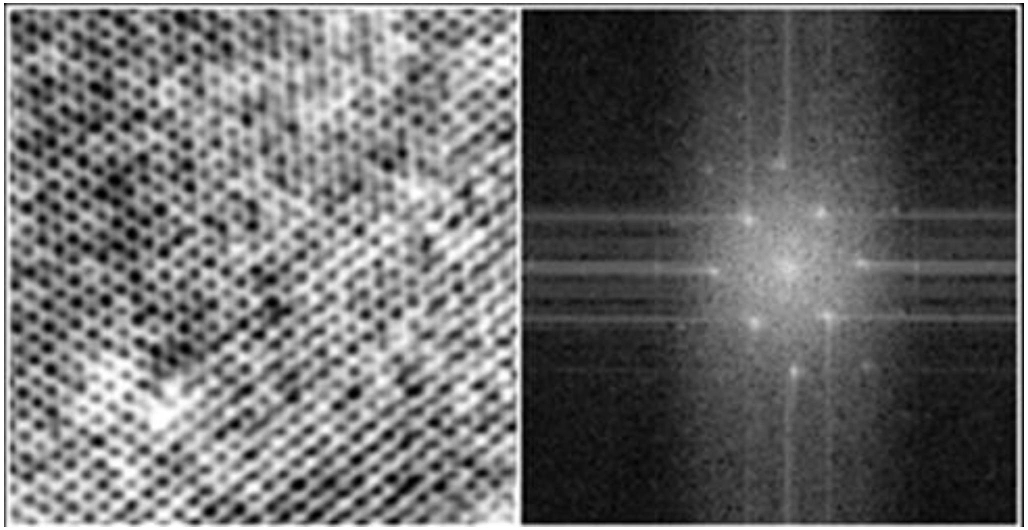


Padrão Não Senoidal

Exemplos de Texturas:



Padrão Não Senoidal com interferências em outras direções



Textura

Espectro de Fourier Bidimensional (Vídeo)

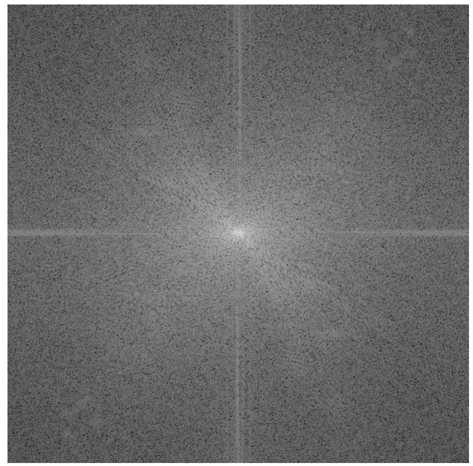
https://youtu.be/D9ziTuJ3OCw

Visualização do espectro

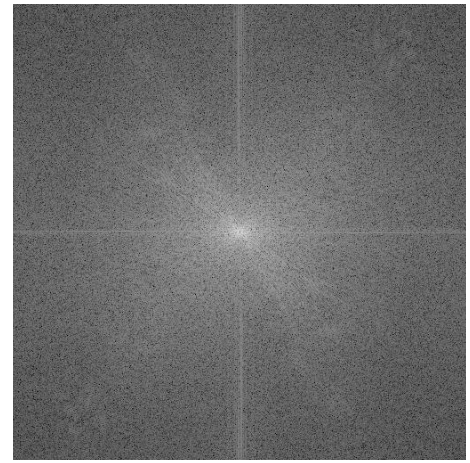
Forma Retangular X Forma Polar

$a_1, a_2, a_3 \ldots$
Amplitudes dos Cossenos

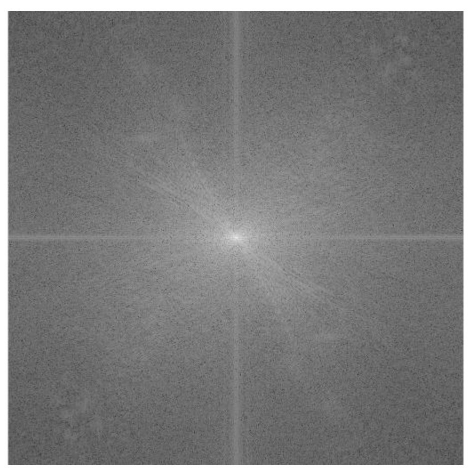
Parte Real



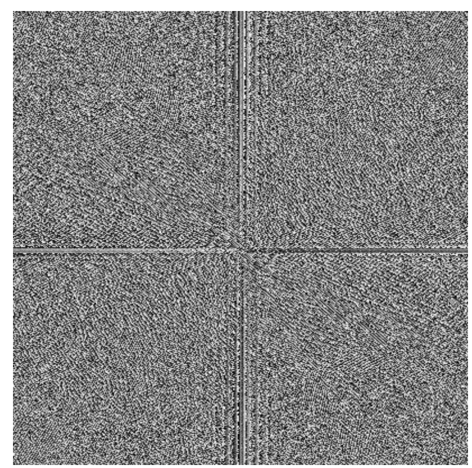
Parte Imaginária



$b_1, b_2, b_3 \ldots$
Amplitudes dos Senos

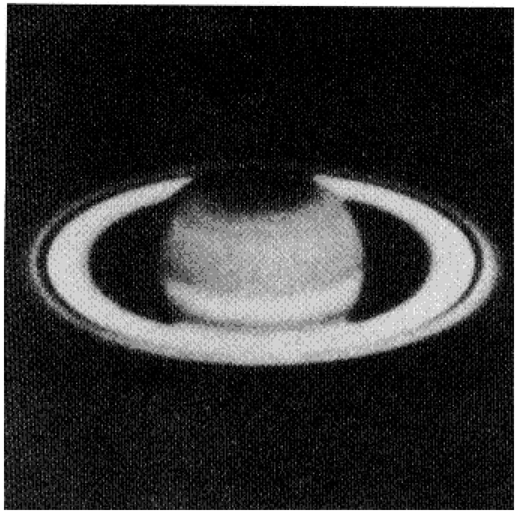


Módulo (Espectro)



Fase

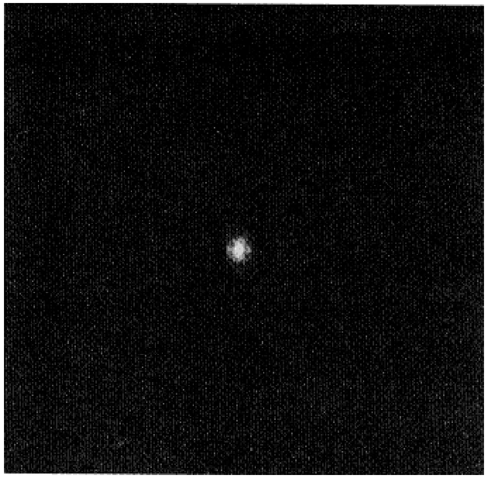
Visualização do espectro



A escala dinâmica dos espectros de Fourier mostrados como funções de Intensidades, são geralmente muito mais alta do que a capacidade de reprodução dos dispositivos.

