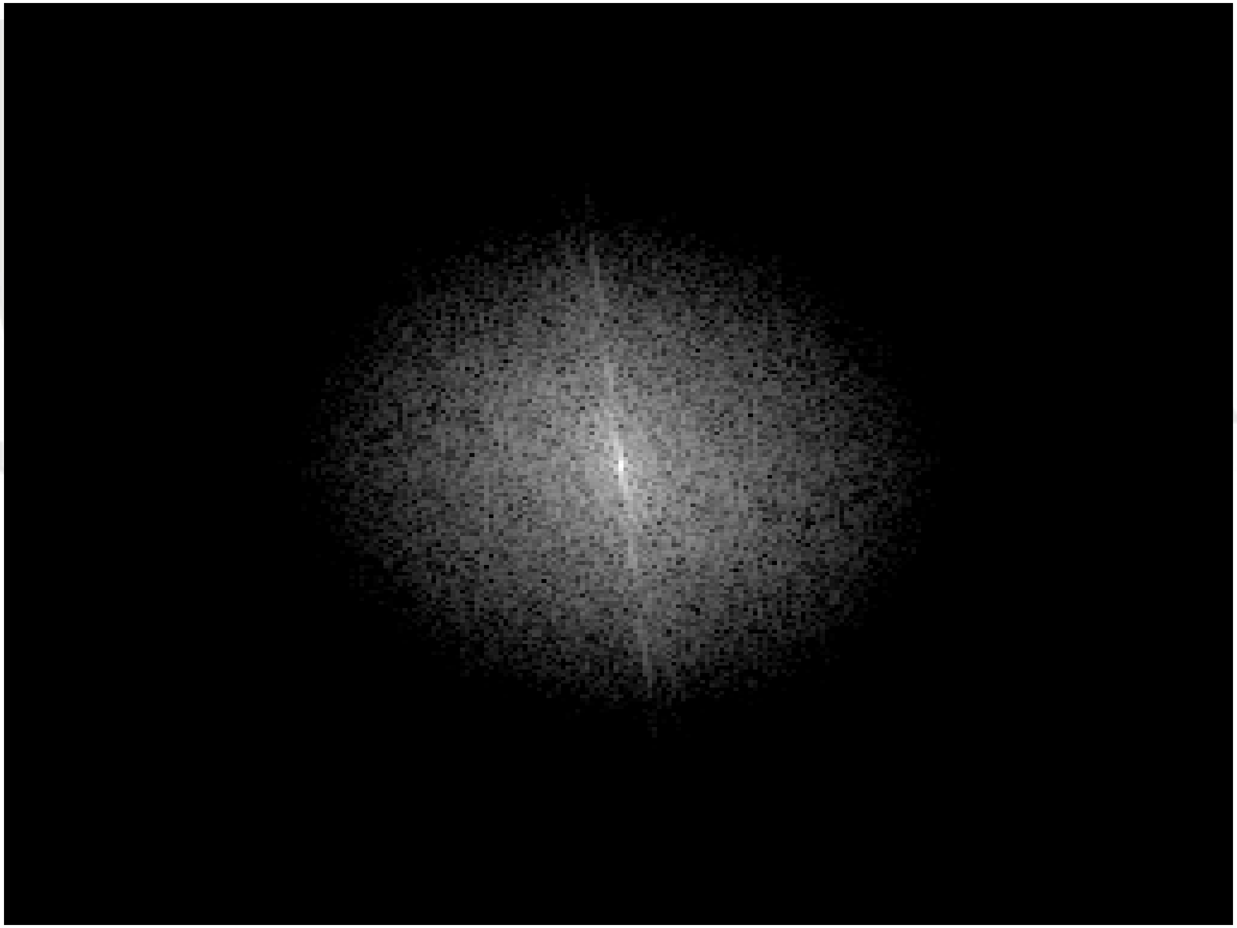
Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo de aplicação: auto-foco

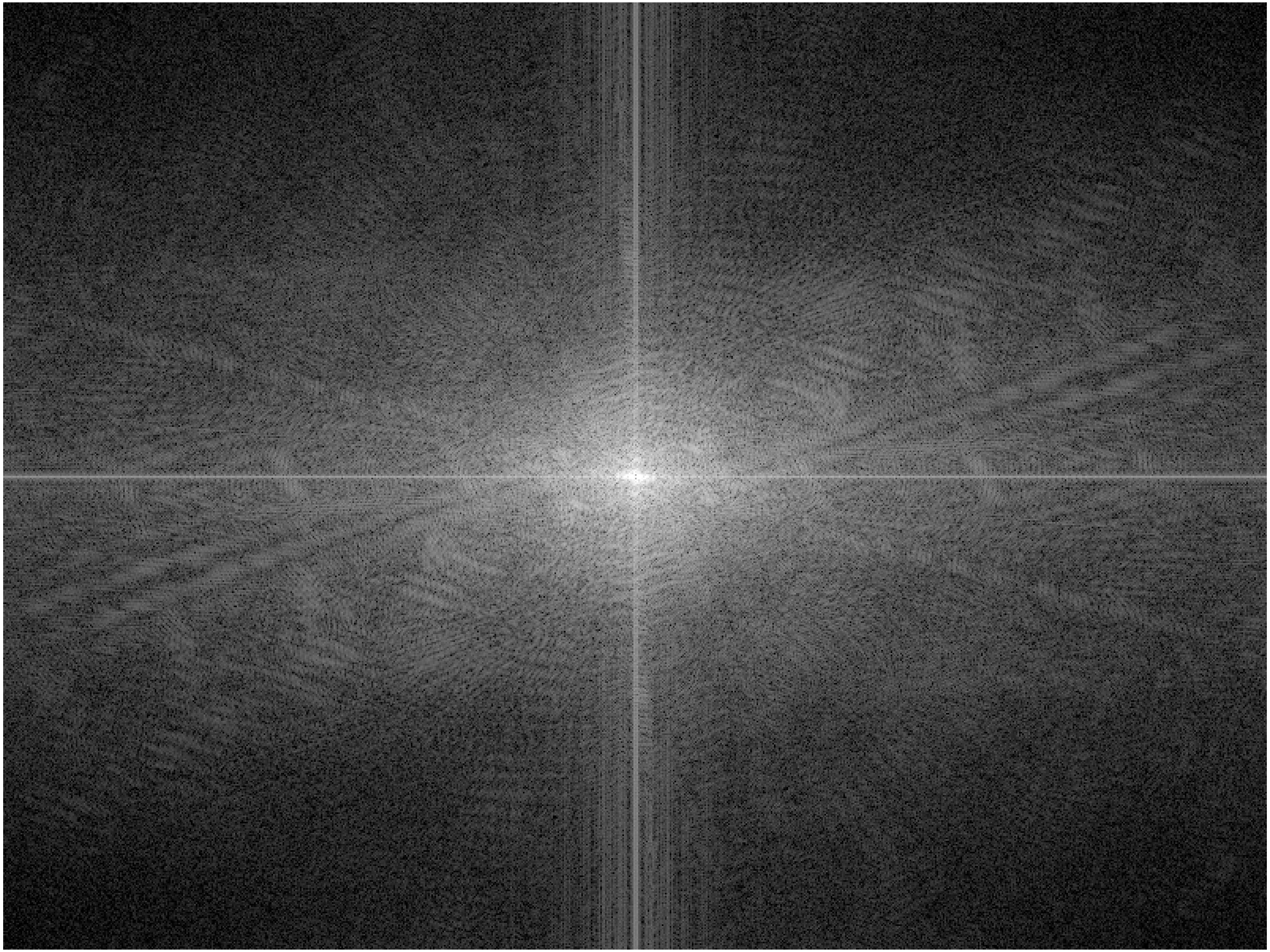
- Como poderíamos criar um algoritmo simples de auto-foco com base na análise da DFT?

Exemplo de aplicação: auto-foco

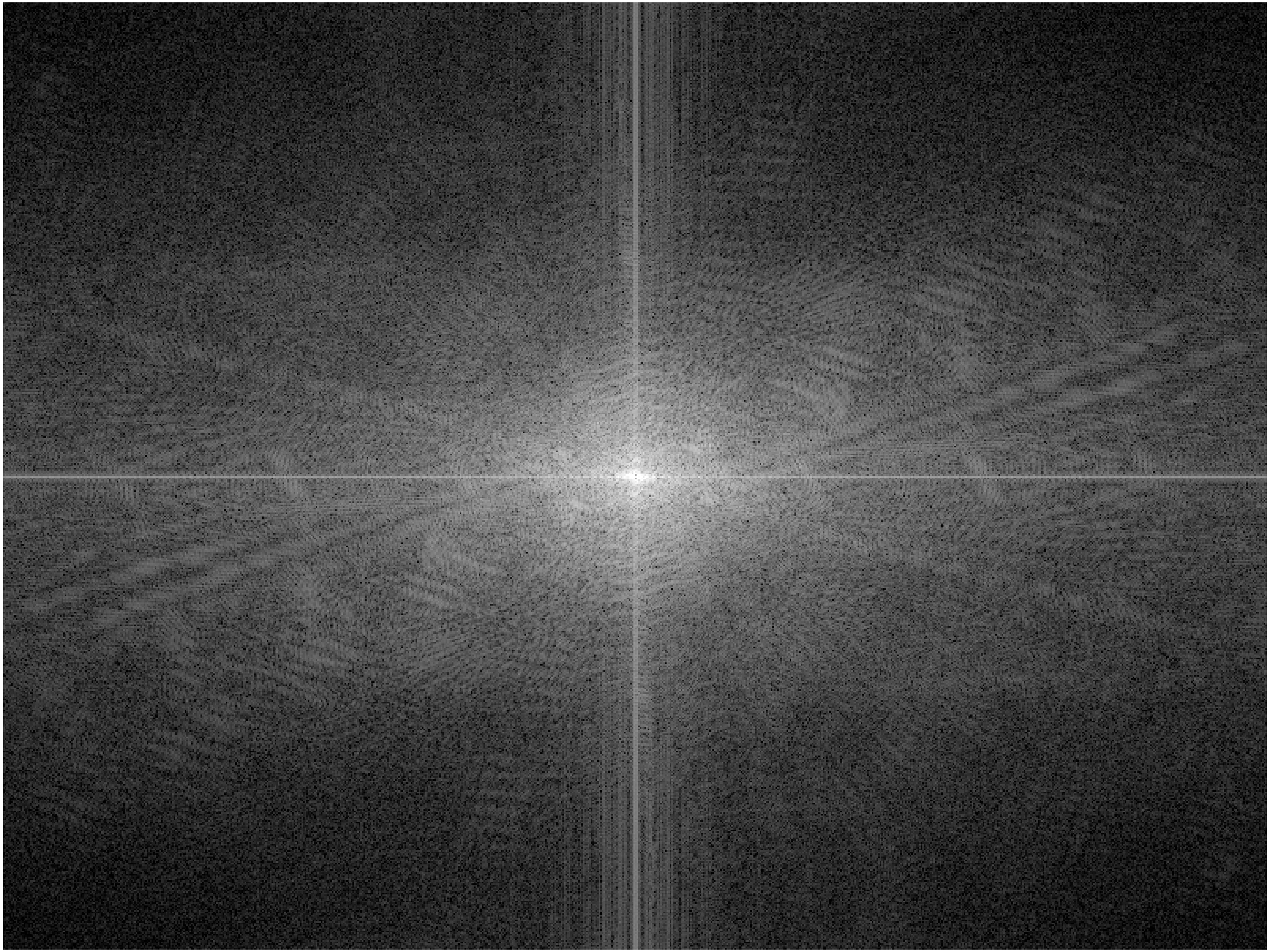
- Como poderíamos criar um algoritmo simples de auto-foco com base na análise da DFT?

- Conceito:
  - Varia o ajuste de foco.
  - Para cada ajuste:
    - Calcula a DFT.
    - Obtém as magnitudes.
    - Faz uma soma ponderada das magnitudes, com peso diretamente proporcional à distância até o centro.
      - Mais longe = frequências mais altas = peso maior.
  - O melhor ajuste é aquele que produz a maior soma.

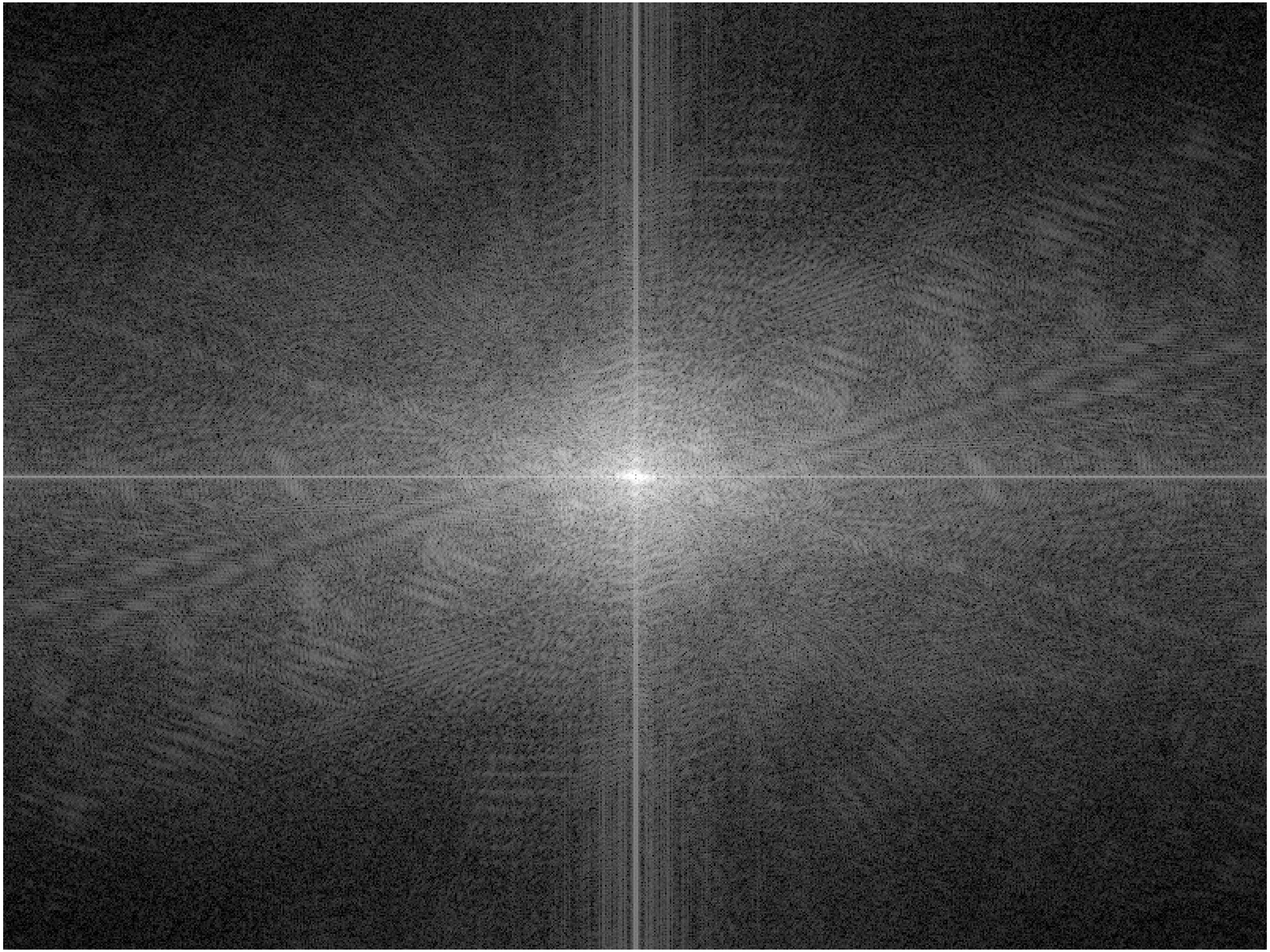
1



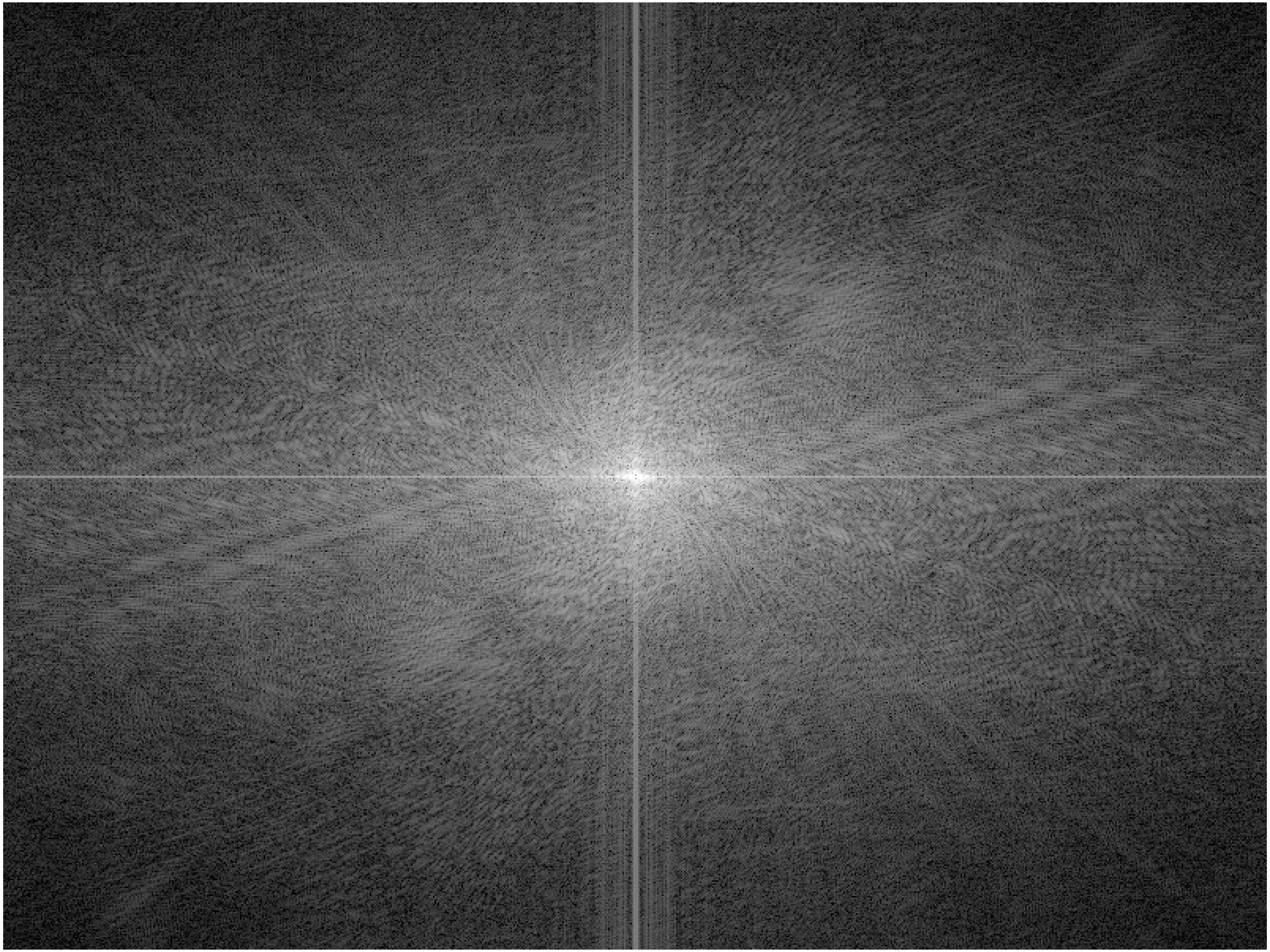
2



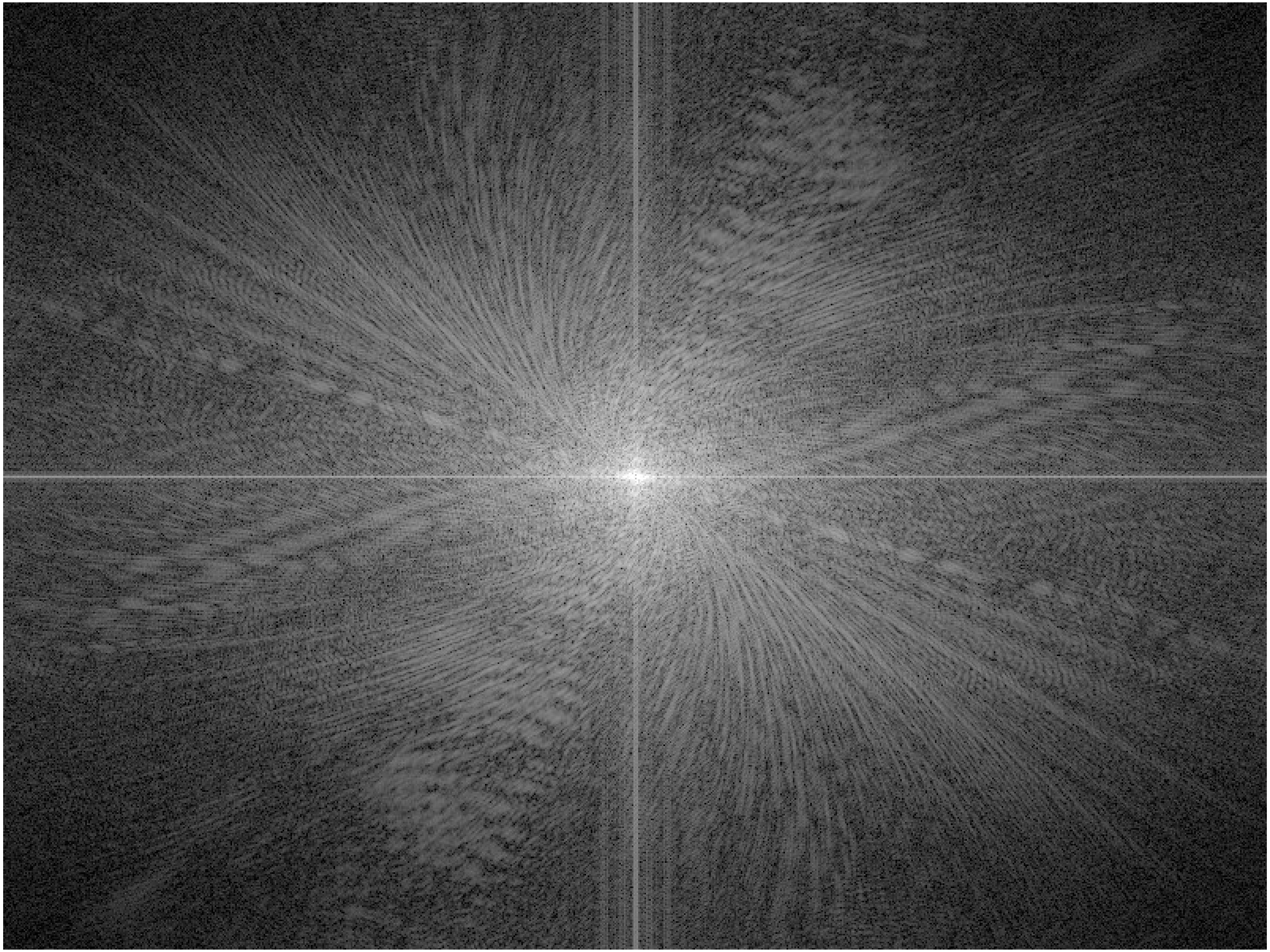
3



4

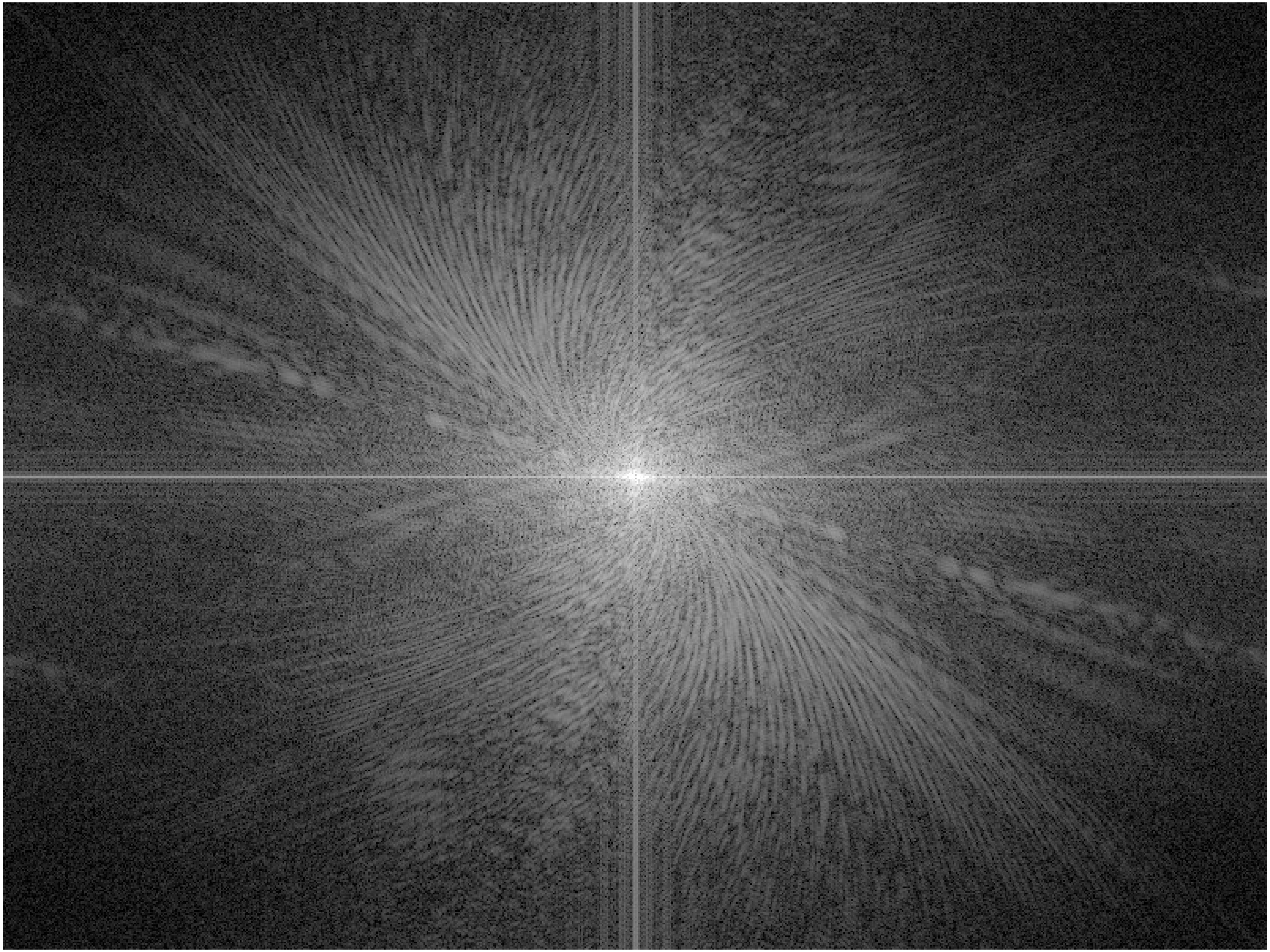


5

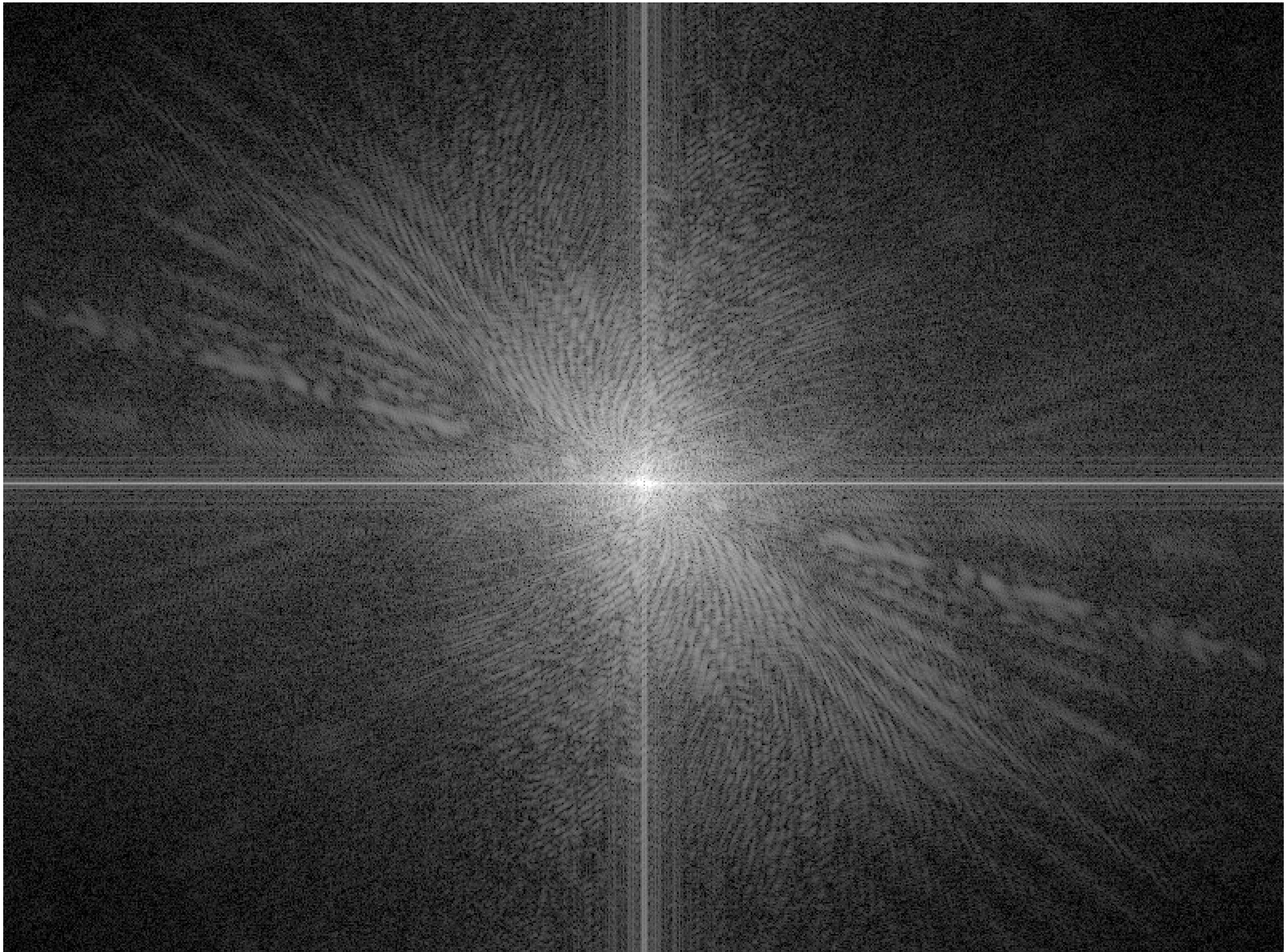


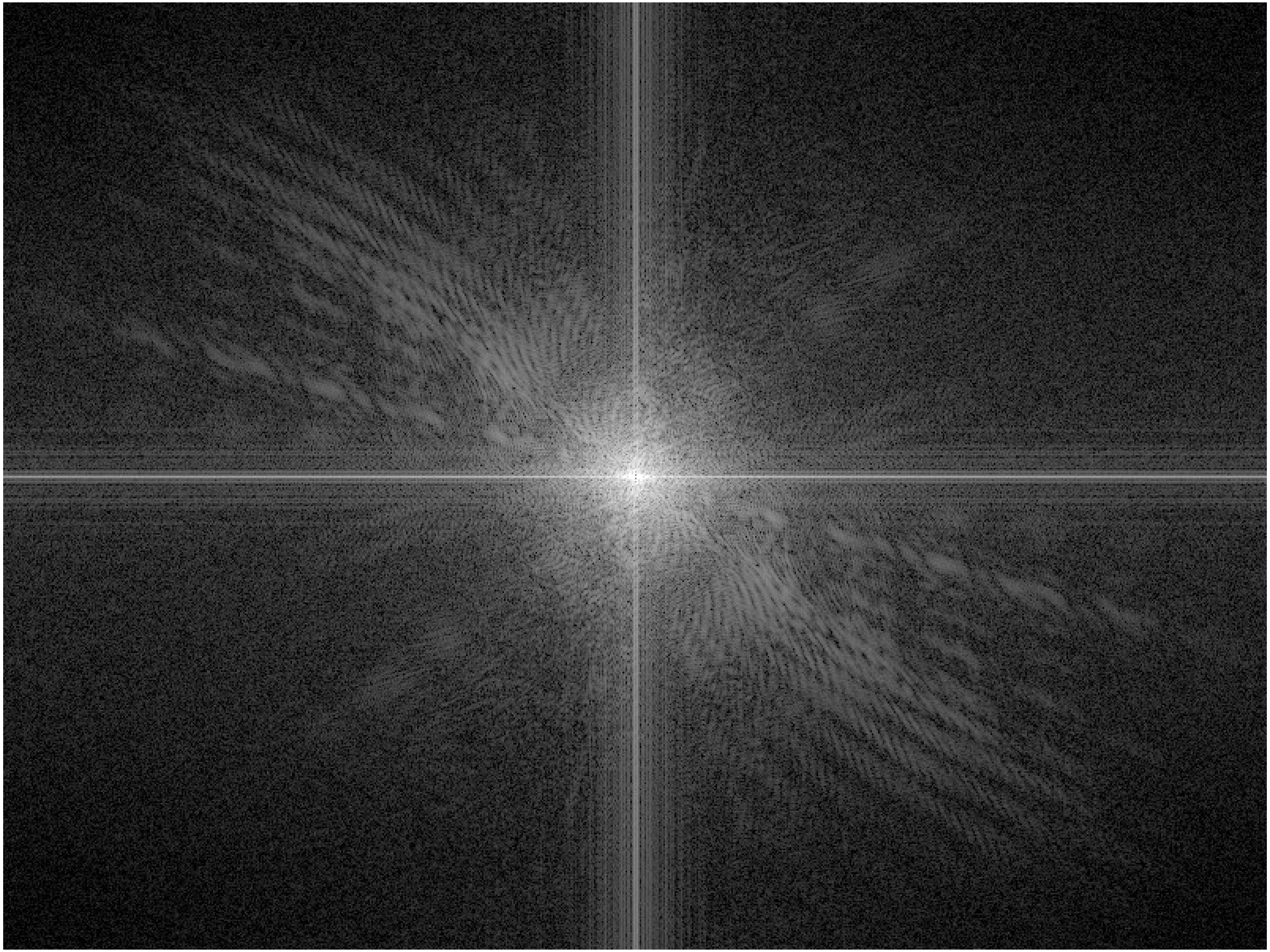
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