Uma técnica útil é comprimir a escala através da exibição de uma função logaritmo.

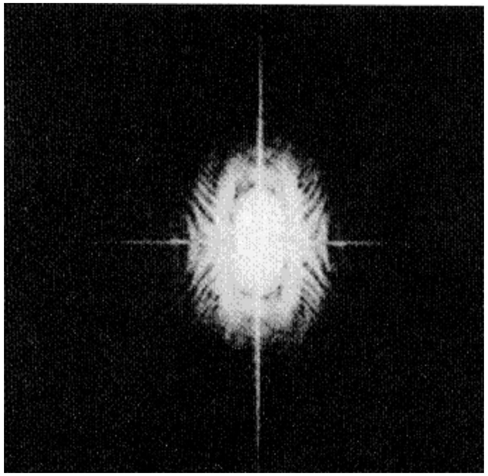
$$D(u,v) = c\log[1+|F(u,v)|]$$



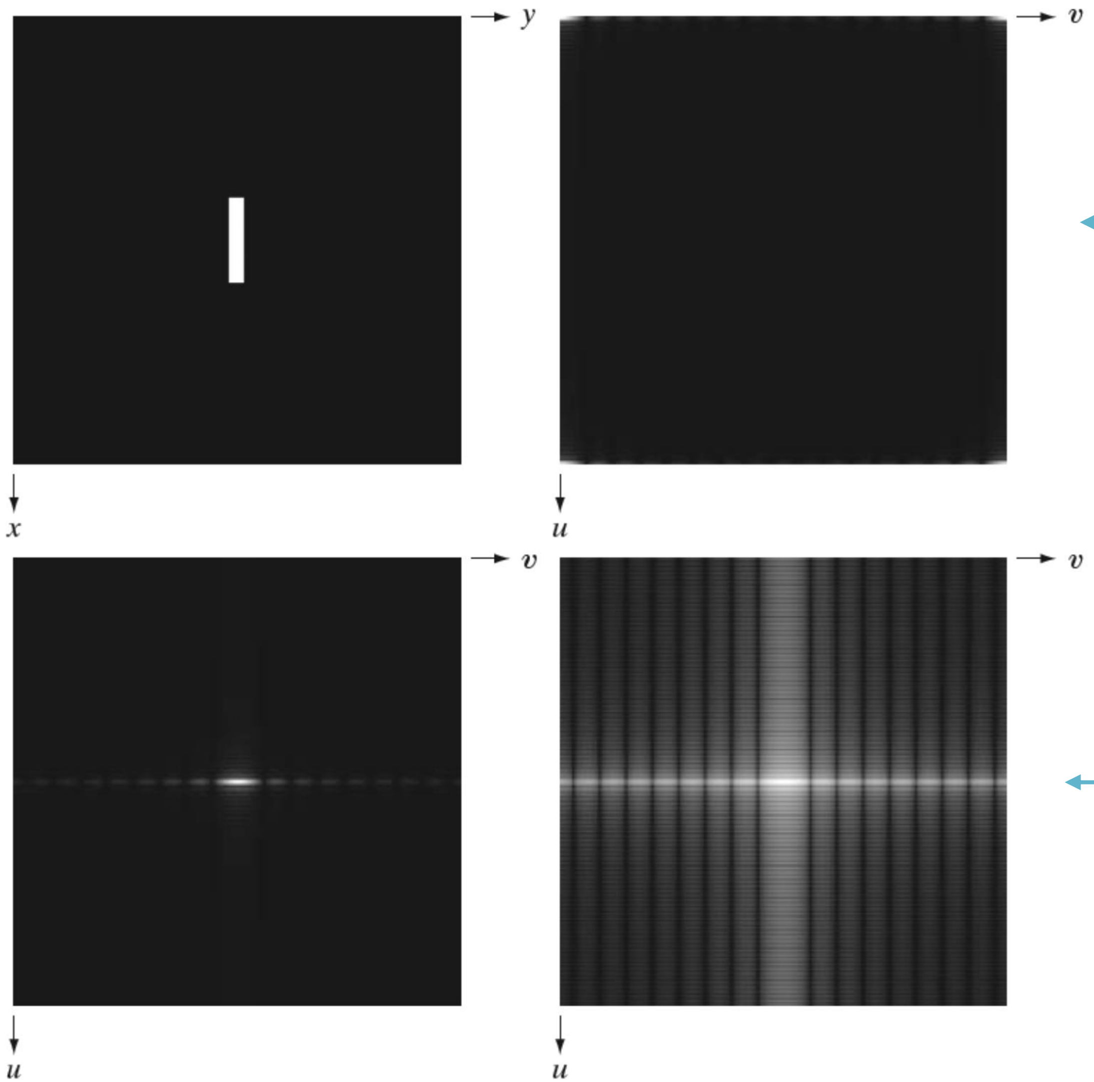
$|F(u,v)|$

[0 a $2{,}5\text{x}10^{6}$]

[0 a 6,4]



Visualização do espectro

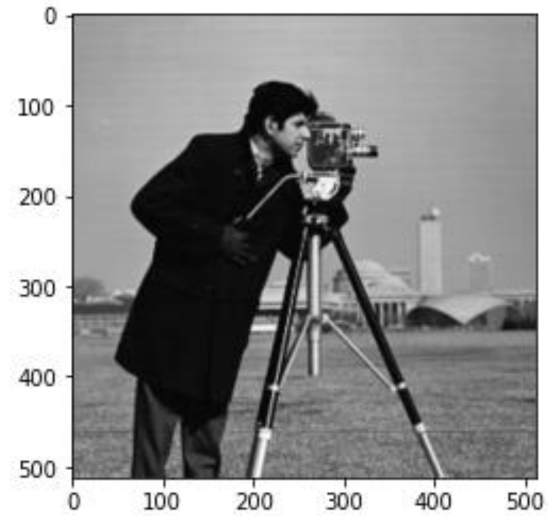


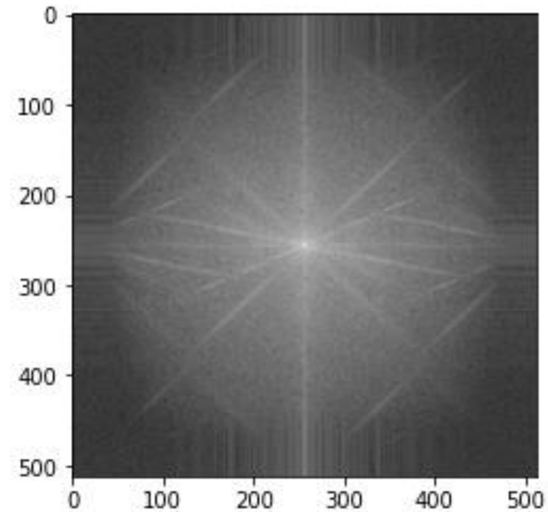
Deslocamento do espectro

Transformação logaritma

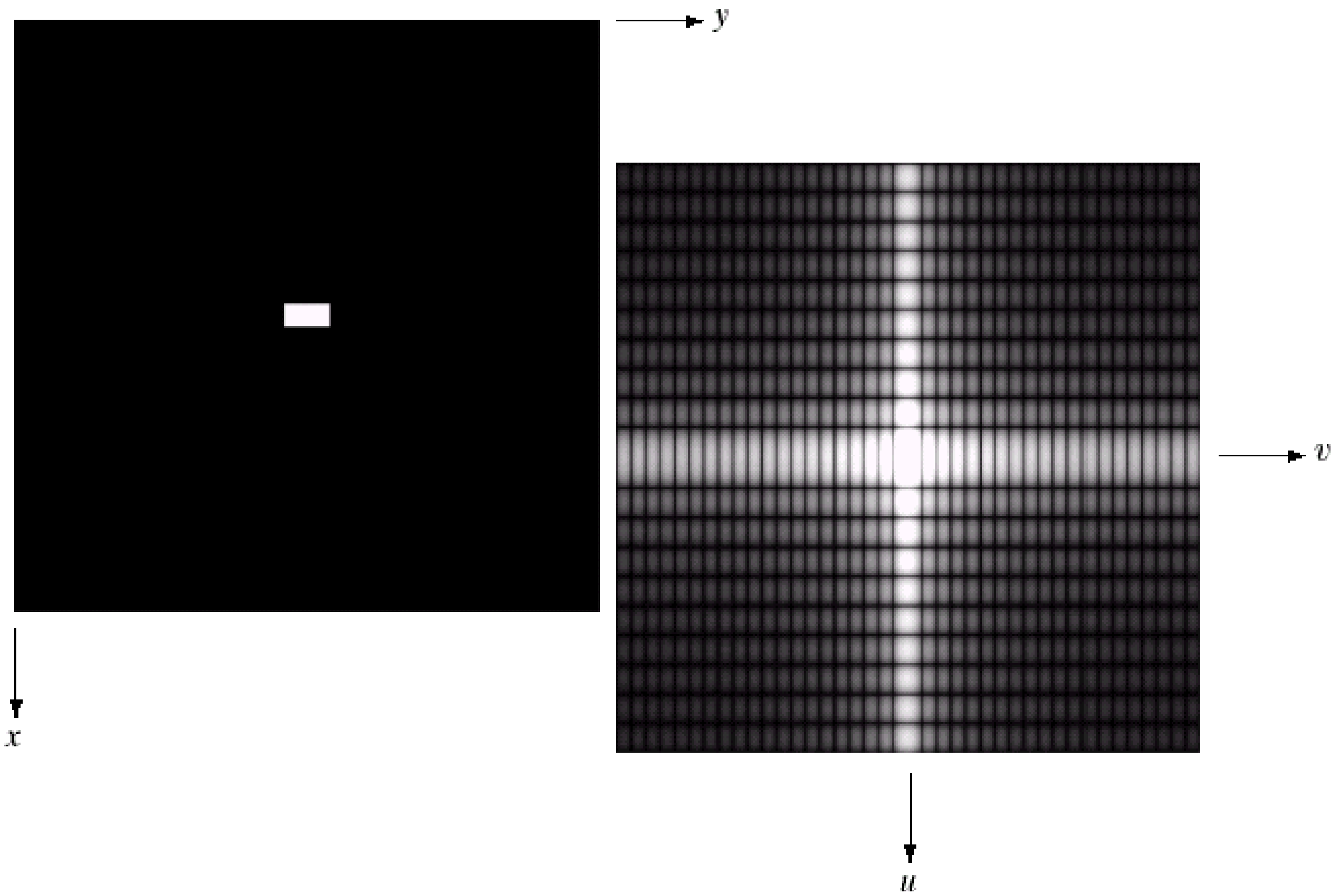
Em Python

```
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

#---importando imagem---
img = cv.imread('cameraman.tif',cv.IMREAD_UNCHANGED)
plt.figure()
plt.imshow(img,'gray')
plt.show()

#---FFT---
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)

#---visualização de espectro---
magnitude_spectrum = 20*np.log(np.abs(fshift)+1)
plt.figure()
plt.imshow(magnitude_spectrum,'gray')
plt.show()
```



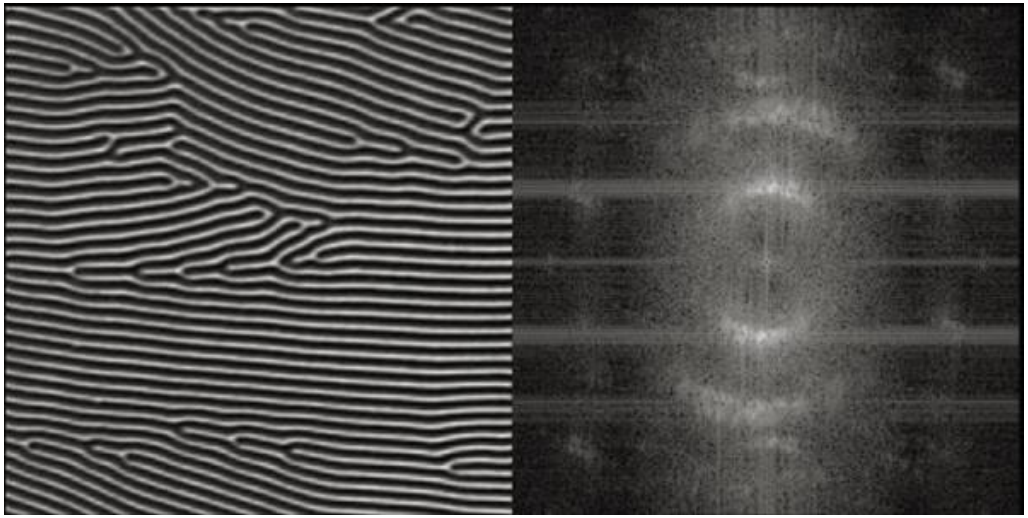


Espectro de uma imagem

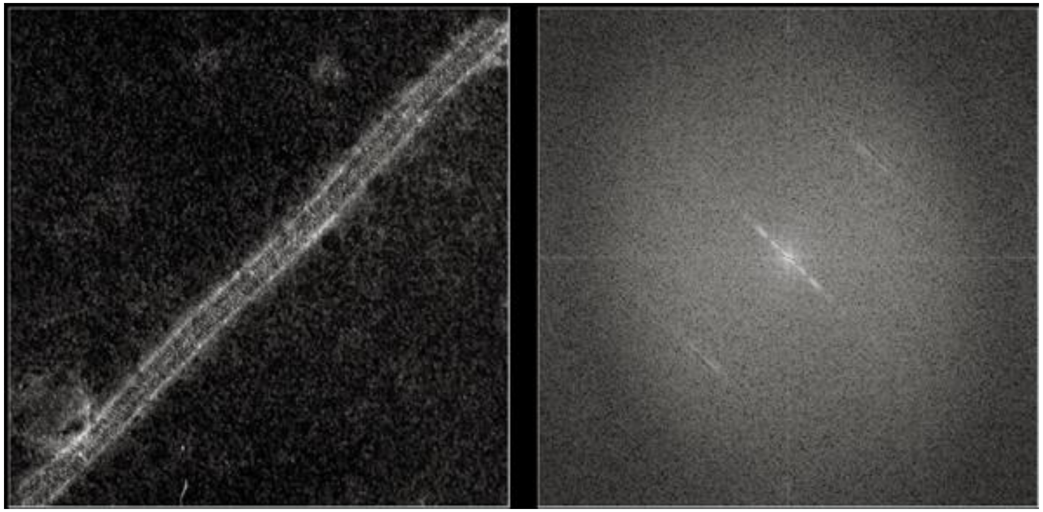


Exemplos:

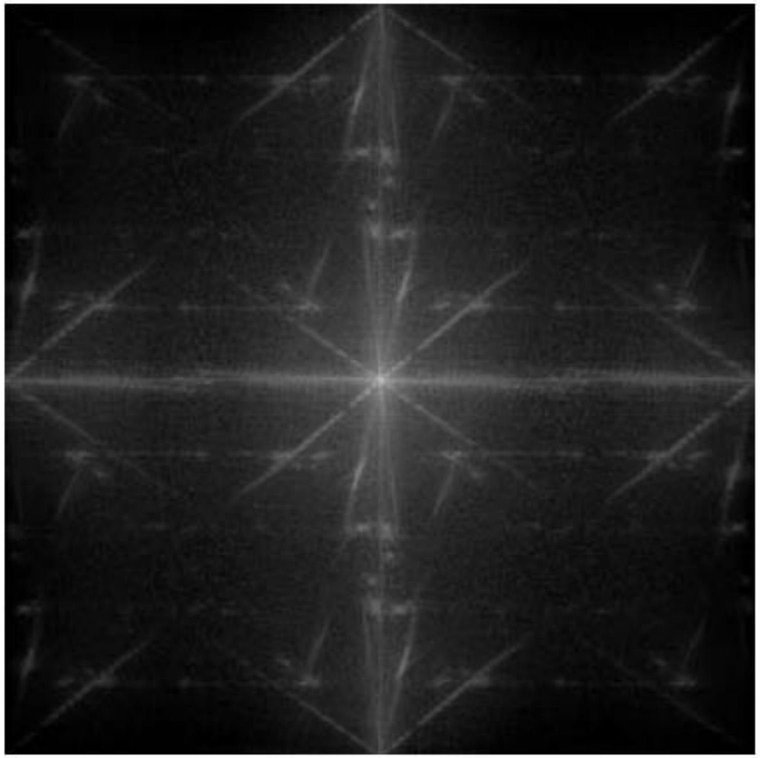
a)



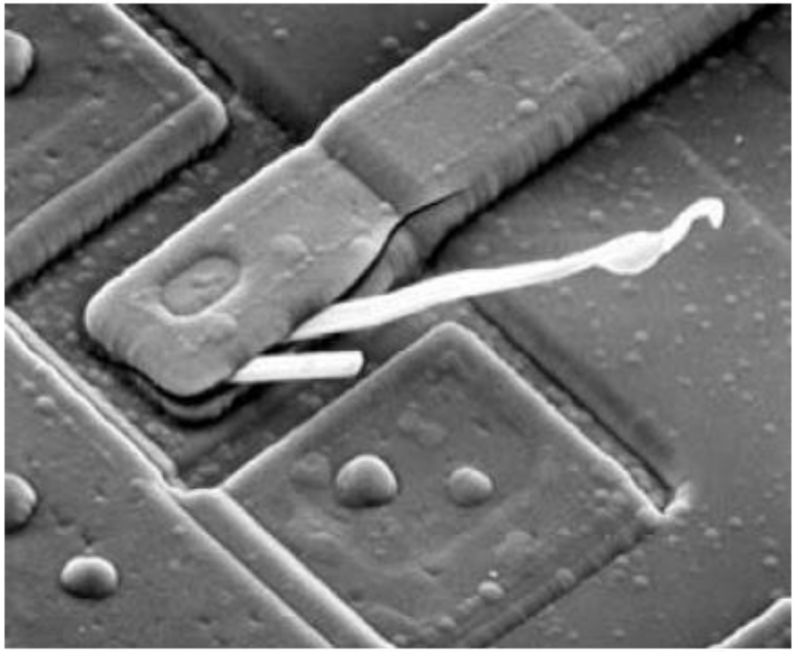
b)

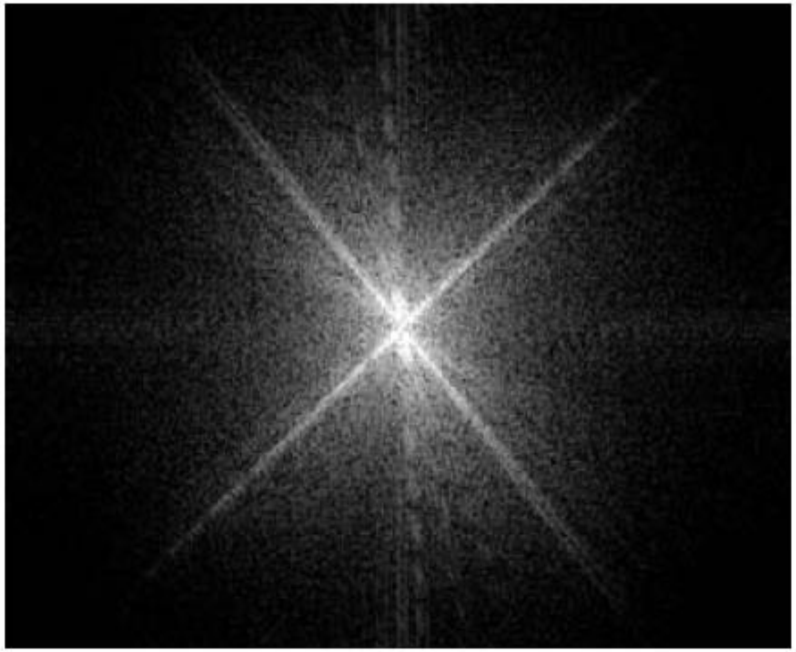


Exemplos:



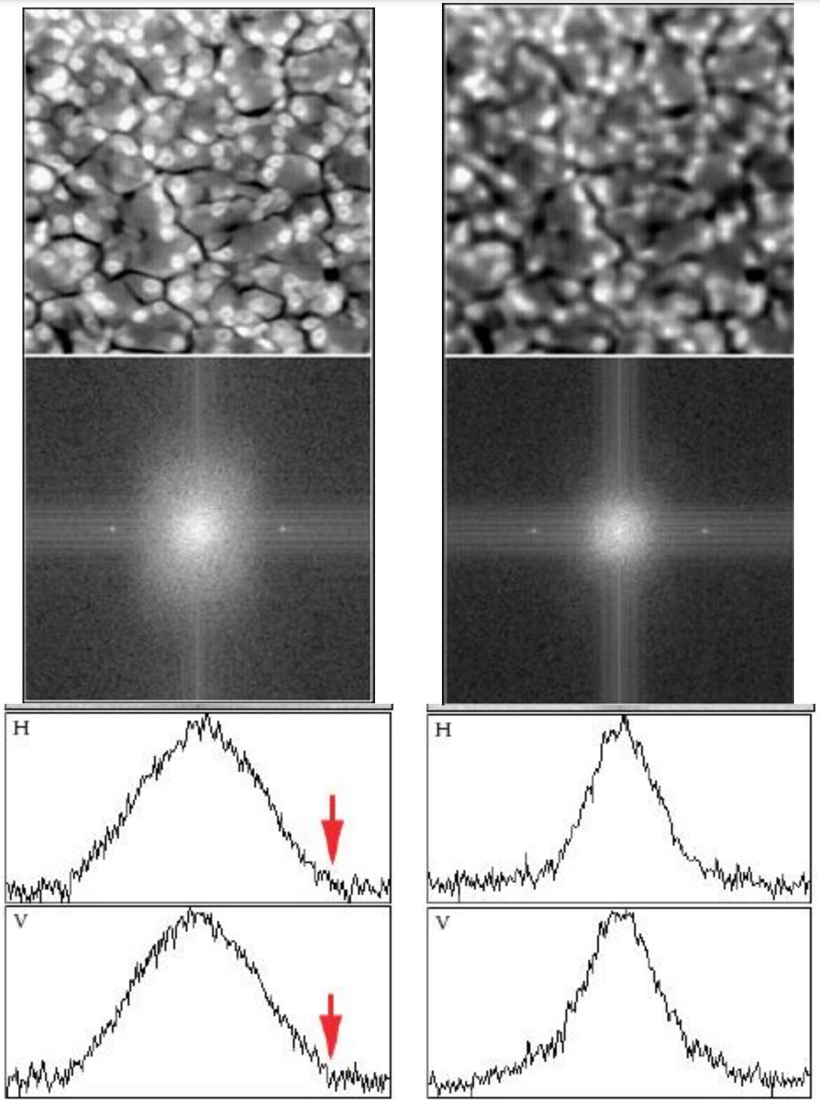
Exemplos:





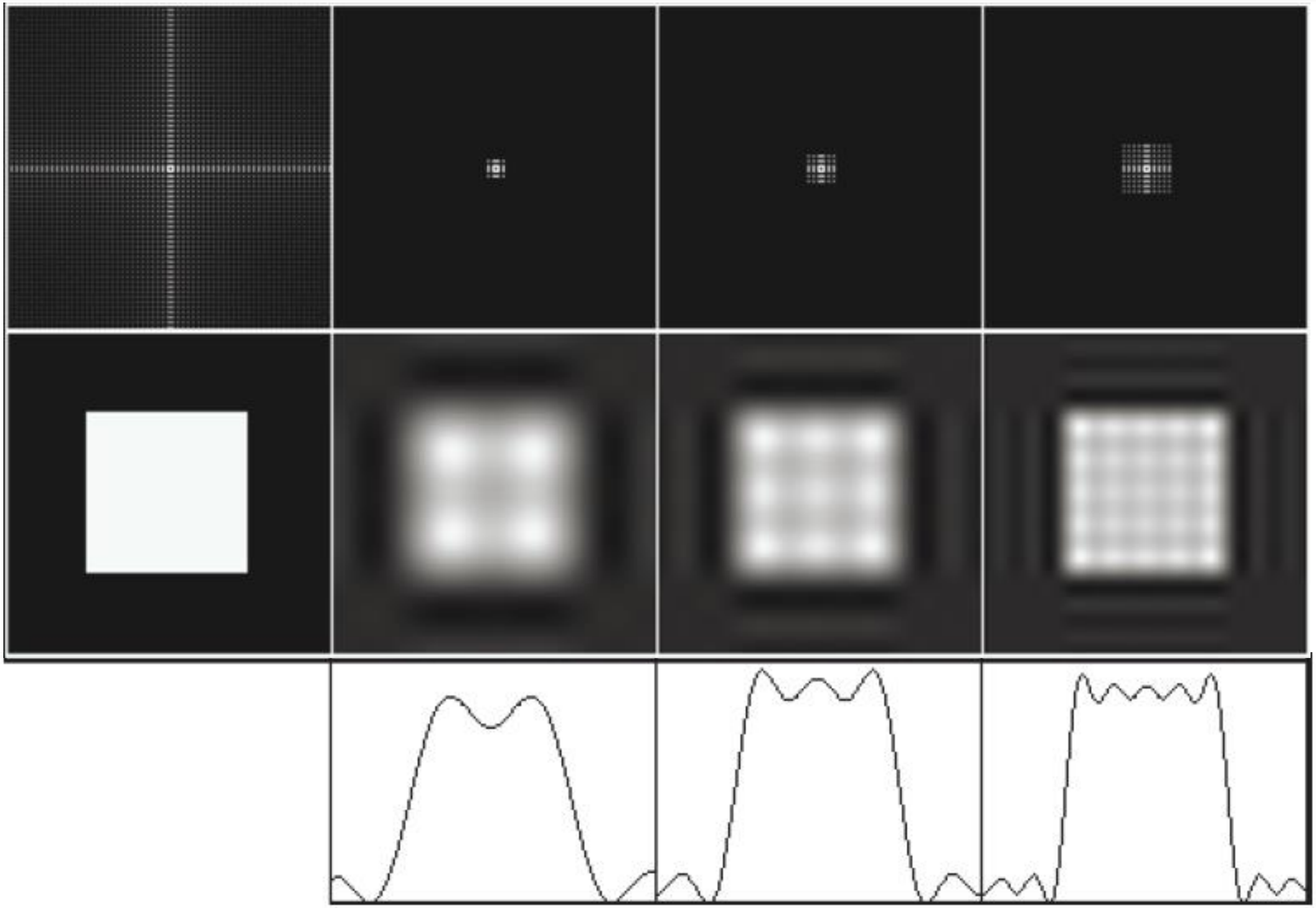
Exemplos:

Alta resolução espacial: presença de componentes de alta frequência



Baixa resolução espacial: perda de componentes de alta frequência

Espectro Unidimensional e Bidimensional



Propriedades da DFT 2-D

3) Separabilidade

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\exp[-j2\pi(\frac{ux+vy}{N})]$$

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\exp[-j2\pi ux/N]\sum_{y=0}^{N-1} f(x,y)\exp[-j2\pi vy/N]$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\exp[j2\pi(\frac{ux+vy}{N})]$$