6

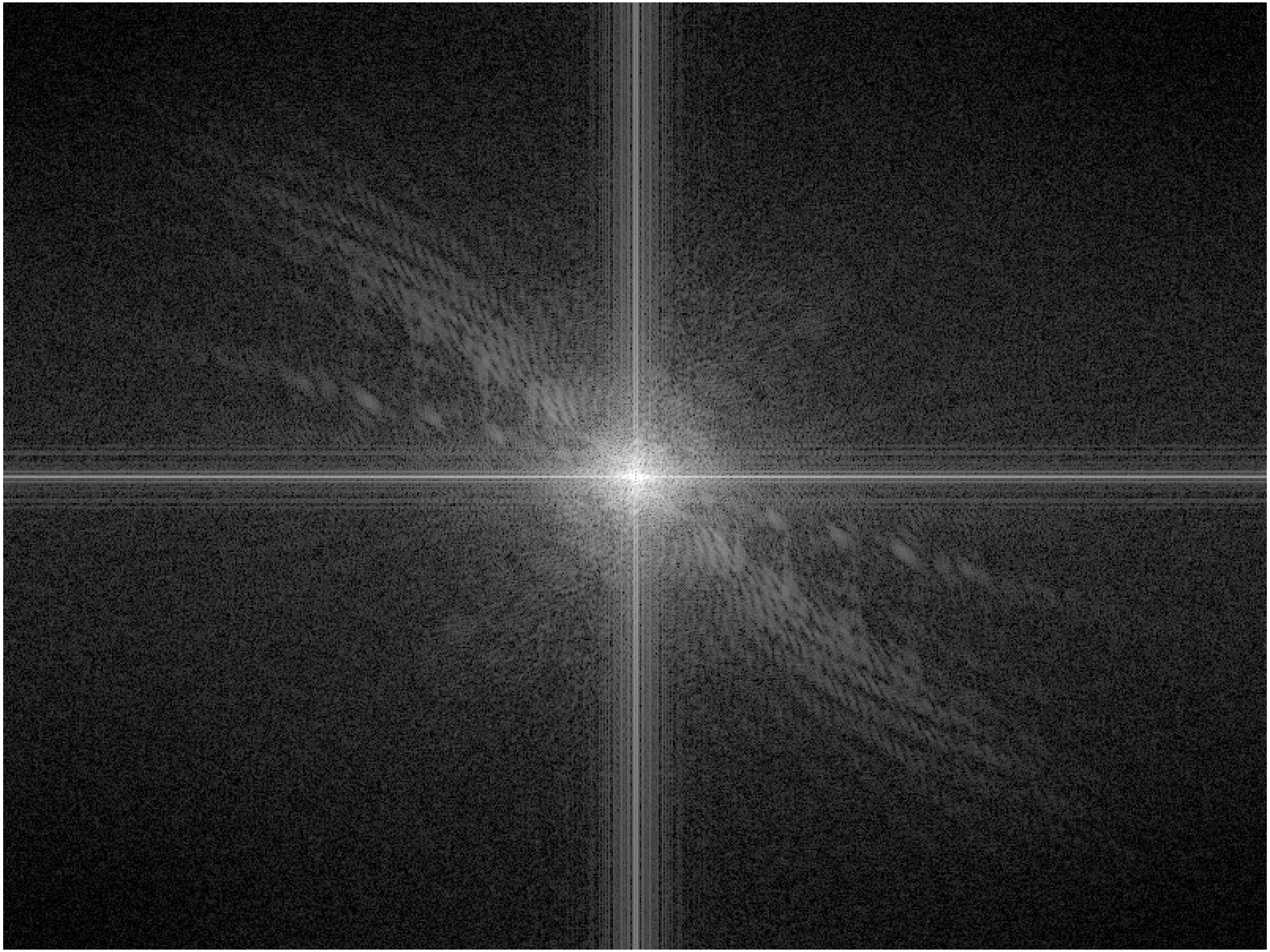


7





9



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

1

2

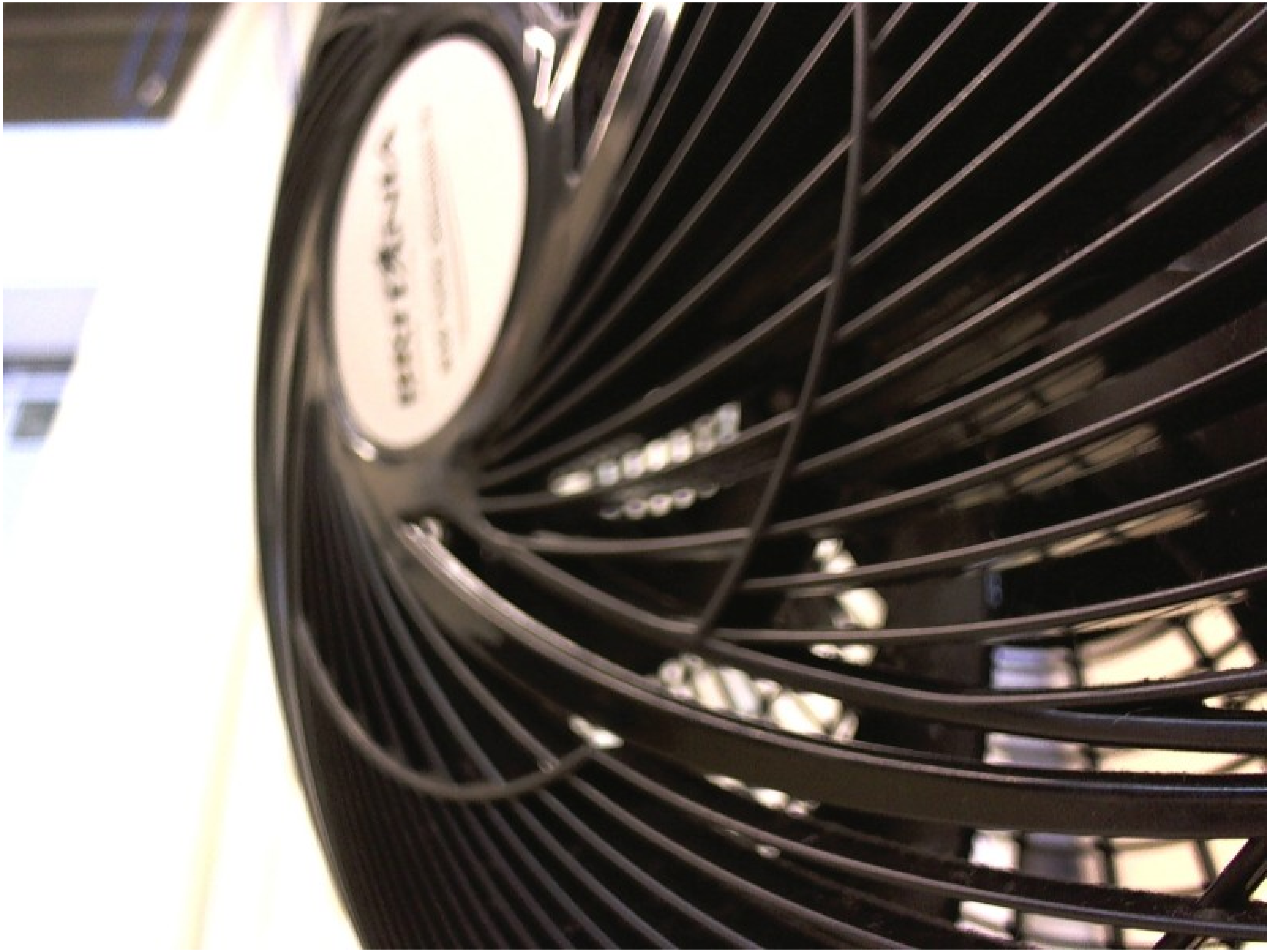
3

4

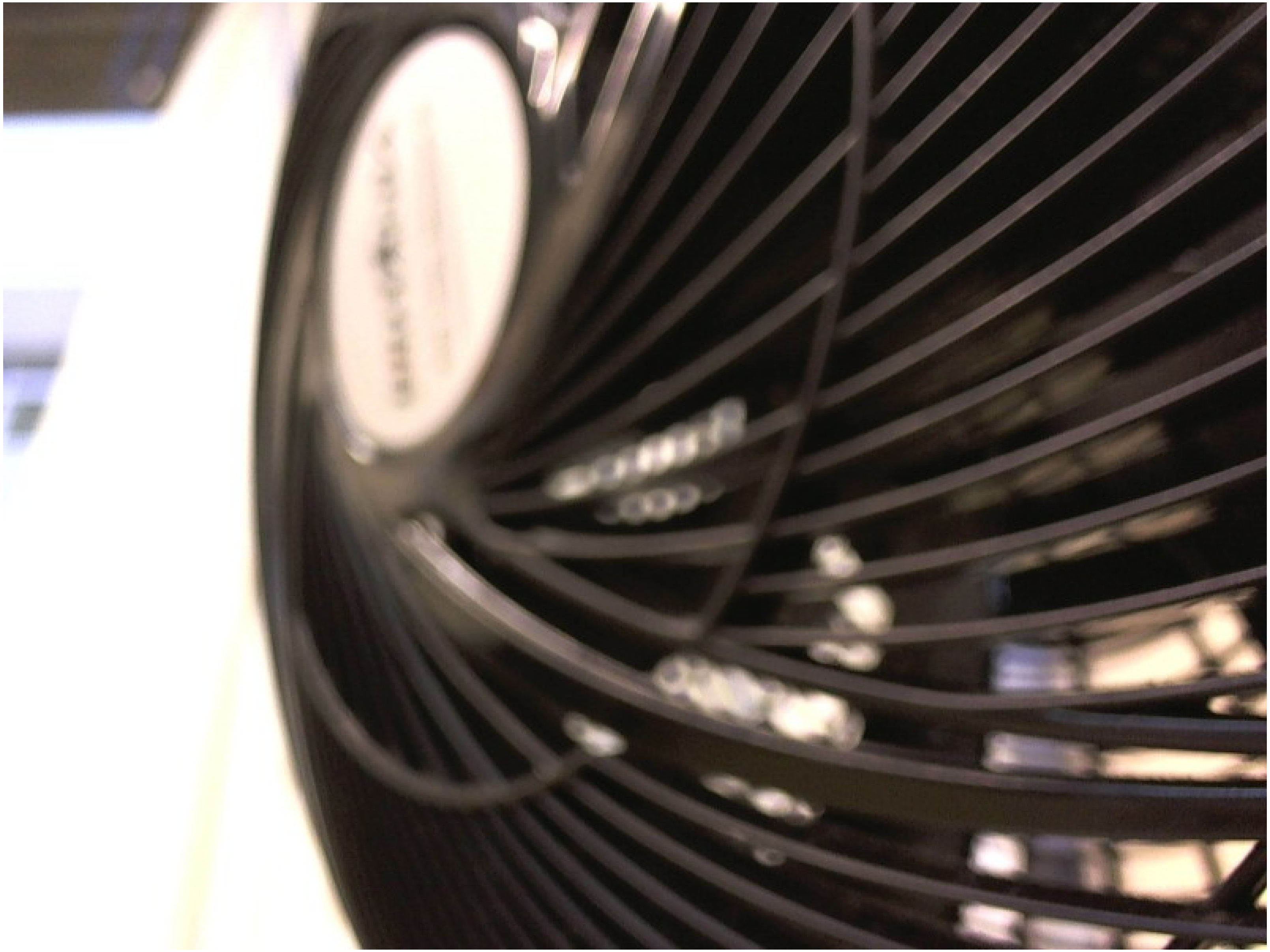
5

6

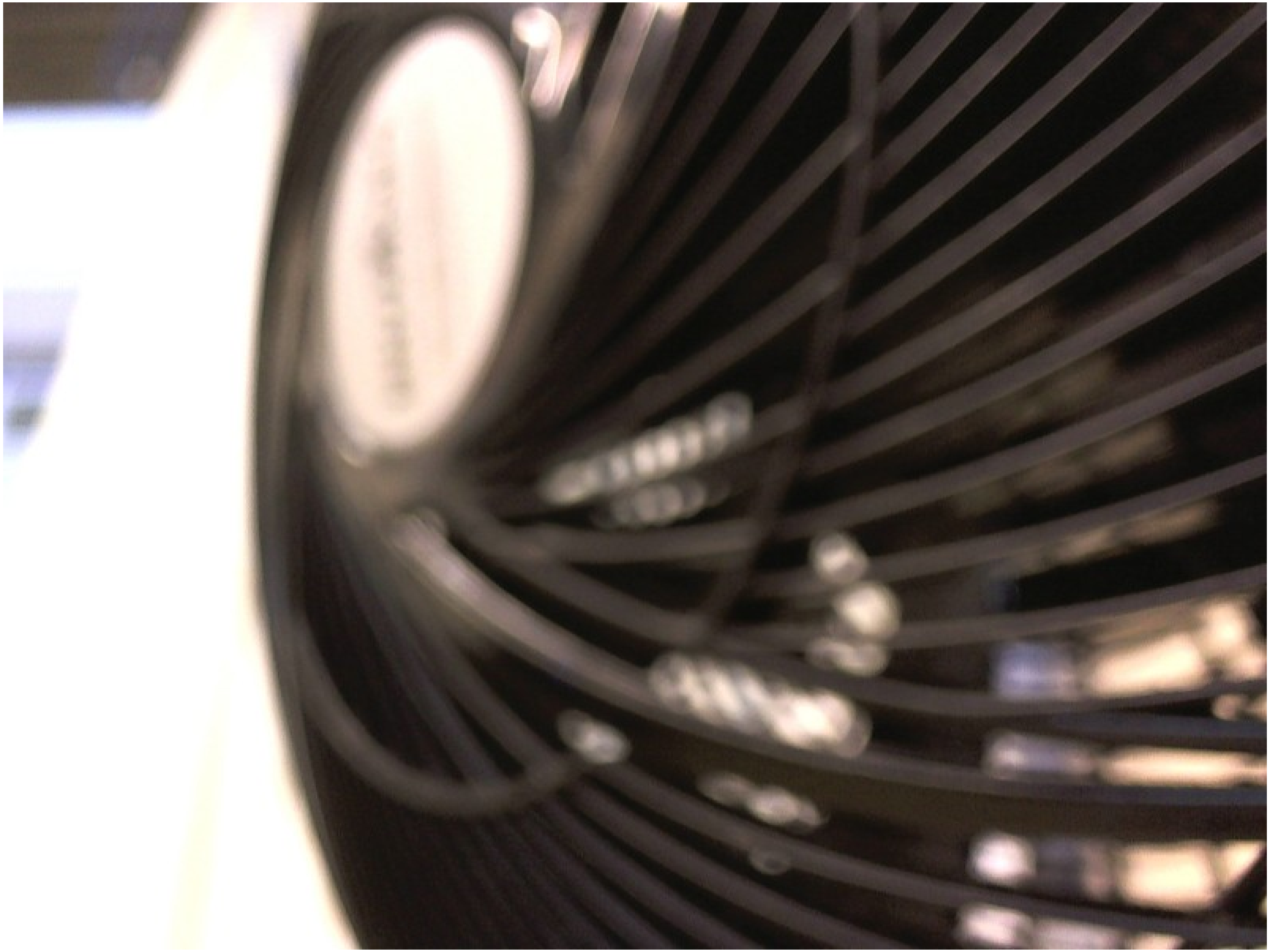
7



8



9



UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DFT: Modificando e reconstruindo

- Podemos modificar a magnitude ou a fase e reconstruir a imagem.

$$mag(k) = \sqrt{F(k)^2 + F'(k)^2}$$

$$fase(k) = atan2(F'(k)/F(k))$$

$$\Downarrow$$

Muda!