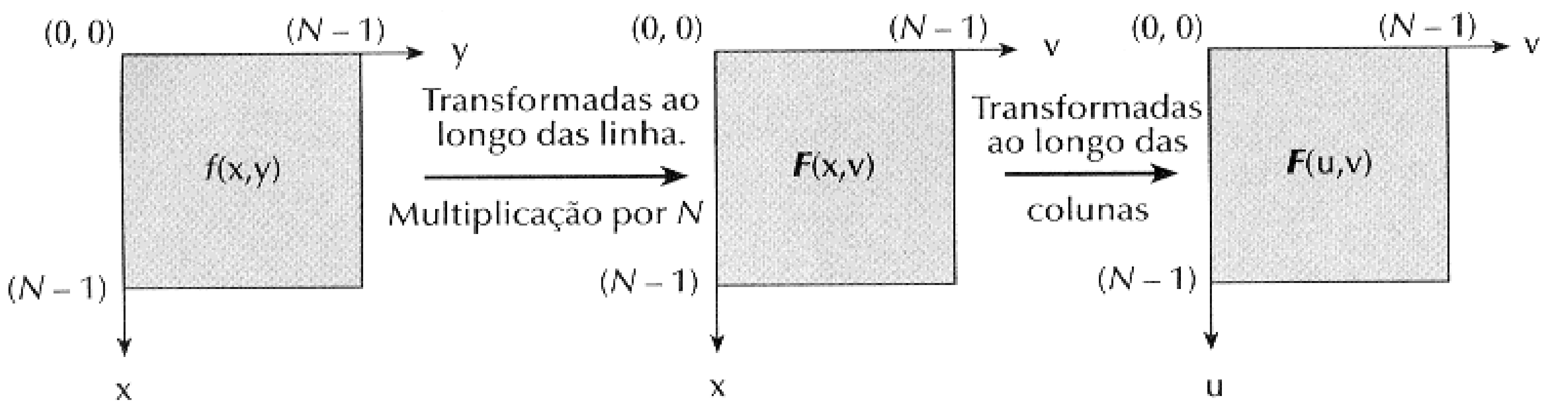
$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\exp[j2\pi ux/N]\sum_{v=0}^{N-1} F(u,v)\exp[j2\pi vy/N]$$

Propriedades da DFT 2-D

A vantagem da **Separabilidade** é que a F(u,v) e a f(x,y) podem ser obtidas em dois passos por aplicações sucessivas da transformada de Fourier unidimensional:

$$\boxed{F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v)\exp[-j2\pi ux/N]}$$

$$\boxed{F(x,v) = N\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp[-j2\pi vy/N]\right]}$$



Propriedades da DFT 2-D

4) Rotação

Introduzindo as coordenadas polares:

$x = r\cos\theta \qquad y = r\,\mathrm{sen}\,\theta \qquad u = w\cos\phi \qquad v = w\,\mathrm{sen}\,\phi$

$f(x,y) \;\; e \;\; F(u,v)$ tornam-se : $f(r,\theta) \;\; e \;\; F(w,\phi)$

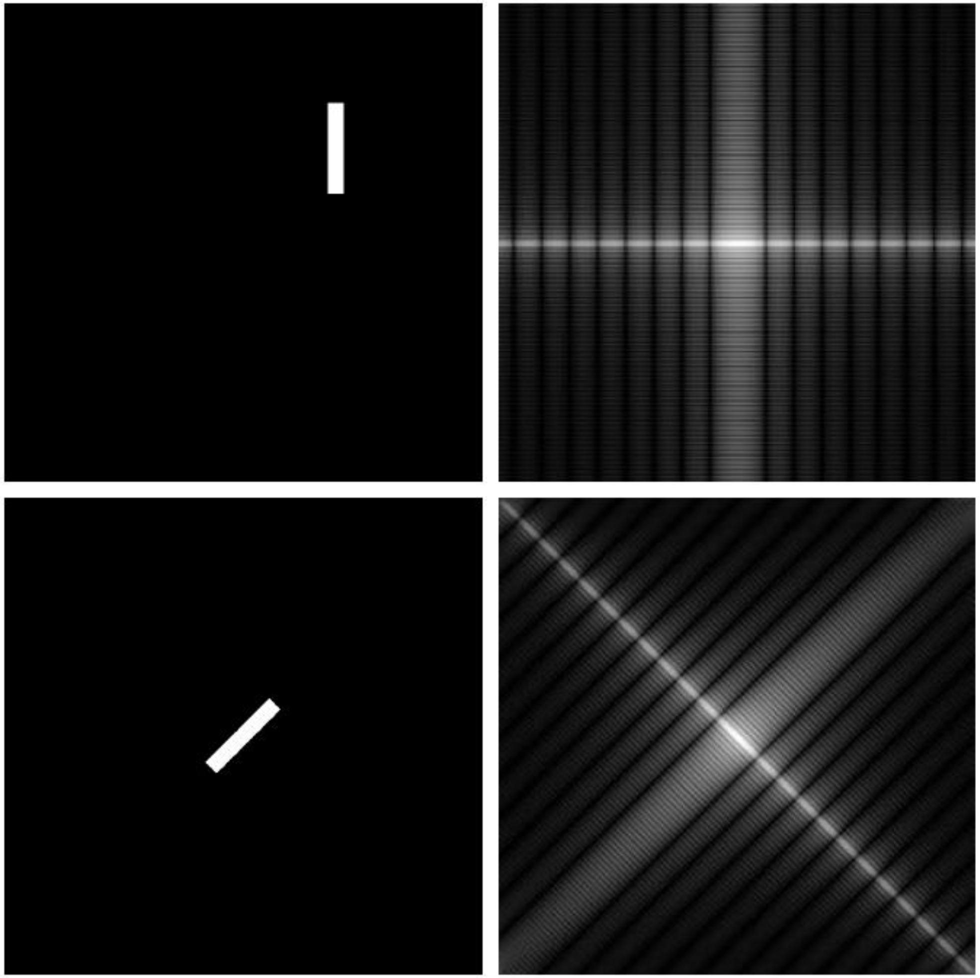
A substituição direta no par de Transformadas resulta:

$$\boxed{f(r,\theta+\theta_0) \Leftrightarrow F(w,\phi+\theta_0)}$$

Ou seja, a rotação de f(x,y) de um ângulo θ_0 implicará em uma rotação de F(u,v) deste mesmo ângulo.