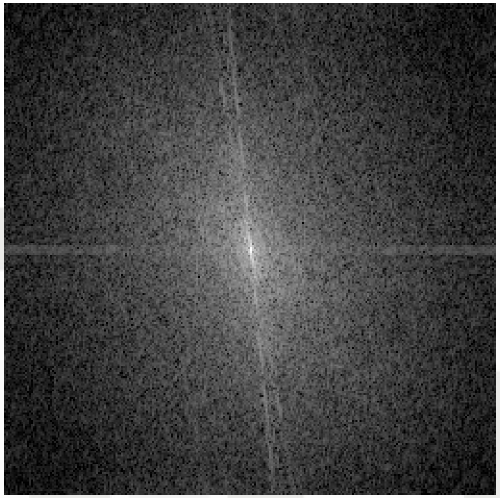
$$\Downarrow$$

$$F(k) = mag(k) \cdot \cos(fase(k))$$

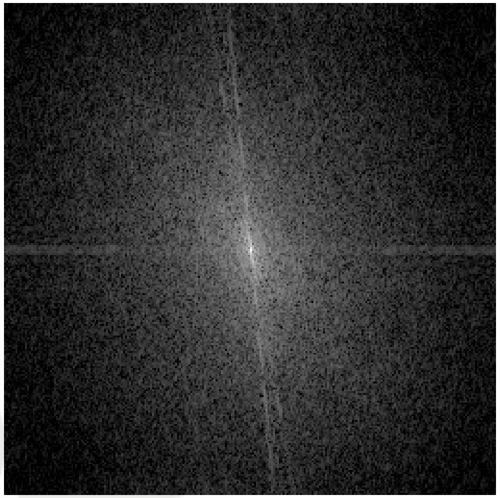
$$F'(k) = mag(k) \cdot \sin(fase(k))$$

Exemplo

Imagem original



Magnitude

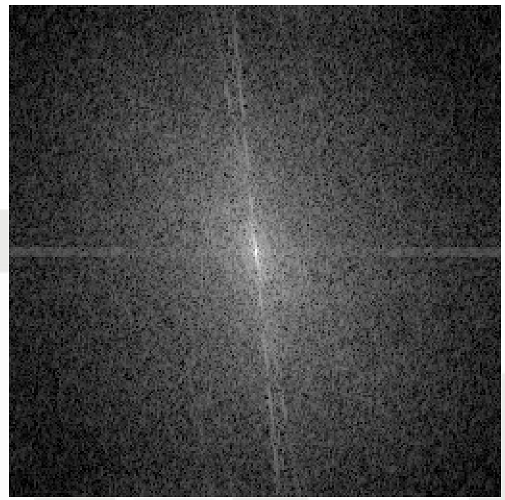


Magnitude * 0.5

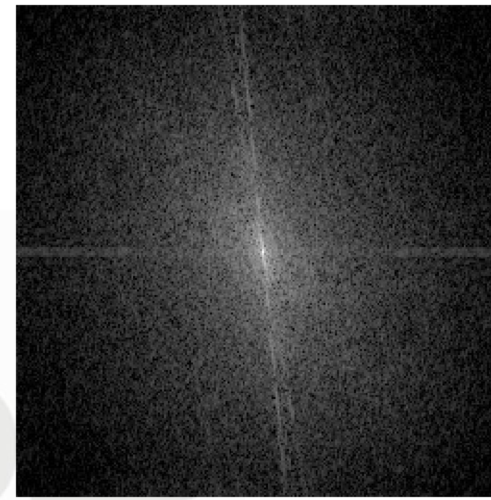
Reconstruída

Exemplo

Imagem original



Magnitude



Magnitude * 0.5

Reconstruída

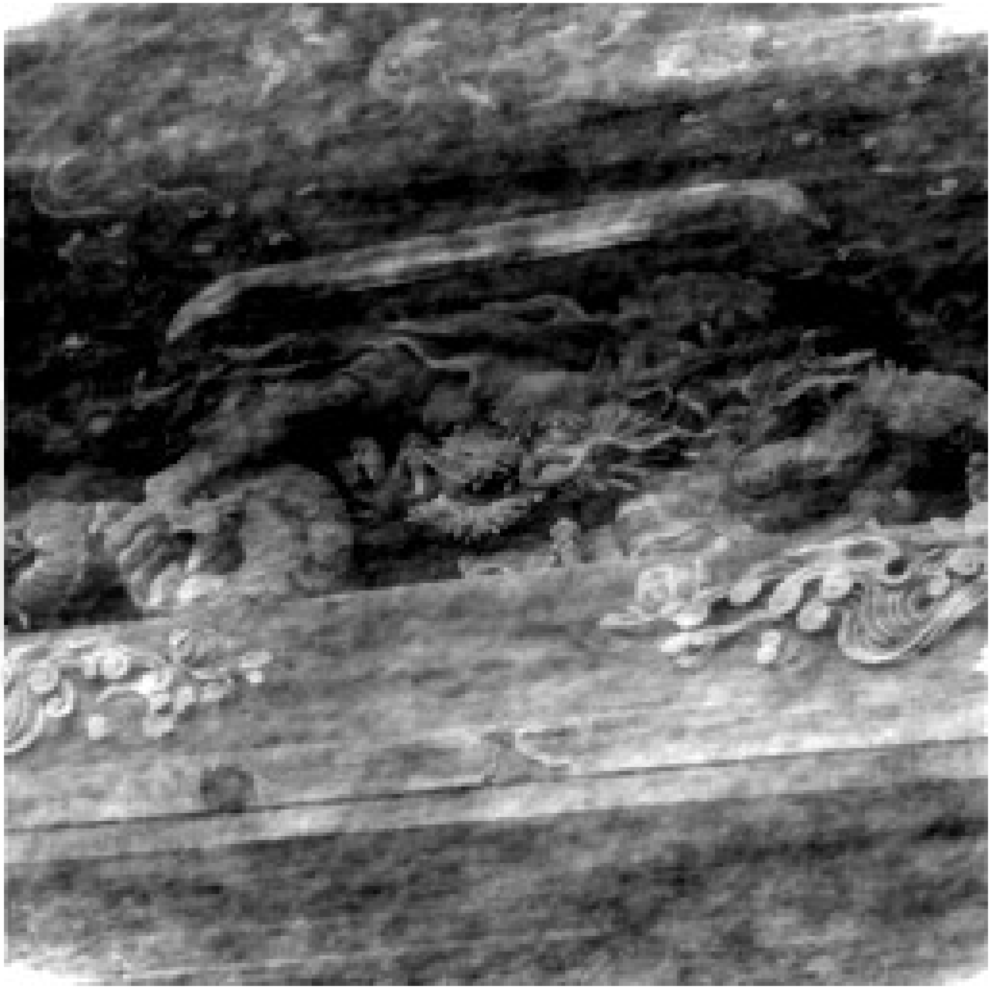
Exemplo

Imagem original

fase * 0.9

Exemplo

Imagem original



fase * 0.5

Exemplo

Imagem original

mag + 1.0

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo

Imagem original

fase + 0.5

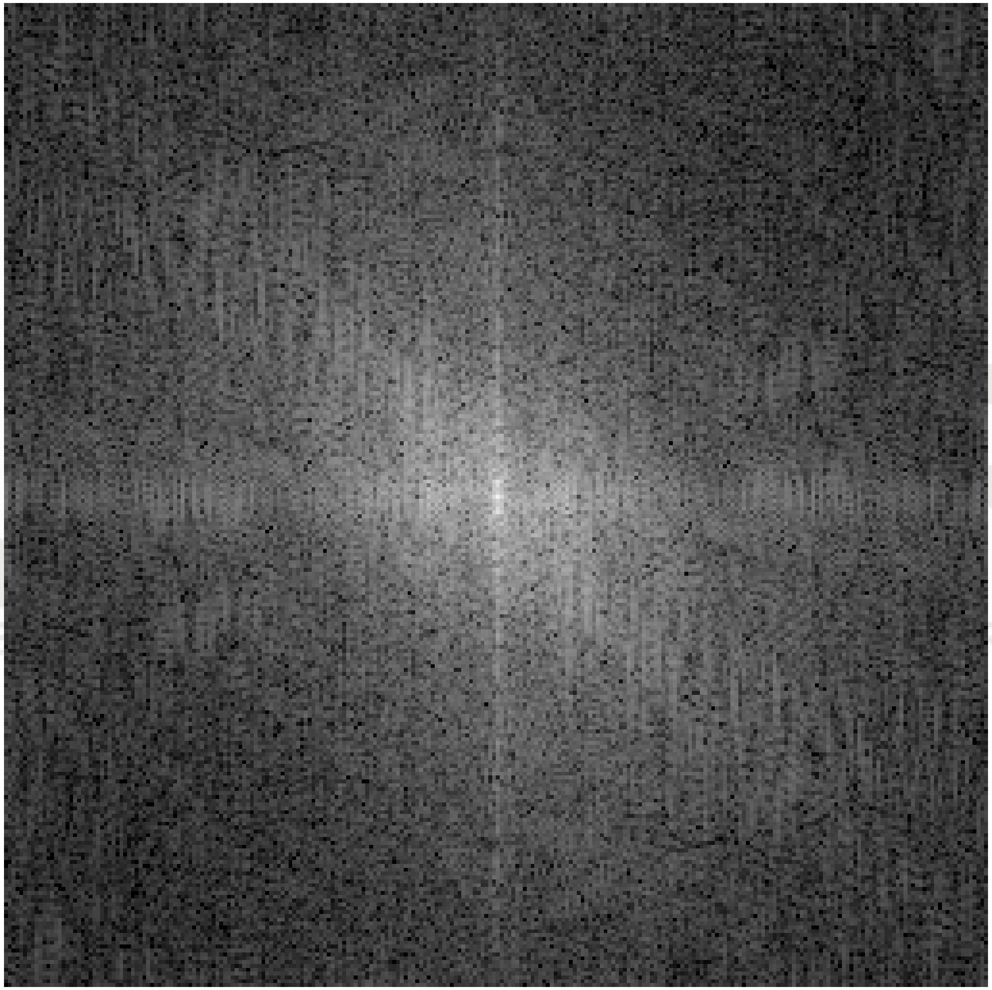
Modificando e reconstruindo

- Tentar modificar aspectos como brilho, rotação e translação a partir da DFT é computacionalmente caro, trabalhoso e pouco intuitivo.
- Que tipo de tarefa poderia se beneficiar da DFT?
  - Vejamos um exemplo no qual:
    - A imagem de entrada tem NxN pixels.
    - A magnitude é forçada para 0 em todas as posições com distância maior que N/4 do centro da imagem.
    - Vamos testar também forçar para 0 em todas as posições com distância maior que N/8 do centro da imagem.

Modificando e reconstruindo

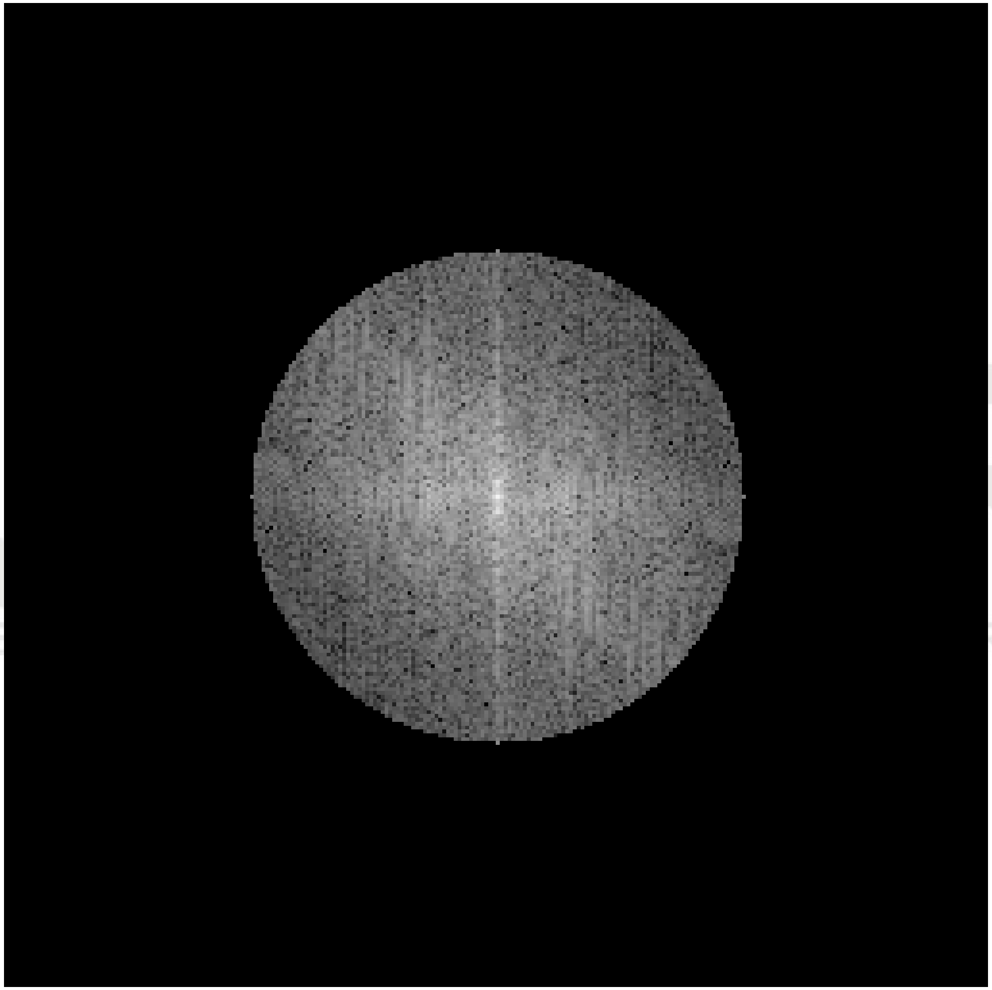
Imagem original

Modificando e reconstruindo



Magnitude

Modificando e reconstruindo

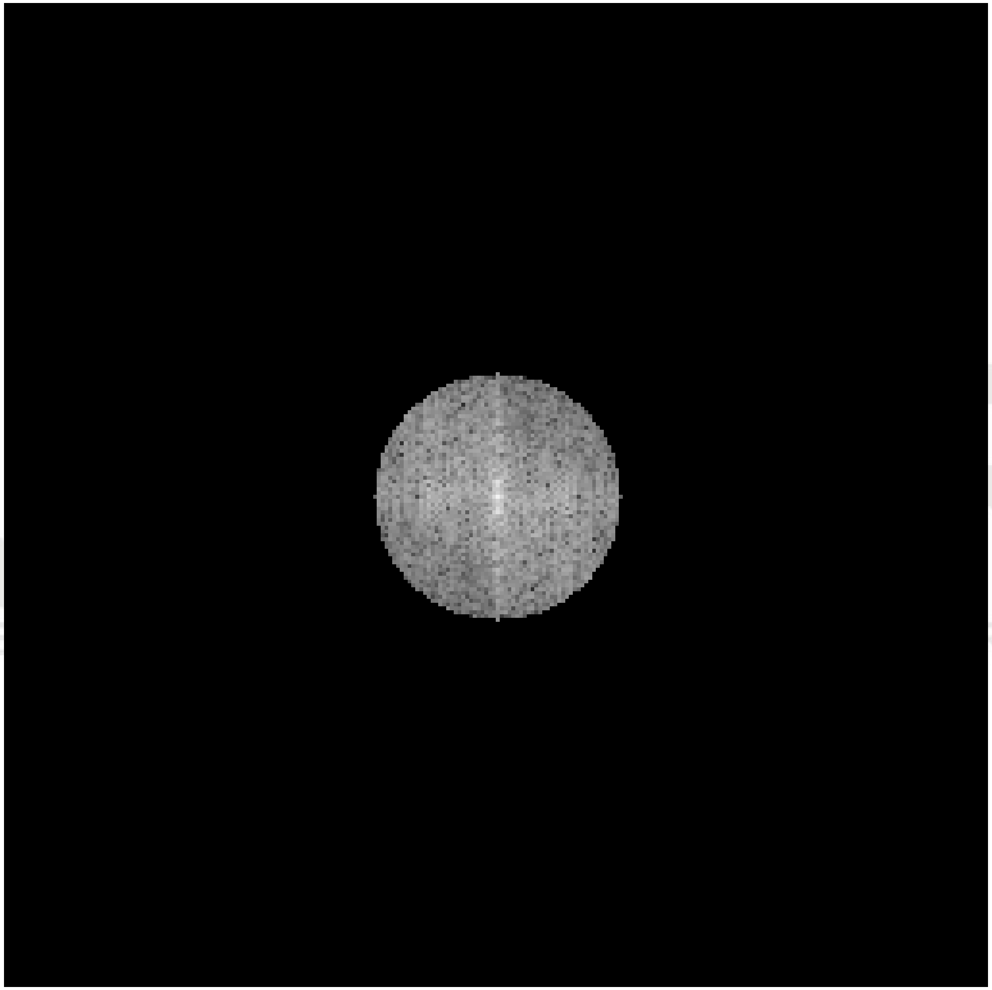


Magnitude alterada

Modificando e reconstruindo

Reconstruída

Modificando e reconstruindo



Magnitude alterada

Modificando e reconstruindo

Reconstruída

Filtragem no domínio da frequência

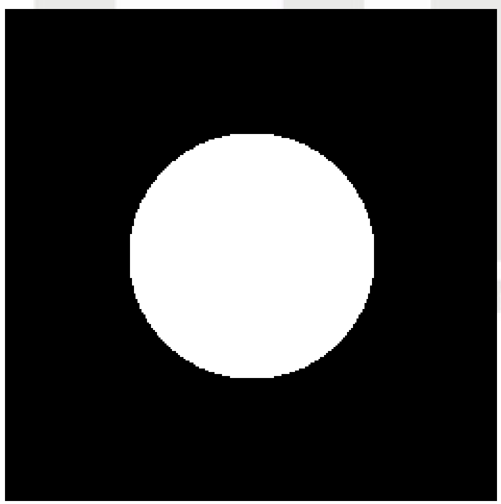
- A filtragem no domínio da frequência consiste em:
  - Computar a DFT da imagem.
  - Alterar a magnitude da DFT.
    - Este tipo de filtro é chamado de “filtro com deslocamento de fase 0”.
  - Reconstruir a imagem.
- Convolução no domínio da frequência: uso mais comum:
  - O filtro é uma imagem com o mesmo tamanho da imagem de entrada.
  - A metade esquerda é igual à metade direita espelhada.
  - A metade superior é igual à metade inferior espelhada.
  - Cada ponto da magnitude é multiplicado pela posição correspondente no filtro:

$$novamag(x,y) = mag(x,y) \cdot filtro(x,y)$$

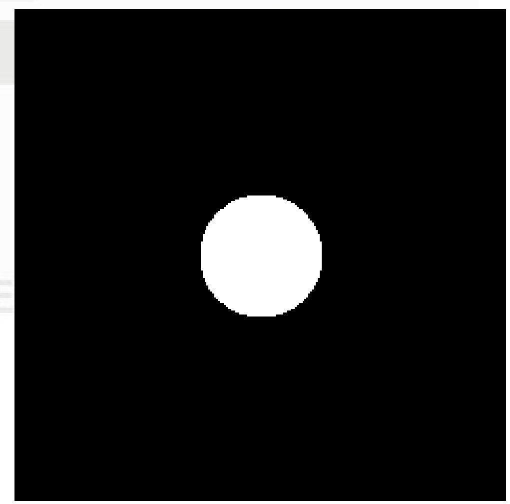
- Toda convolução no domínio da frequência tem uma equivalente no domínio espacial!

Filtragem no domínio da frequência

- Os filtros do exemplo anterior são filtros *passa-baixa* ideais.
  - Por que "*passa-baixa*"?
  - Por que "ideais"?



Corta a partir de uma distância N/4 do centro



Corta a partir de uma distância N/8 do centro

Filtragem passa-baixa

- Para evitar os artefatos gerados pelos filtros passa-baixa ideais, podemos usar um filtro Gaussiano.
  - O filtro é gerado do mesmo jeito que o kernel de um filtro Gaussiano no domínio espacial!

Filtragem Gaussiana

- Para evitar os artefatos gerados pelos filtros passa-baixa ideais, podemos usar um filtro Gaussiano.
  - O filtro é gerado do mesmo jeito que o kernel de um filtro Gaussiano no domínio espacial!

```
for (cada linha i)
{
   y = i - altura/2
   gy = exp (-(y²)/(2σ²))

   for (cada coluna j)
   {
      x = j - largura/2
      gx = exp (-(x²)/(2σ²))

      filtro (i,j) = gx * gy
   }
}
```

Este código **gera** os coeficientes do filtro. A **aplicação** do filtro é uma multiplicação da magnitude da DFT pelos coeficientes.

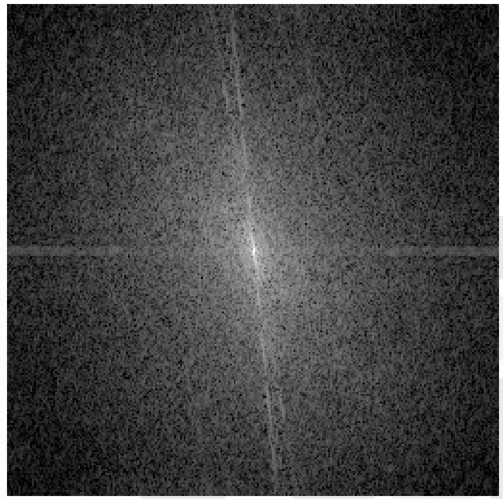
Este código pode ser otimizado...