Propriedades da DFT 2-D

4) Rotação



Propriedades da DFT 2-D

5) Distributividade

A Transformada de Fourier e sua Inversa são Distributivas com relação à Adição.

$$\boxed{\Im\left\{f_1(x,y)+f_2(x,y)\right\}=\Im\left\{f_1(x,y)\right\}+\Im\left\{f_2(x,y)\right\}}$$

A Transformada de Fourier e sua Inversa Não são Distributivas com relação à Multiplicação.

$$\boxed{\Im\left\{f_1(x,y).f_2(x,y)\right\}\neq\Im\left\{f_1(x,y)\right\}.\Im\left\{f_2(x,y)\right\}}$$

Propriedades da DFT 2-D

6) Mudança de Escala:

Para dois escalares a e b:

$$\boxed{af(x,y) \Leftrightarrow aF(u,v)}$$

7) Valor Médio:

Substituindo-se $u = v = 0$ na equação da transformada 2-D:

$$F(0,0) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

Logo, o valor médio de f(x,y) é:

$$\boxed{\overline{f}(x,y) = \frac{1}{N}F(0,0)}$$

Em Python

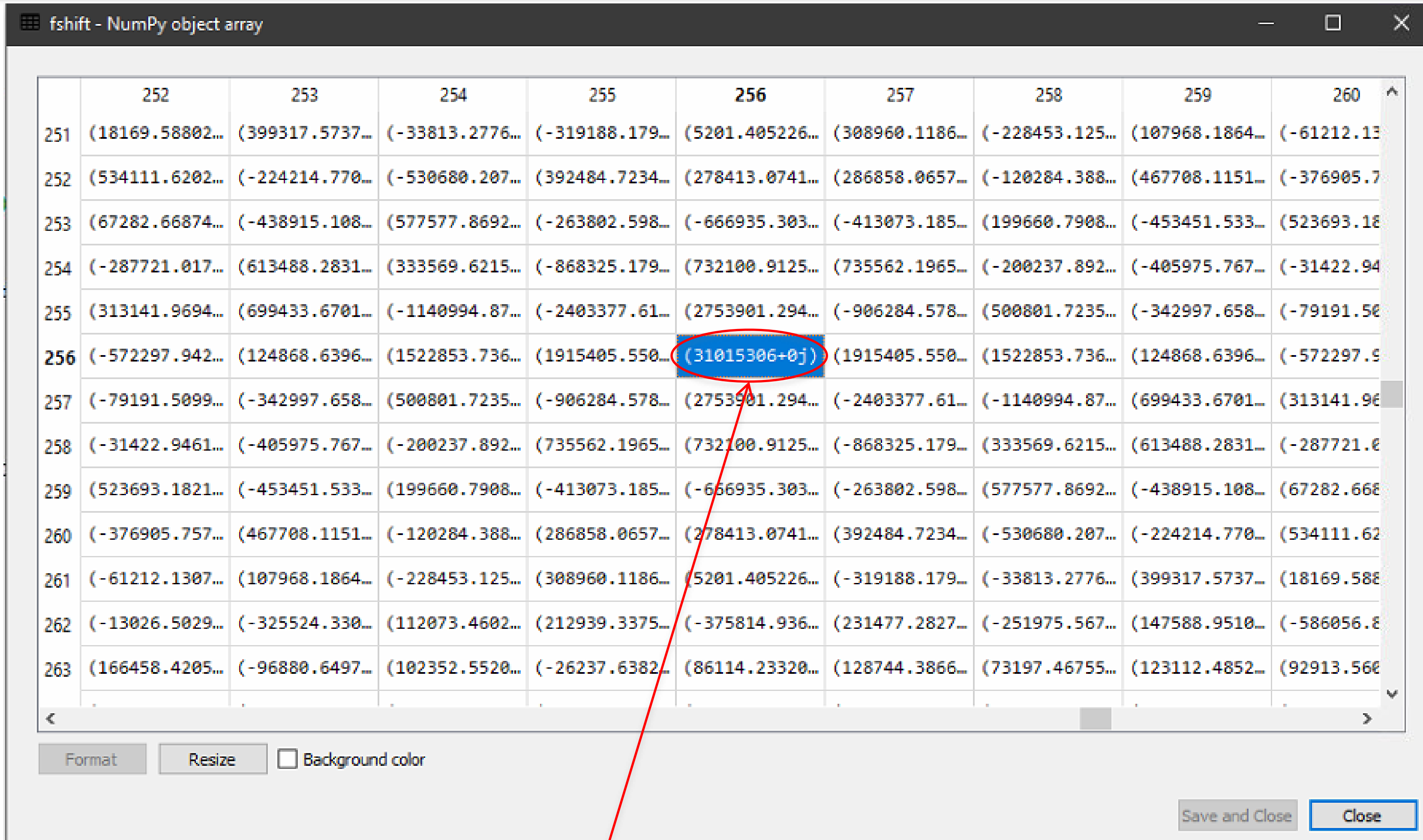
```python
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

#---importando imagem---
img = cv.imread('cameraman.tif', cv.IMREAD_UNCHANGED)
plt.figure()
plt.imshow(img,'gray')
plt.show()
M,N = img.shape
print("Tamanho da imagem: ({},{})".format(M,N))

soma = np.sum(img)
media = soma/(M*N)
print("Soma:{} / Média:{}".format(soma,media))

#---FFT---
f = np.fft.fft2(img)
#---Valor DC---
valorDC = f[0,0]
print("ValorDC:{}".format(valorDC))

#---shift no FFT---
fshift = np.fft.fftshift(f)
cx = int(M/2)
cy = int(N/2)
valorDC = fshift[cx,cy]
print("ValorDC:{}".format(valorDC))
```

```
Tamanho da imagem: (512,512)
Soma:31015306 / Média:118.31400299072266
ValorDC f[0,0]:(31015306+0j)
ValorDC fshift[256,256]:(31015306+0j)
```

512*512*118,314

= 31015306

Em Python



	252	253	254	255	256	257	258	259	260
251	(18169.58802…	(399317.5737…	(-33813.2776…	(-319188.179…	(5201.405226…	(308960.1186…	(-228453.125…	(107968.1864…	(-61212.13
252	(534111.6202…	(-224214.770…	(-530680.207…	(392484.7234…	(278413.0741…	(286858.0657…	(-120284.388…	(467708.1151…	(-376905.7
253	(67282.66874…	(-438915.108…	(577577.8692…	(-263802.598…	(-666935.303…	(-413073.185…	(199660.7908…	(-453451.533…	(523693.18
254	(-287721.017…	(613488.2831…	(333569.6215…	(-868325.179…	(732100.9125…	(735562.1965…	(-200237.892…	(-405975.767…	(-31422.94
255	(313141.9694…	(699433.6701…	(-1140994.87…	(-2403377.61…	(2753901.294…	(-906284.578…	(500801.7235…	(-342997.658…	(-79191.5
256	(-572297.942…	(124868.6396…	(1522853.736…	(1915405.550…	(31015306+0j)	(1915405.550…	(1522853.736…	(124868.6396…	(-572297.9
257	(-79191.5099…	(-342997.658…	(500801.7235…	(-906284.578…	(2753901.294…	(-2403377.61…	(-1140994.87…	(699433.6701…	(313141.96
258	(-31422.9461…	(-405975.767…	(-200237.892…	(735562.1965…	(732100.9125…	(-868325.179…	(333569.6215…	(613488.2831…	(-287721.0
259	(523693.1821…	(-453451.533…	(199660.7908…	(-413073.185…	(-666935.303…	(-263802.598…	(577577.8692…	(-438915.108…	(67282.66
260	(-376905.757…	(467708.1151…	(-120284.388…	(286858.0657…	(278413.0741…	(392484.7234…	(-530680.207…	(-224214.770…	(534111.62
261	(-61212.1307…	(107968.1864…	(-228453.125…	(308960.1186…	(5201.405226…	(-319188.179…	(-33813.2776…	(399317.5737…	(18169.58
262	(-13026.5029…	(-325524.330…	(112073.4602…	(212939.3375…	(-375814.936…	(231477.2827…	(-251975.567…	(147588.9510…	(-586056.8
263	(166458.4205…	(-96880.6497…	(102352.5520…	(-26237.6382…	(86114.23320…	(128744.3866…	(73197.46755…	(123112.4852…	(92913.56