Filtragem Gaussiana

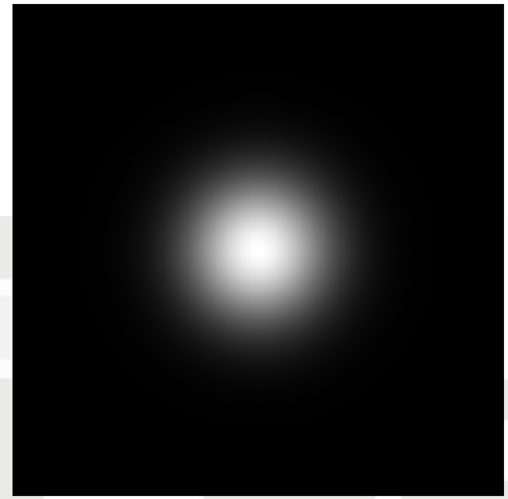
```
for (i de 0 até N/2)
{
   y = i – N/2
   gy = exp (-(y²)/(2σ²))

   for (j de 0 até M/2)
   {
      x = j – M/2
      gx = exp (-(x²)/(2σ²))

      filtro (i,j) = filtro (i,M-1-j) =
      filtro (N-1-i,j) = filtro (N-1-i,M-1-j) = gx * gy
   }
}
```

Exemplo

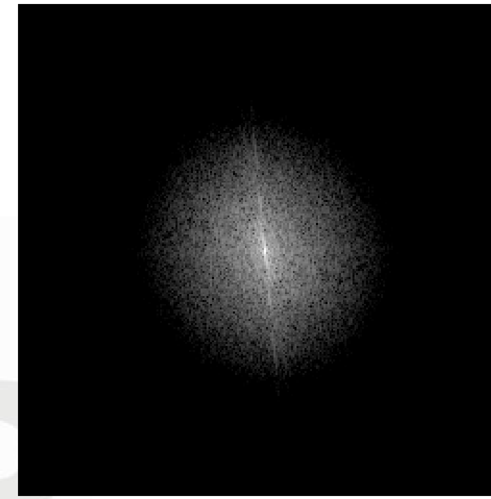
Imagem original



Magnitude



Filtro (σ=26)

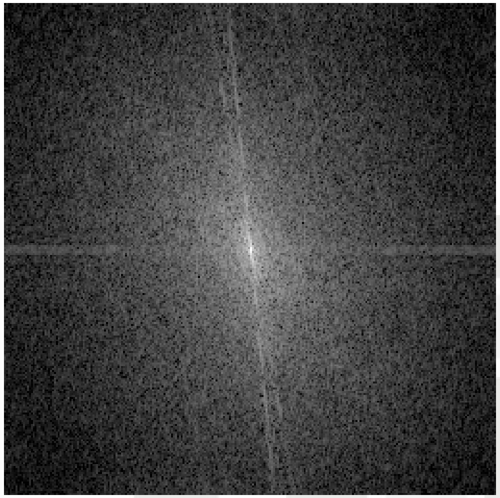


Magnitude · Filtro

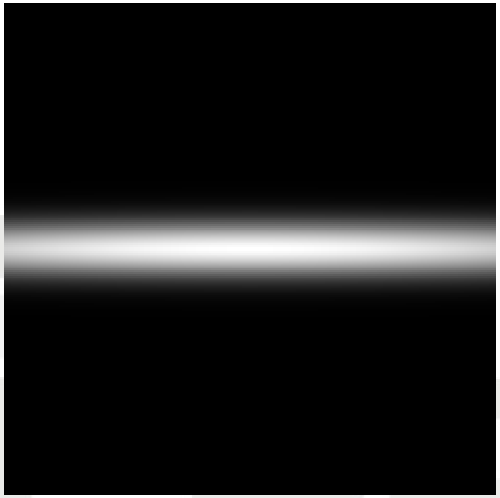
Imagem reconstruída

Exemplo

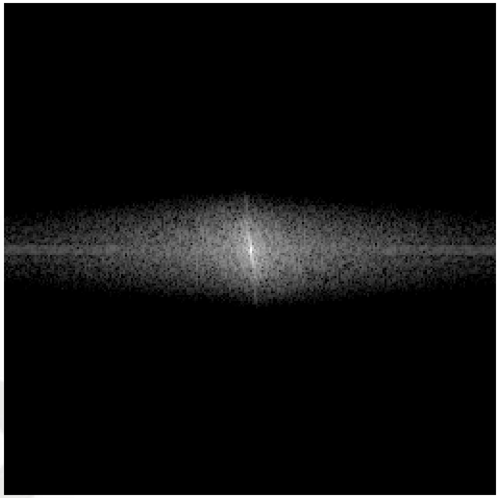
Imagem original



Magnitude



Filtro (σ_x=200,σ_y=10)

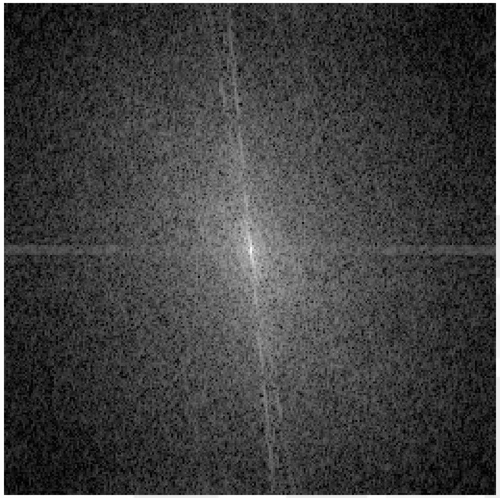


Magnitude · Filtro

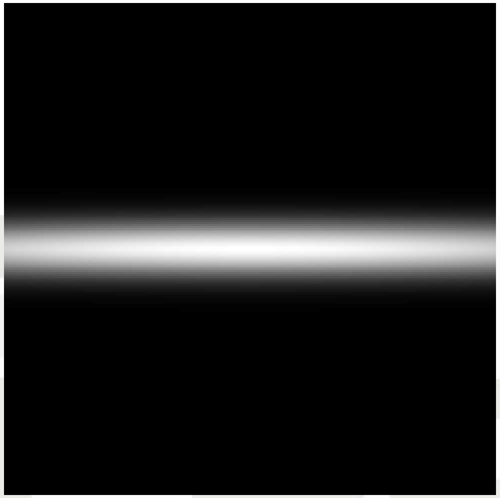
Podemos usar valores diferentes de σ na horizontal e na vertical.

Exemplo

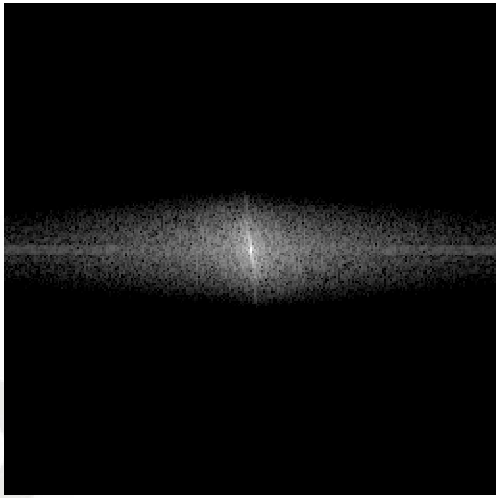
Imagem original



Magnitude



Filtro (σ_x=200,σ_y=10)



Magnitude · Filtro

Podemos usar valores diferentes de σ na horizontal e na vertical.

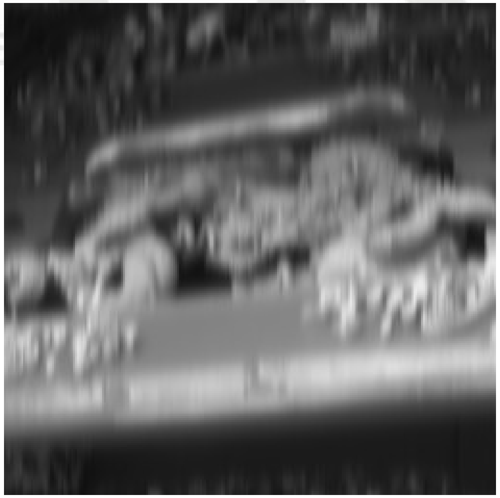


Imagem reconstruída

Domínio espacial x frequência

- Todo filtro Gaussiano no domínio da frequência tem um filtro Gaussiano equivalente no domínio espacial.
  - Mas note que o funcionamento do σ é diferente nos dois casos.
    - Por quê?!
- Em que circunstâncias a filtragem Gaussiana no domínio da frequência é vantajosa?

Filtragem passa-alta

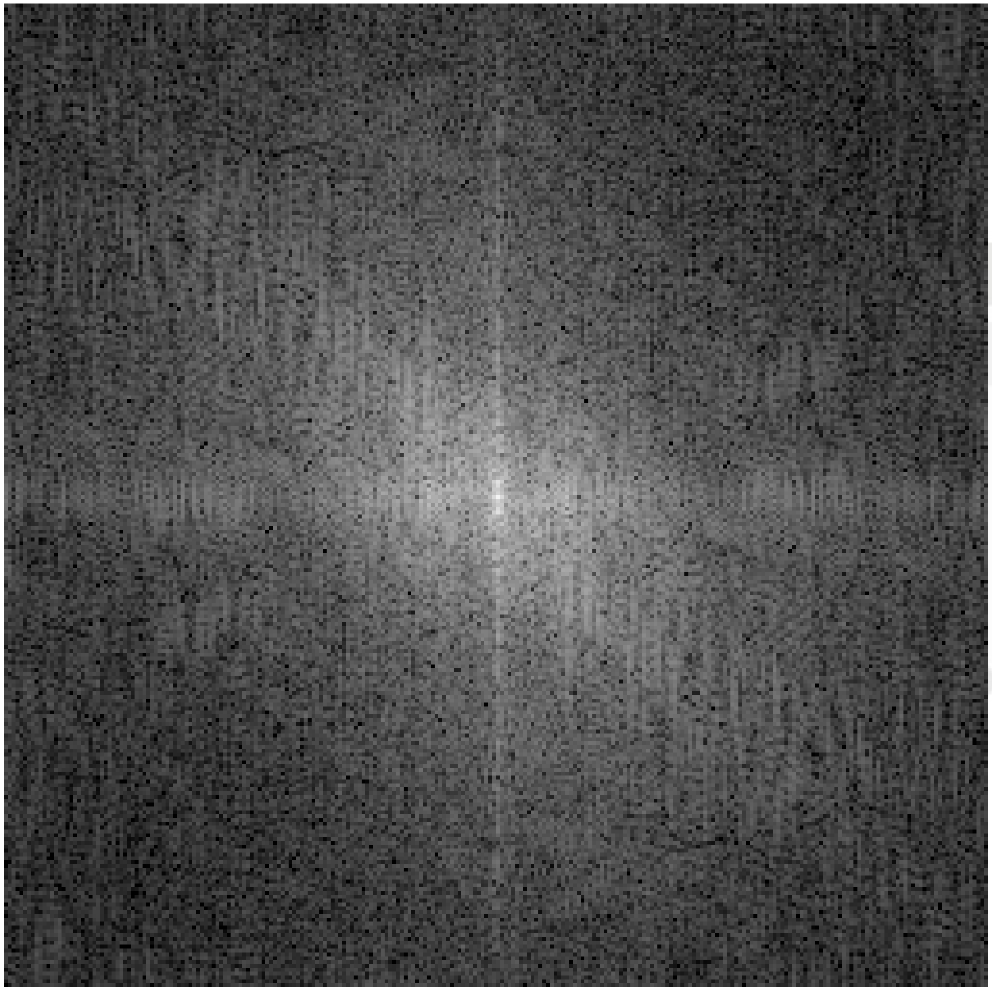
•Assim como podemos cortar frequências mais altas, podemos cortar frequências mais baixas...

Filtragem passa-alta

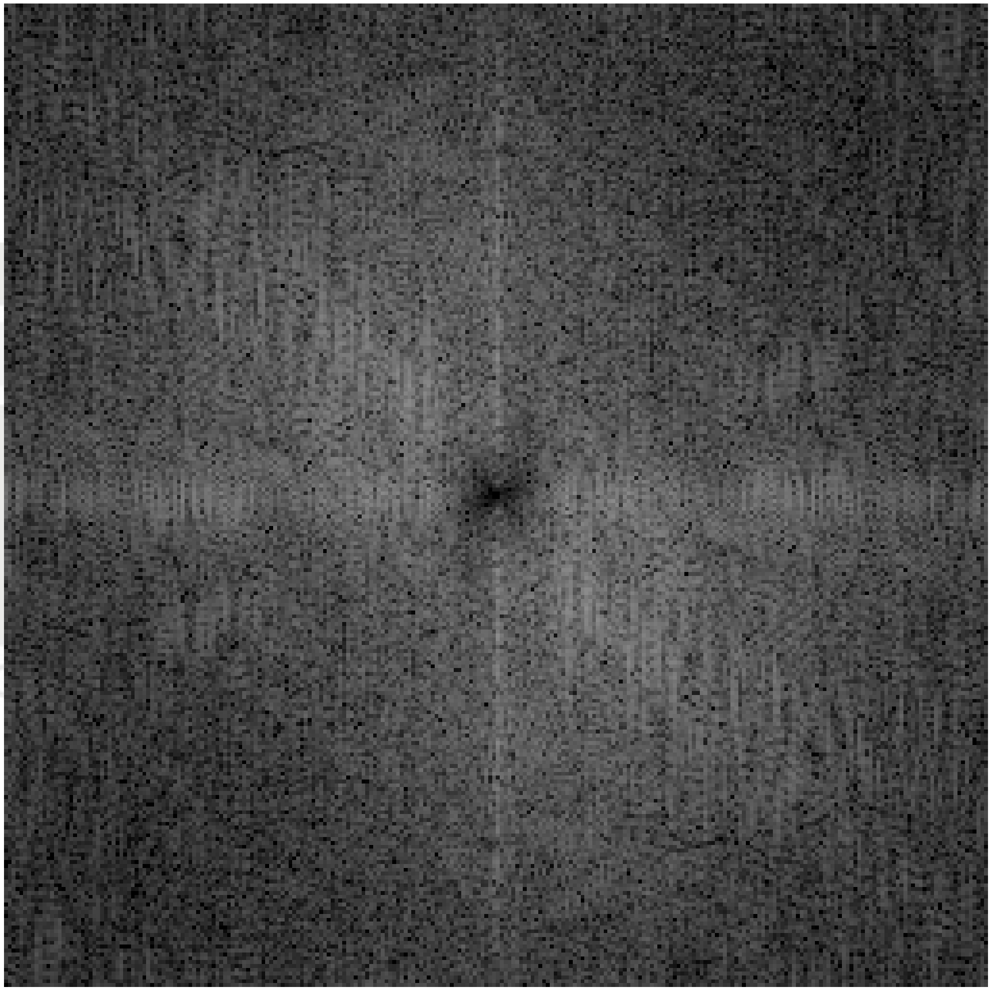
Imagem original

Filtro (1 – filtro Gaussiano com σ = 32)