Soma de todos os valores de pixel da imagem = Valor Médio * (M x N)

Em Python

fshift - NumPy object array

	254	255	256	257	258	
250	(-251975.5671…	(231477.28271278966+10910…	(-375814.9366890566+507332.67690…	(212939.33759380912-9057…	(112073.4602…	(-325
251	(-33813.27762…	(-319188.17972859193-1238…	(5201.405226383324+5163.39593314…	(308960.11869567644+2624…	(-228453.125…	(1079
252	(-530680.2072…	(392484.7234019517+116902…	(278413.07412932883+1272594.3172…	(286858.0657433753-27737…	(-120284.388…	(4677
253	(577577.86922…	(-263802.59822857723-2283…	(-666935.3036280419+356164.80619…	(-413073.18566989526-231…	(199660.7908…	(-453
254	(333569.62151…	(-868325.1791232547-91946…	(732100.9125463488+2002580.68234…	(735562.1965035057-38927…	(-200237.892…	(-405
255	(-1140994.872…	(-2403377.6149647254+3046…	(2753901.2941309055+3380432.5879…	(-906284.578532978-87043…	(500801.7235…	(-342
256	(1522853.7360…	(1915405.55060915-4666241…	(31015306+0j)	(1915405.5506091495+4666…	(1522853.736…	(1248
257	(500801.72351…	(-906284.5785329778+87043…	(2753901.294130905-3380432.58797…	(-2403377.6149647254-304…	(-1140994.87…	(6994
258	(-200237.8926…	(735562.1965035056+389277…	(732100.9125463485-2002580.68234…	(-868325.1791232547+9194…	(333569.6215…	(6134
259	(199660.79080…	(-413073.1856698954+23118…	(-666935.3036280416-356164.80619…	(-263802.59822857723+228…	(577577.8692…	(-438
260	(-120284.3887…	(286858.06574337534+27737…	(278413.07412932883-1272594.3172…	(392484.7234019517-11690…	(-530680.207…	(-224
261	(-228453.1258…	(308960.11869567644-26245…	(5201.405226358342-5163.395933144…	(-319188.17972859193+123…	(-33813.2776…	(3993
262	(112073.46029…	(212939.33759380906+90570…	(-375814.9366890566-507332.67690…	(231477.28271278966-1091…	(-251975.567…	(1475

Format Resize Background color

Complexos Conjugados

Save and Close Close

Propriedades da DFT 2-D

8) Convolução

Teorema da Convolução.

$$f(x,y)*g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

$$f(x,y)g(x,y) \Leftrightarrow F(u,v)*G(u,v)$$

Convolução *no domínio do tempo/espaço*

Multiplicação *no domínio da frequência*

Multiplicação *no domínio do tempo/espaço*

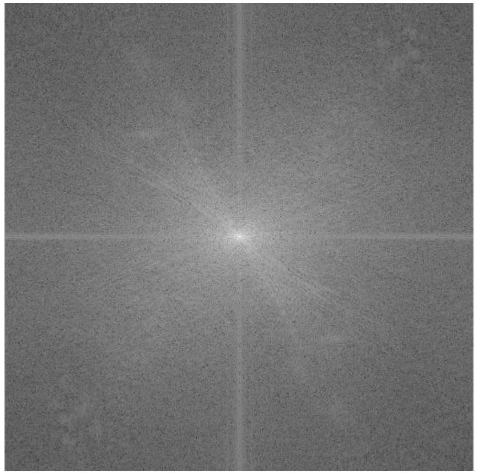
Convolução *no domínio da frequência*

Magnitude x Fase

O que é mais importante?

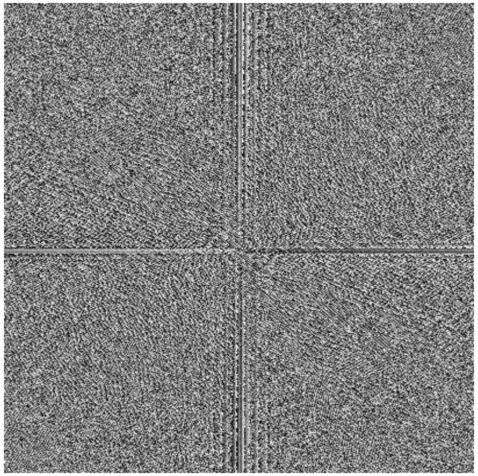
MÓDULO:

Amplitude de cada onda 2D



FASE:

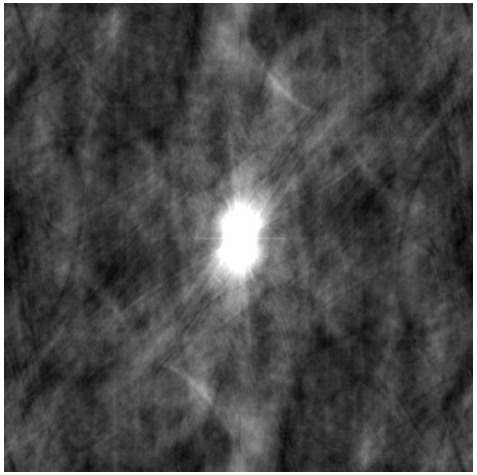
Direção de cada onda 2D



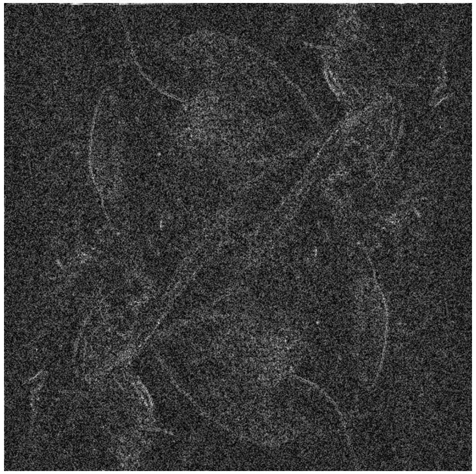
Magnitude x Fase

O que é mais importante?

Transformação inversa usando apenas o MÓDULO



Transformação inversa usando apenas a FASE



Parte Real x Parte Imaginária

Transformação inversa usando apenas a parte REAL

Transformação inversa usando apenas a parte IMAGINÁRIA

FIM