Filtragem passa-alta



Magnitude



Magnitude alterada

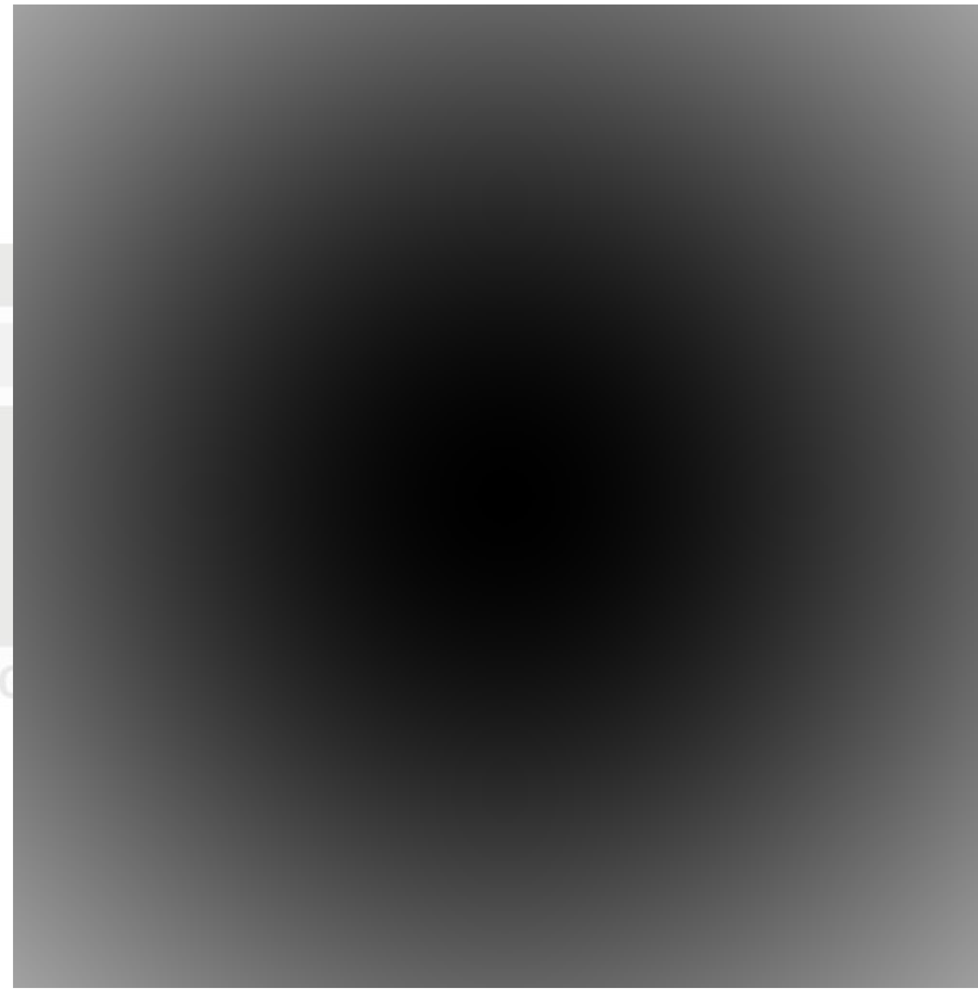
Filtragem passa-alta

Imagem original

Reconstruída

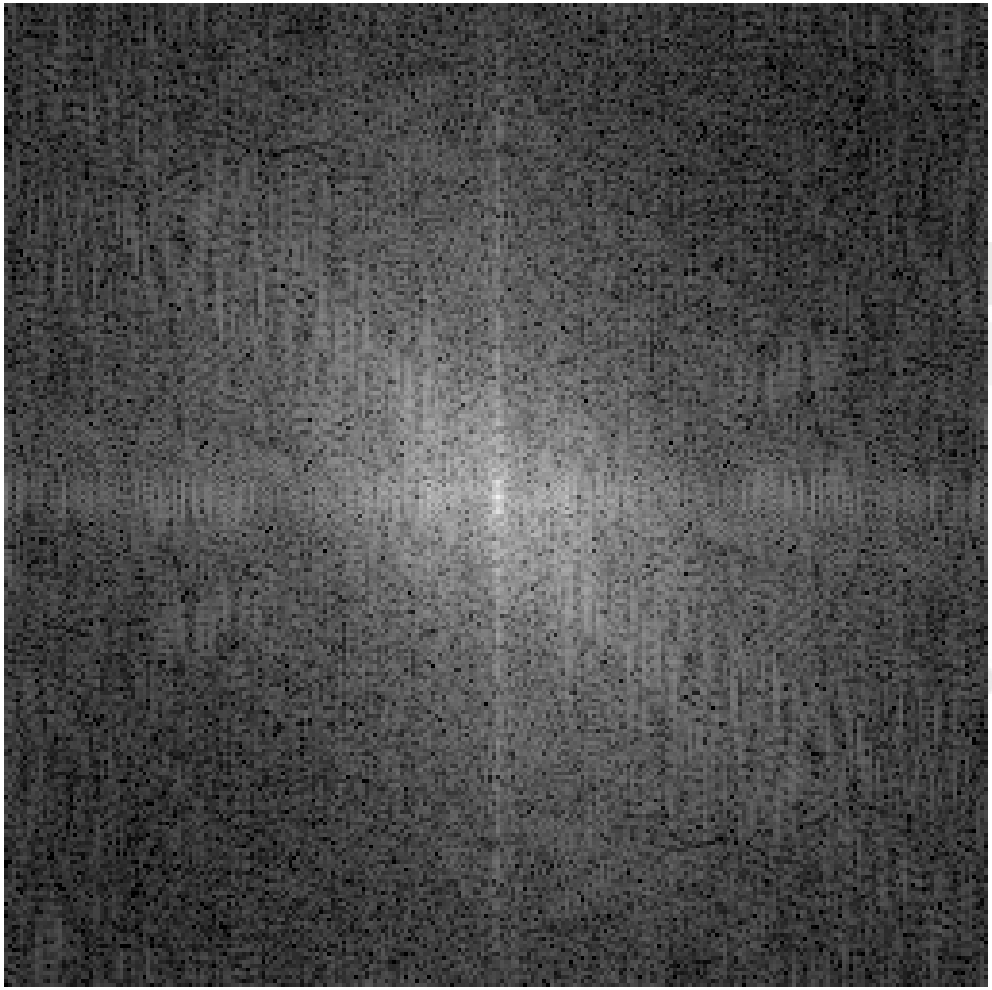
Filtragem passa-alta

Imagem original

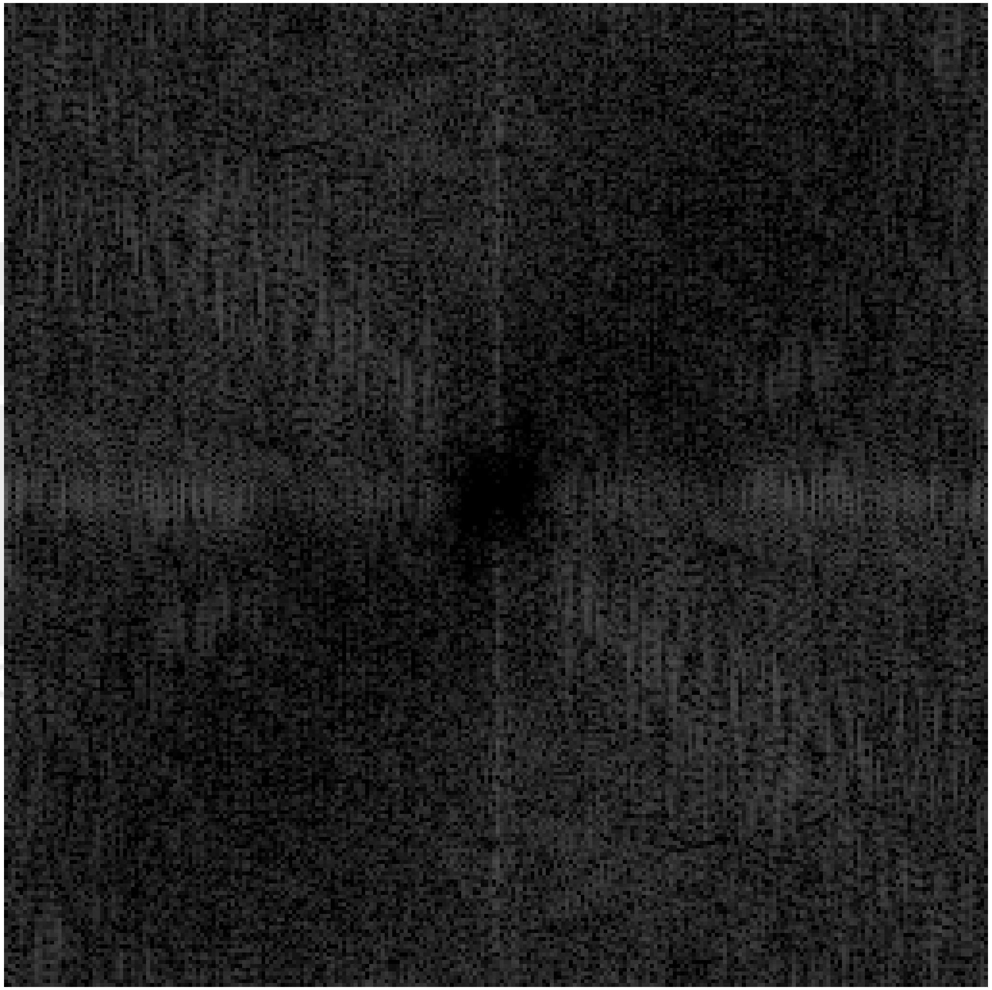


Filtro (1 – filtro Gaussiano com σ = 128)

Filtragem passa-alta



Magnitude



Magnitude alterada

Filtragem passa-alta

Imagem original

Reconstruída

???

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Filtragem passa-alta

Imagem original

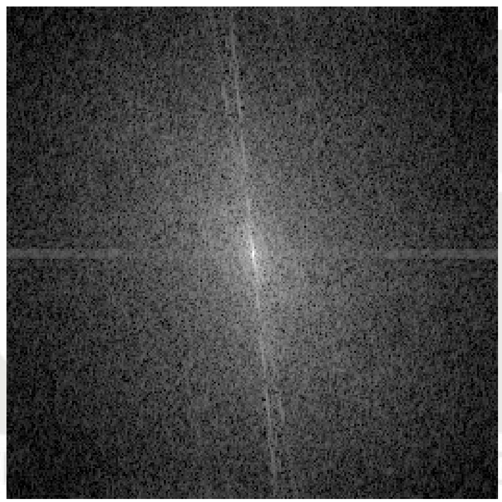
Reconstruída (normalizada)

Realçando frequências

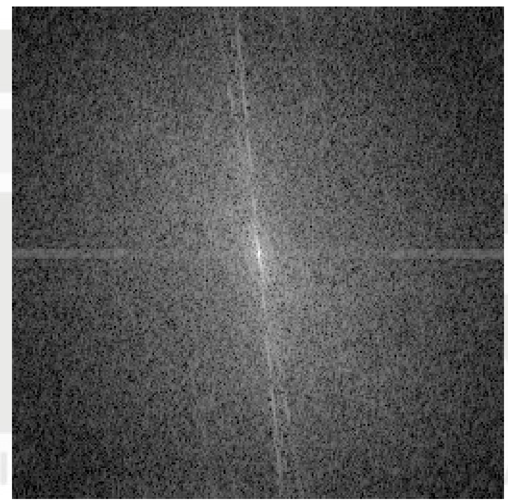
- Se existirem coeficientes com valor maior que 1, o filtro pode realçar certas frequências.
  - O filtro abaixo, de um exemplo anterior, foi gerado subtraindo de 1 um filtro Gaussiano com $\sigma = 32$.
  - Como seria um filtro que soma 1 a cada coeficiente do filtro abaixo?
    - Por que não estamos mostrando este outro filtro?

Exemplo

Imagem original



Magnitude



Magnitude · Filtro

Imagem reconstruída

Exemplo

Imagem original

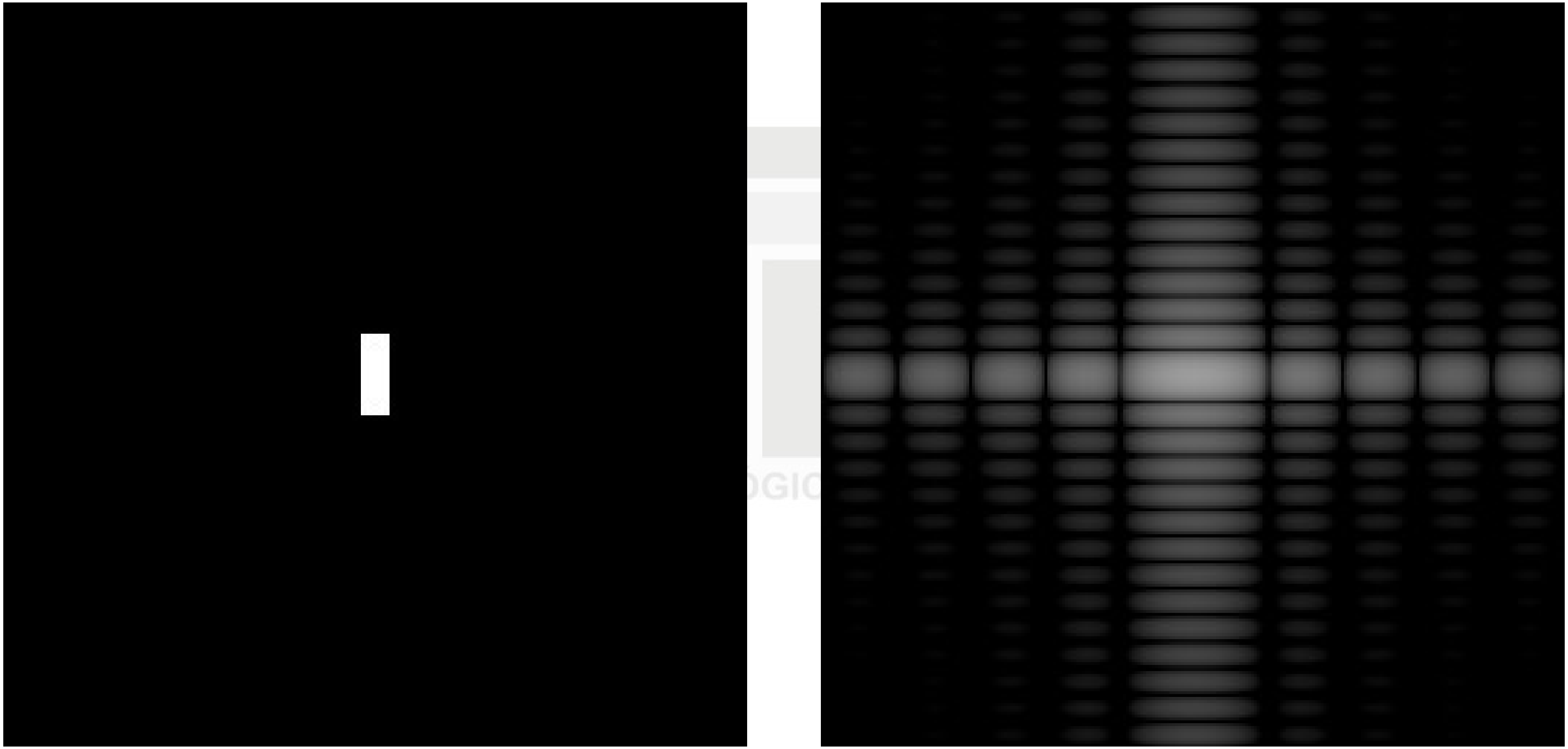
Imagem reconstruída
É uma versão com detalhes realçados!

Um problema...

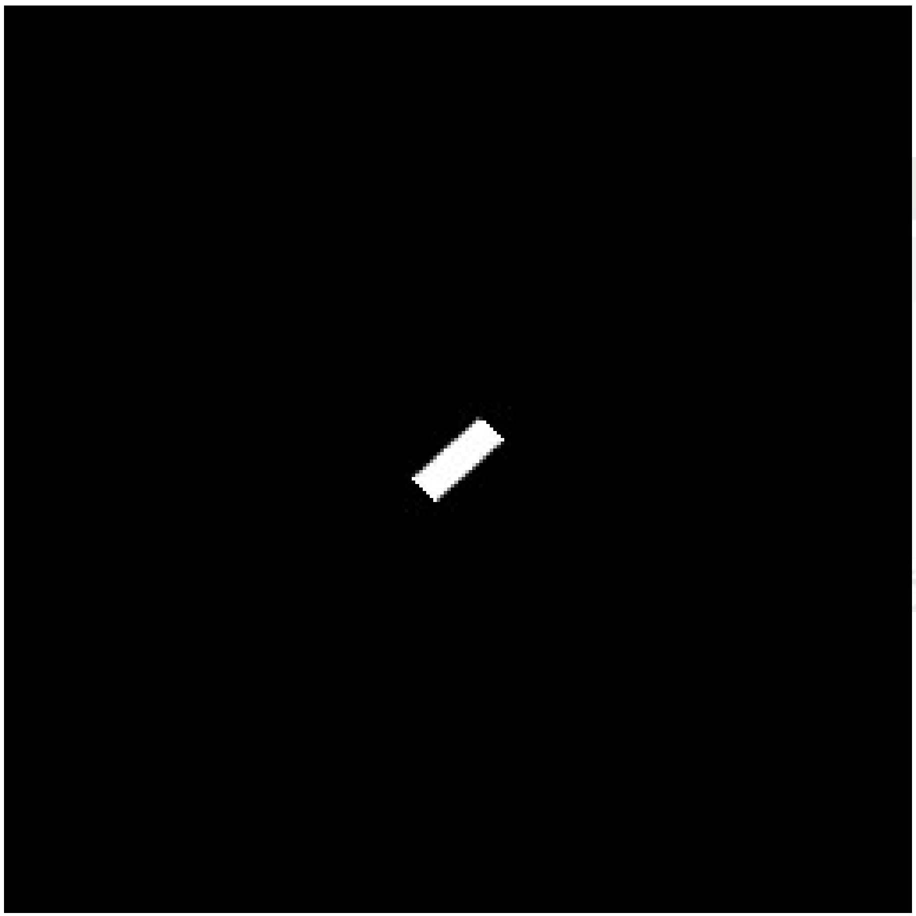
•A transformada de Fourier supõe sinais infinitos.
- Em imagens, normalmente significa imaginar repetições infinitas (*tiles*).

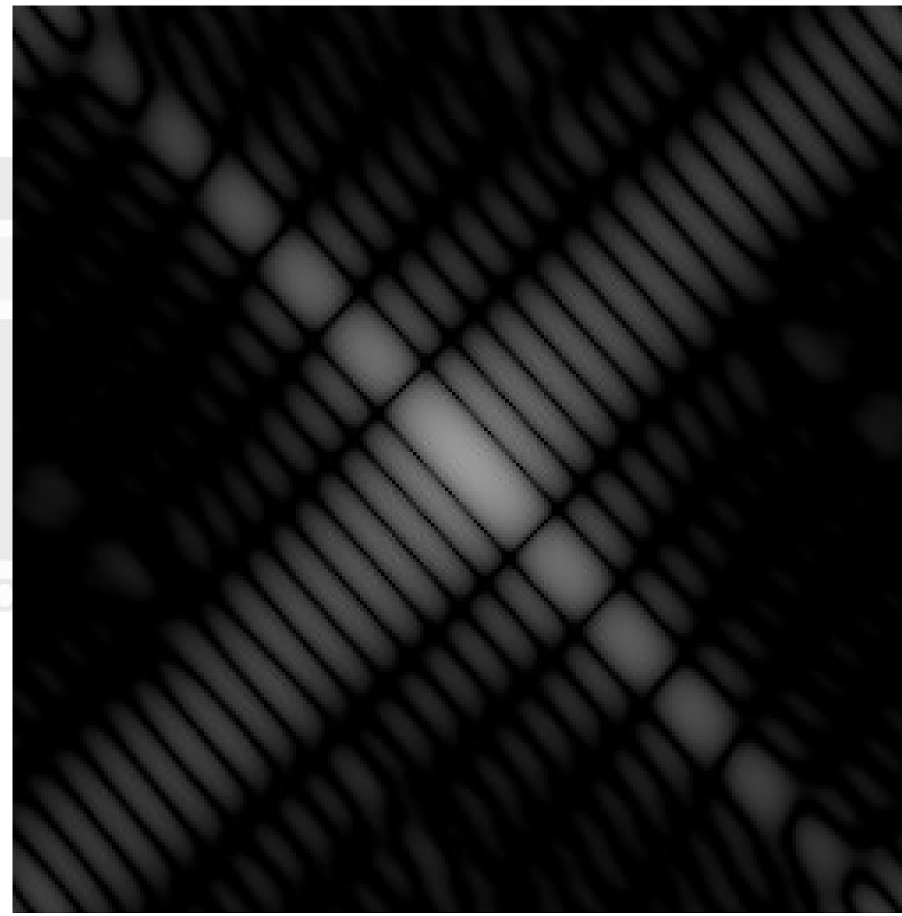
•O espectro pode conter artefatos – frequências “falsas” devido ao contraste entre margens opostas (esquerda x direita, topo x baixo).
- Pode ser problemático para algumas aplicações de análise de sinais.

Sem problemas...

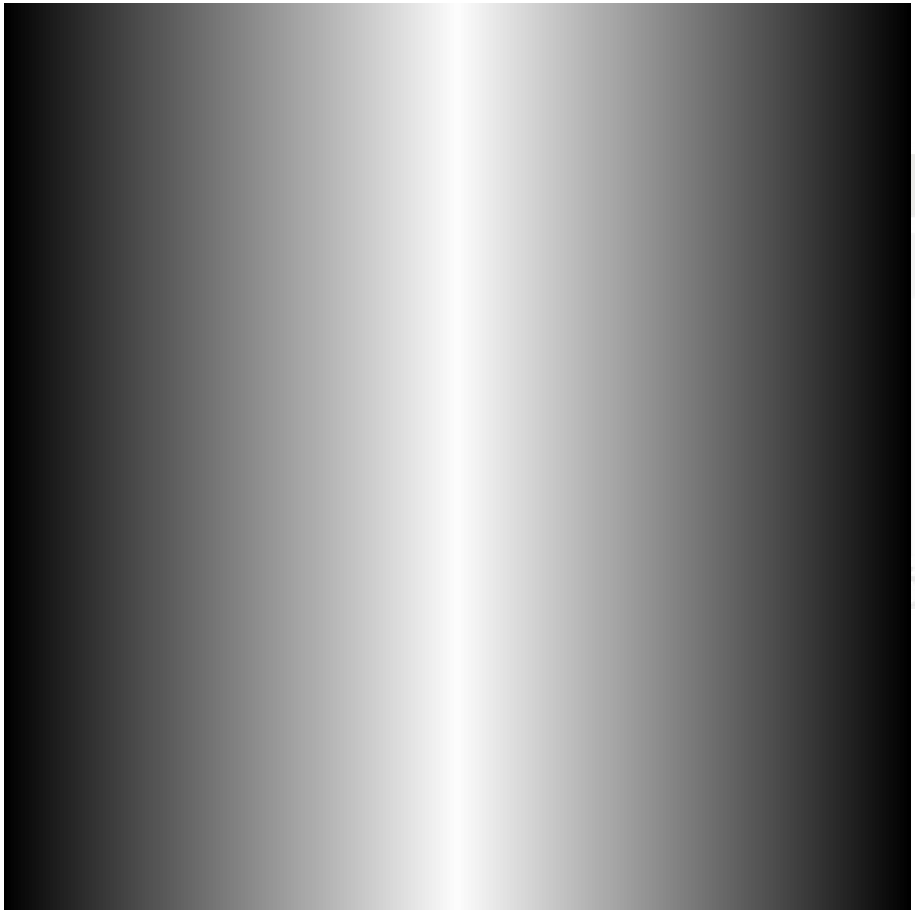


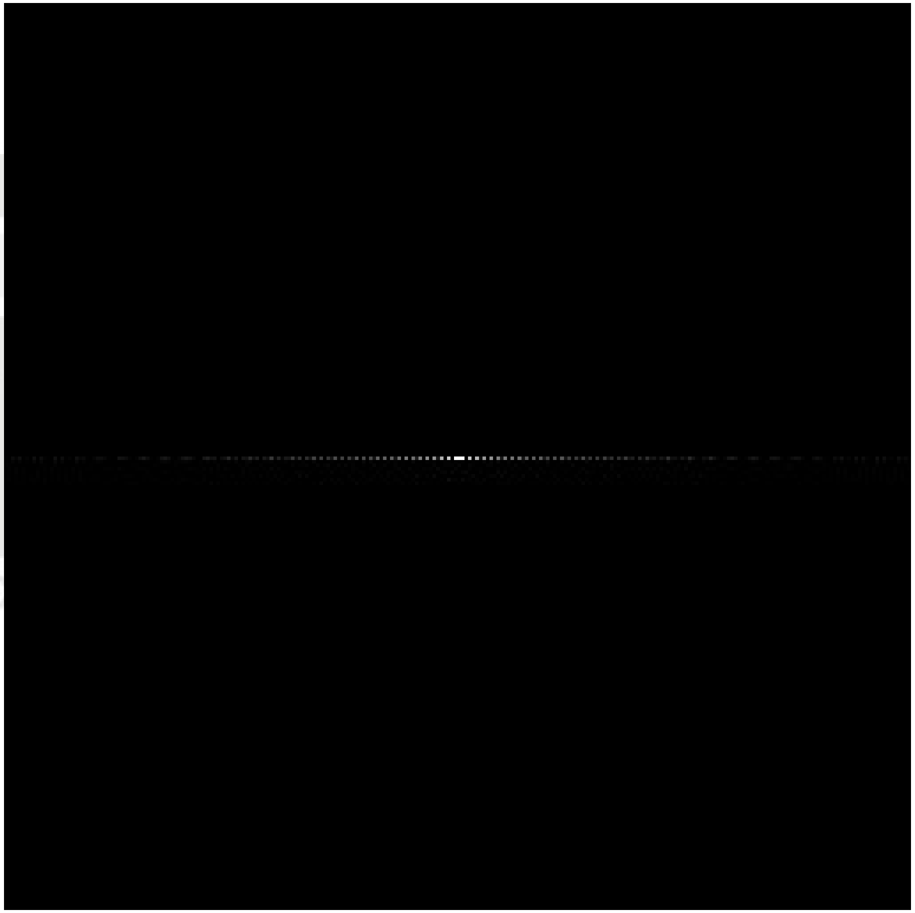
Sem problemas...



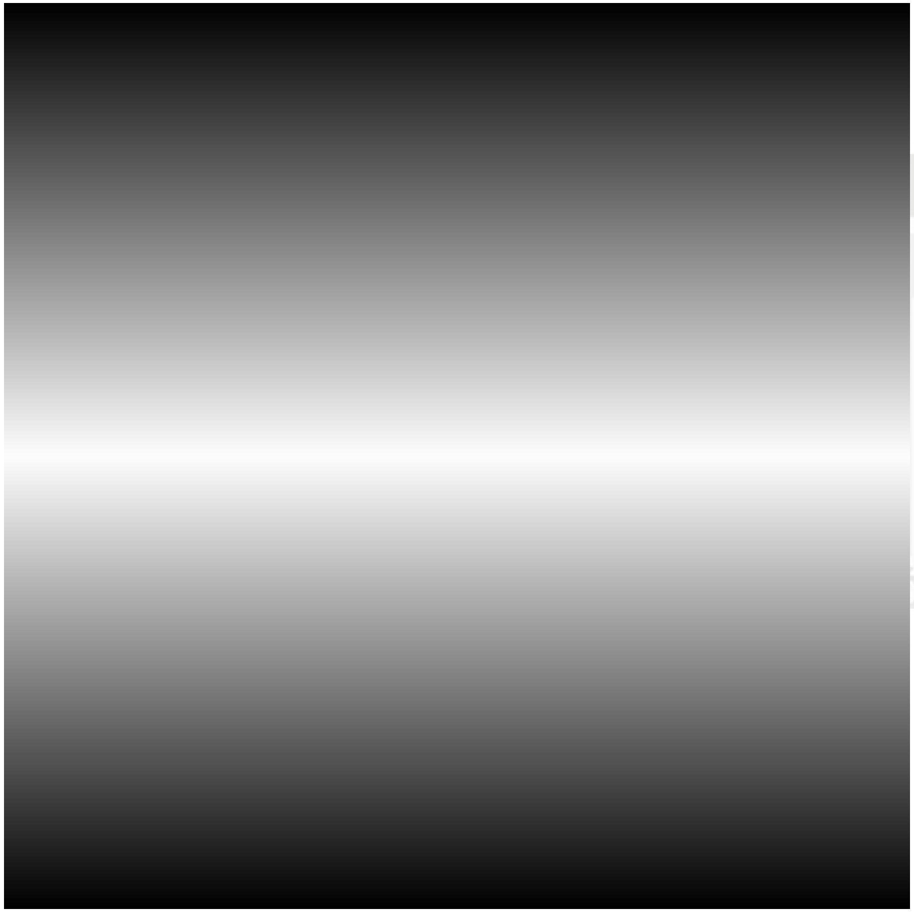


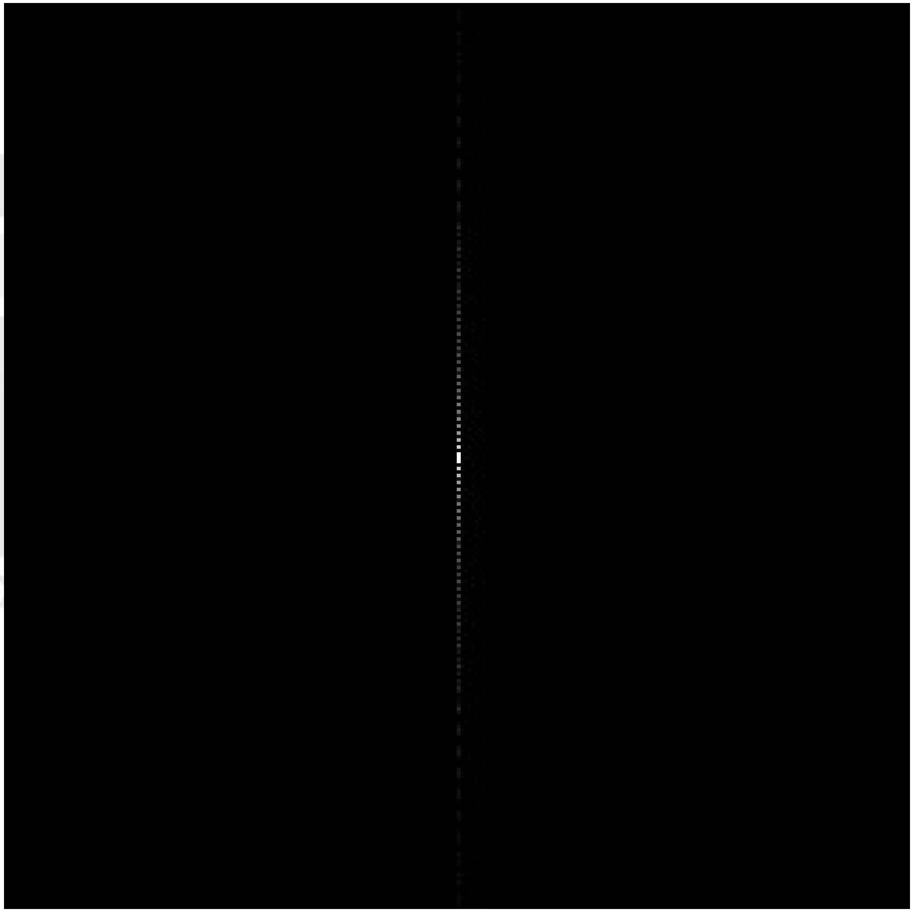
Sem problemas...



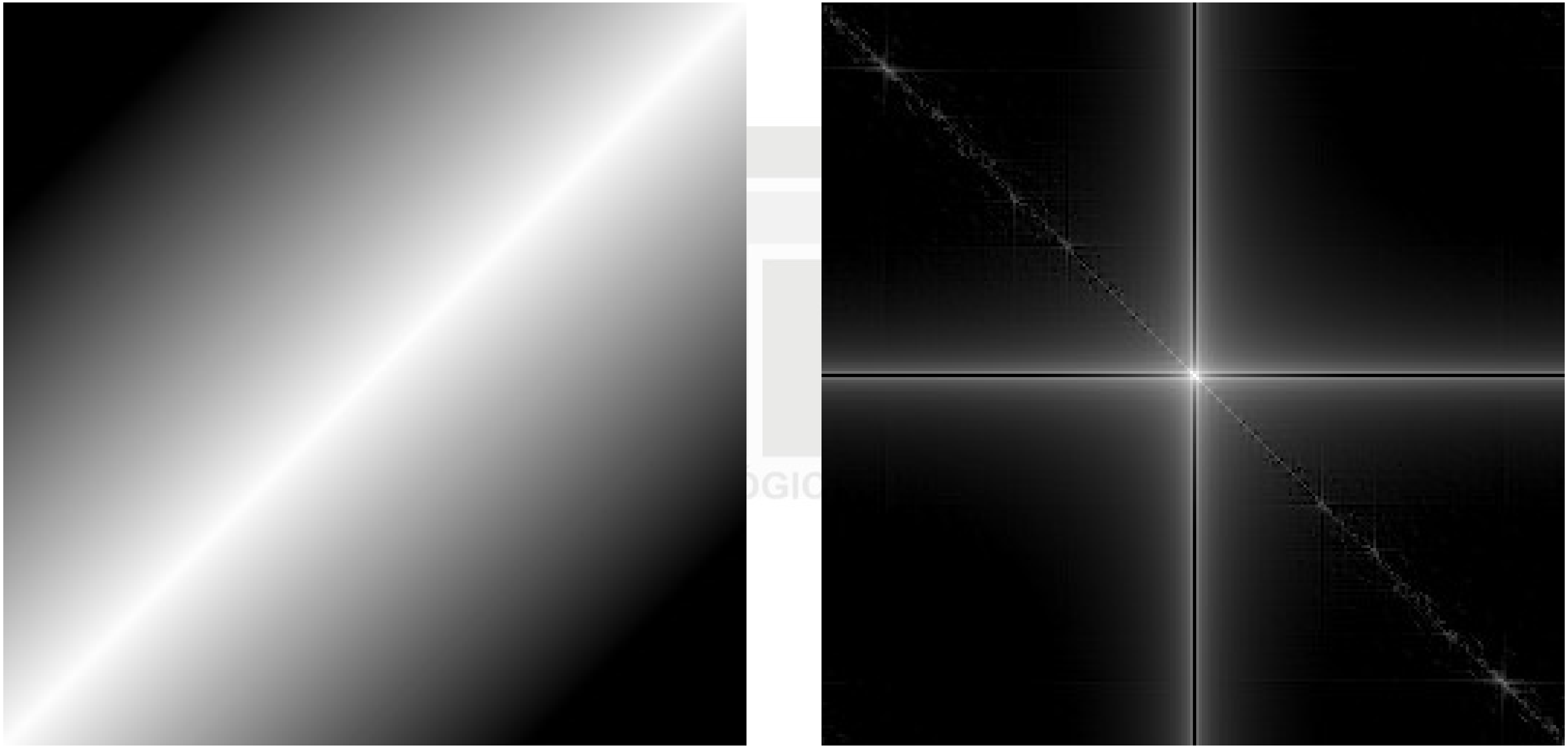


Sem problemas...

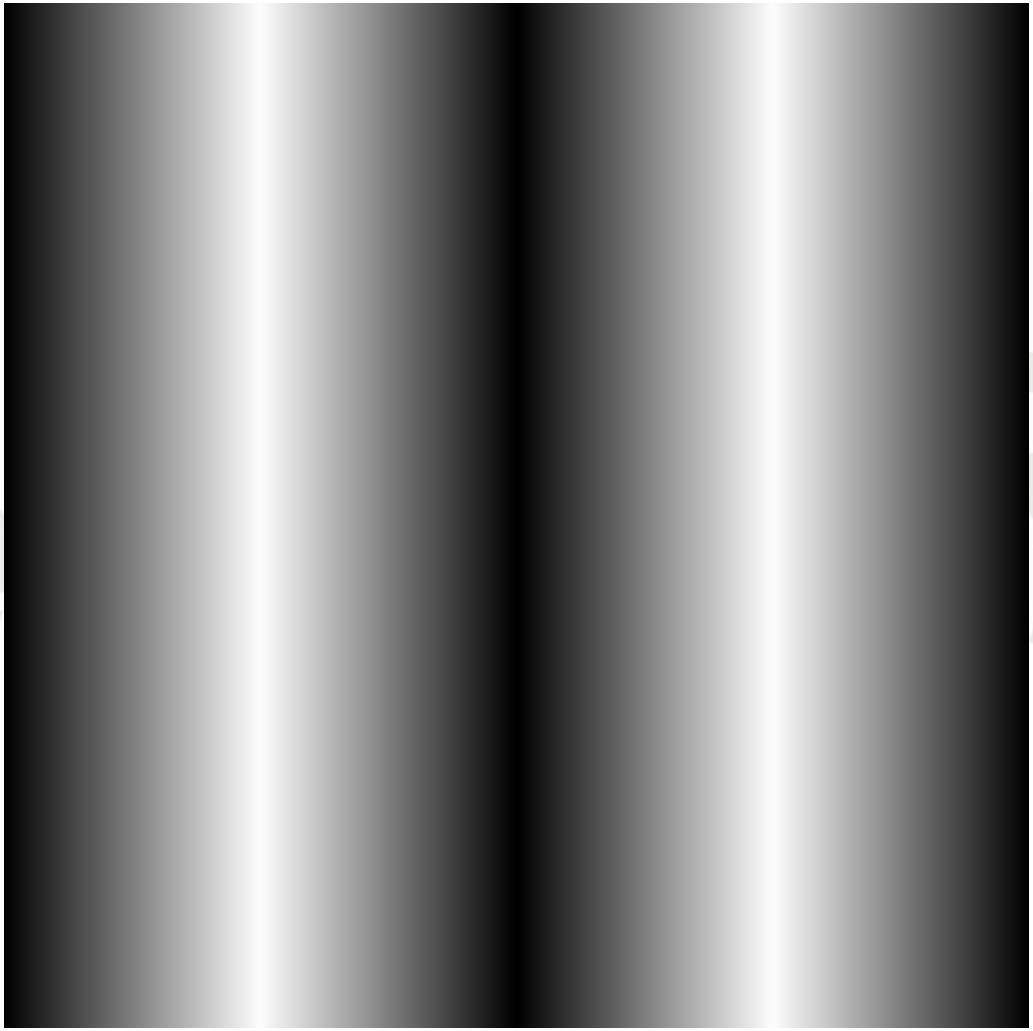




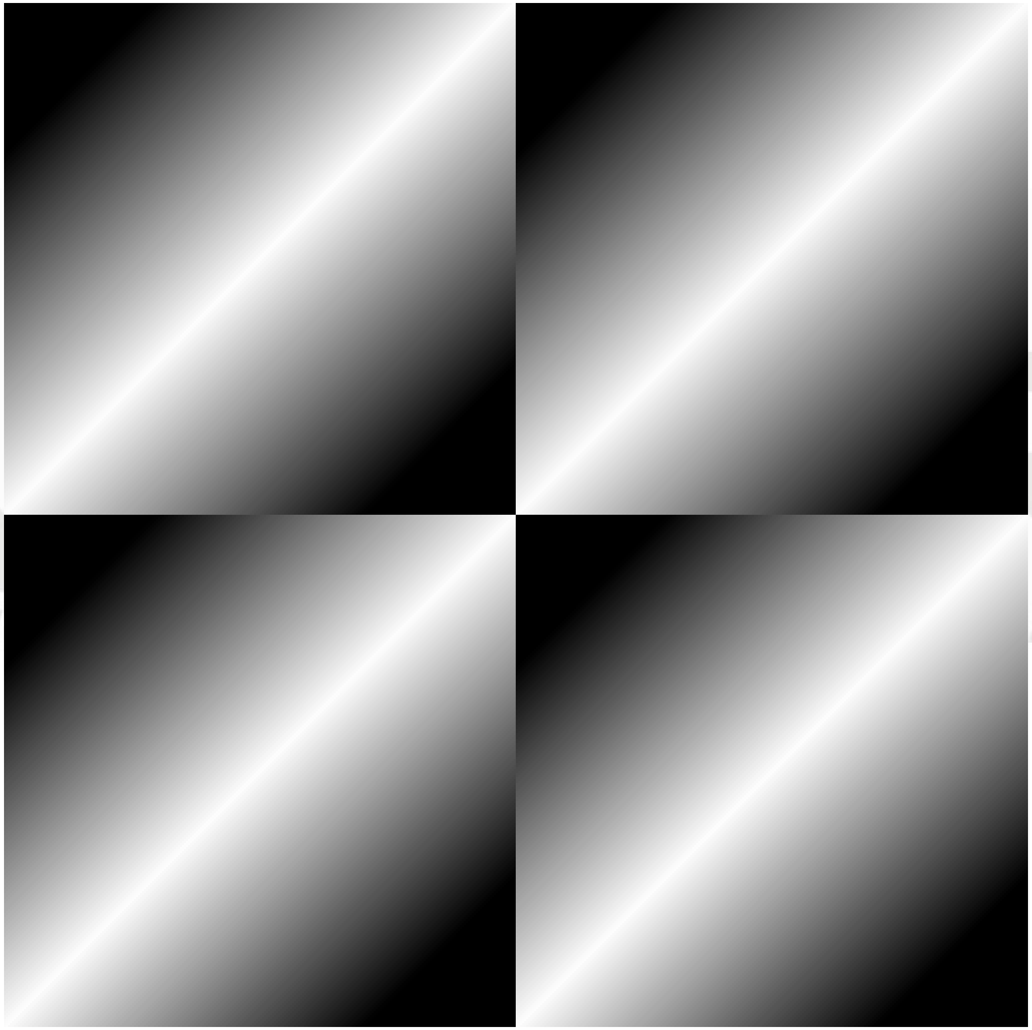
???



???



???

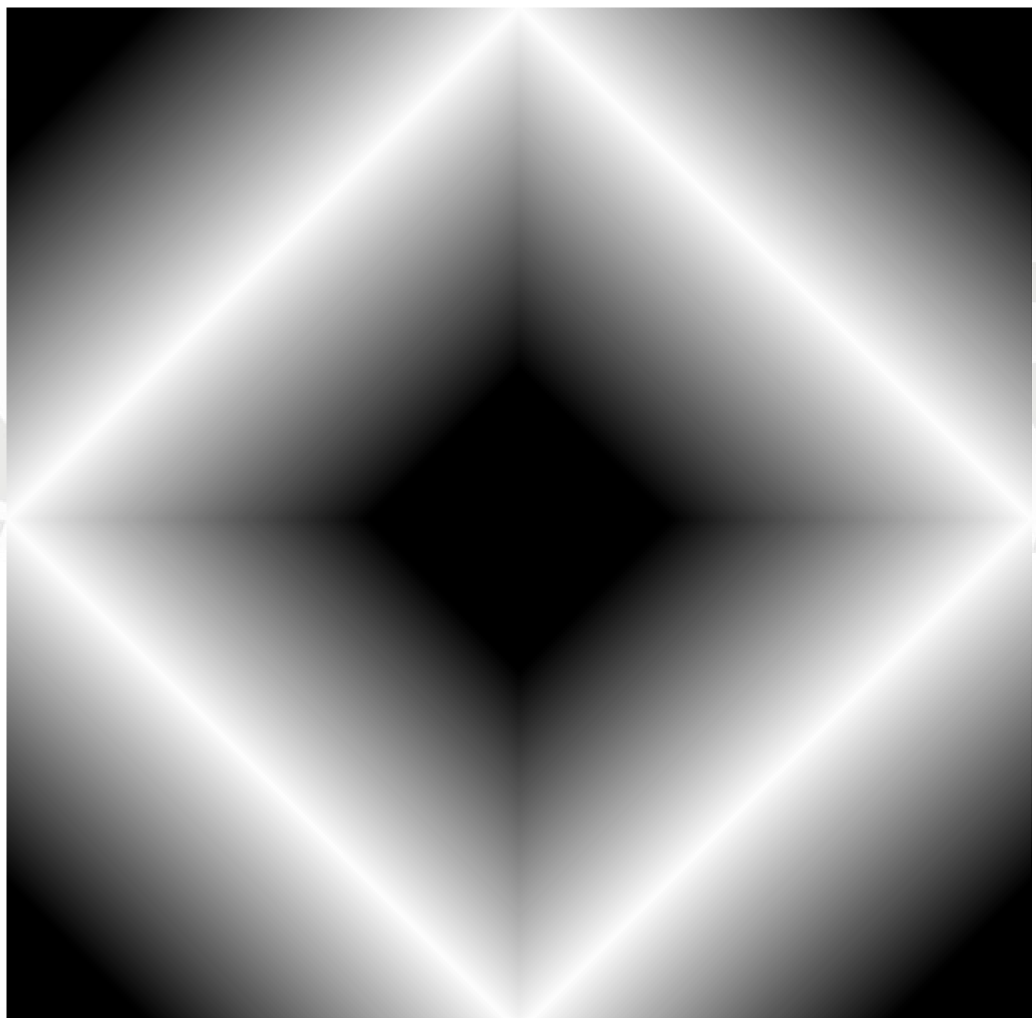


Como resolver?

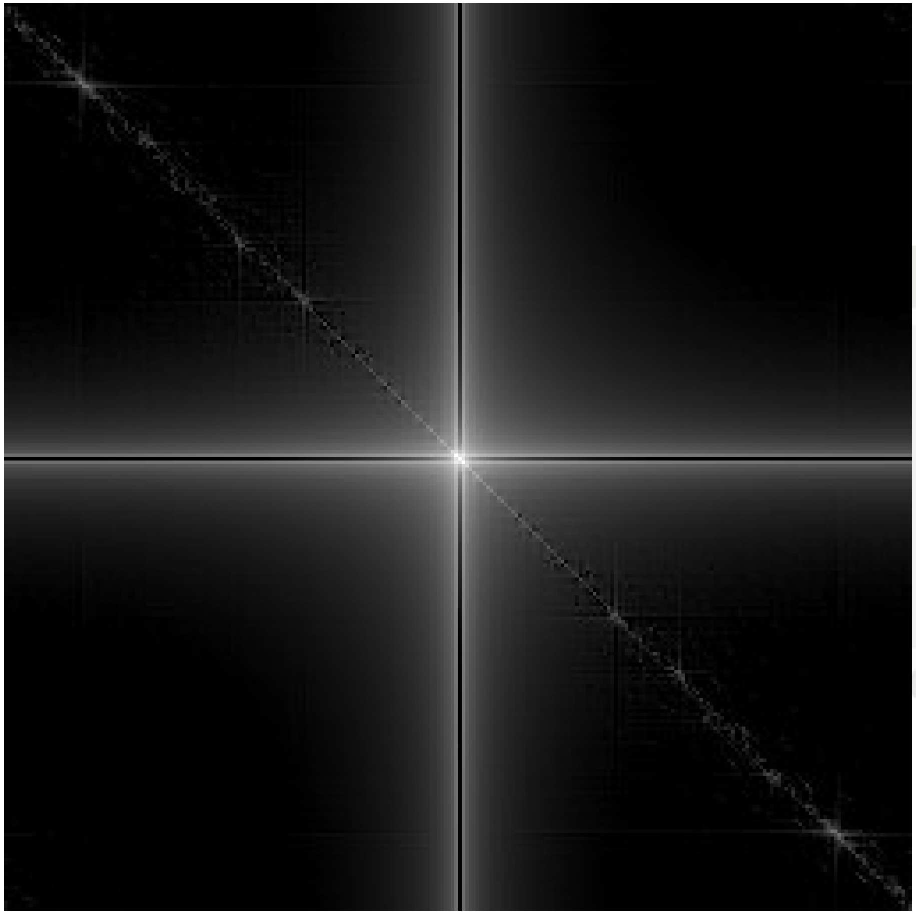
- Como?

Como resolver?

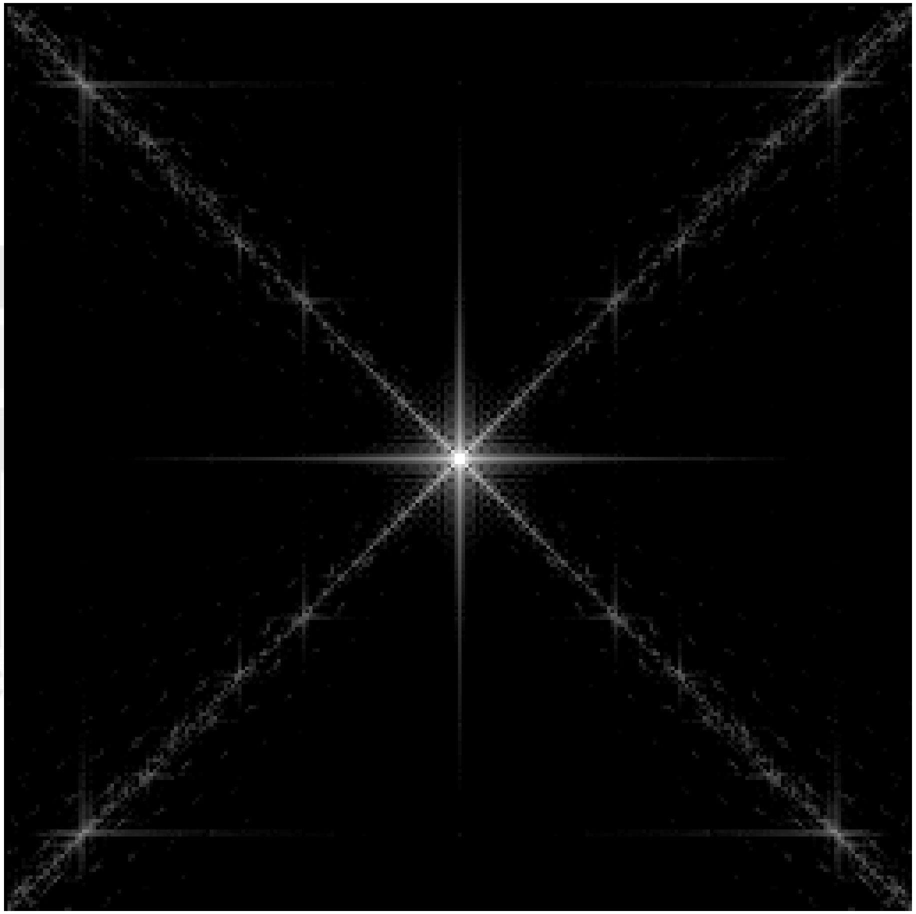
- Repetições espelhadas?



Como resolver?



Tiles



Repetições espelhadas

Windowing (janelamento?)

- Truque: escurecer as margens da imagem.
  - Mesmo considerando repetições simples, não teremos artefatos.
- Existem <u>várias</u> funções usadas na prática para *windowing*.
  - Vamos testar aqui duas funções.

Janela de Hann

$$w(x)=\frac{1-\cos\left(\frac{2\pi x}{W-1}\right)}{2}$$

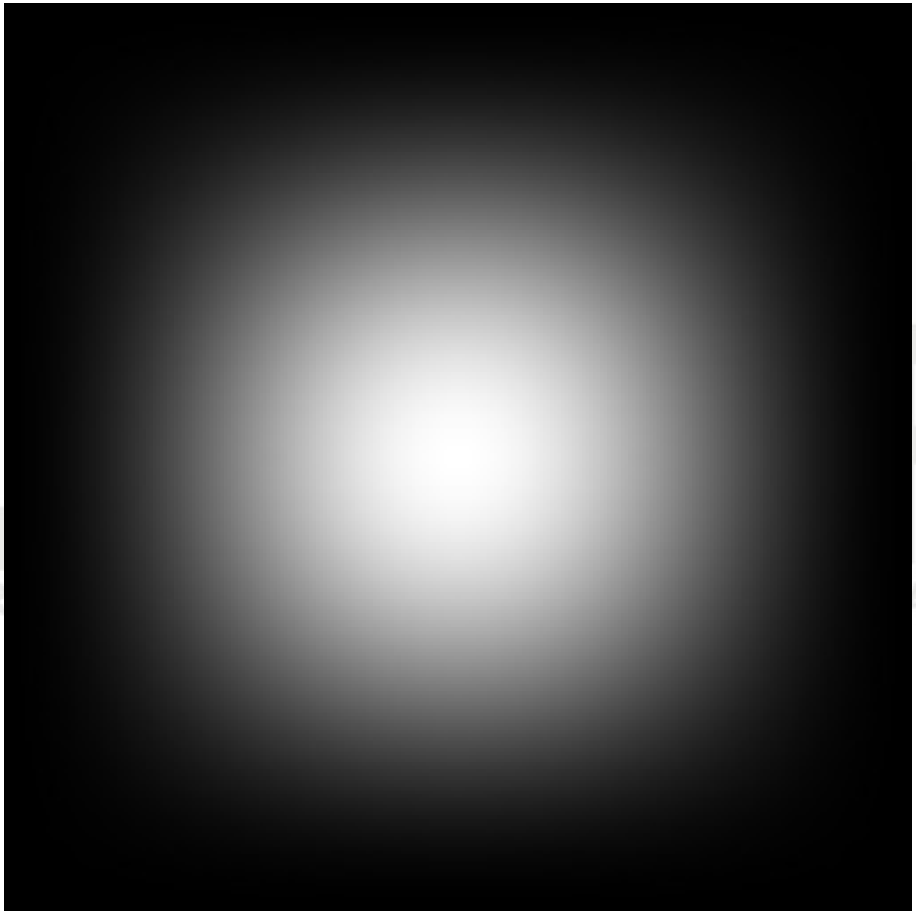
W é a largura da imagem...

•Para 2d, basta calcular *w(x)* * *w(y)*.

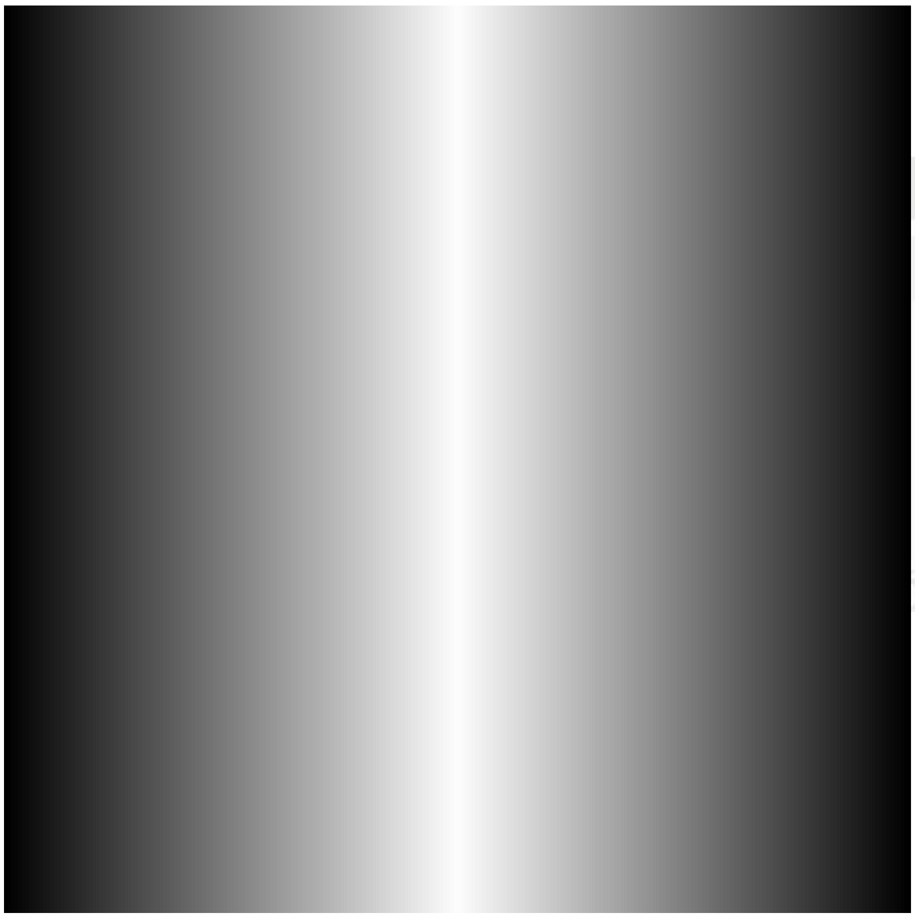
•Para aplicar a janela, basta multiplicar cada pixel por *w(x)* e *w(y):*

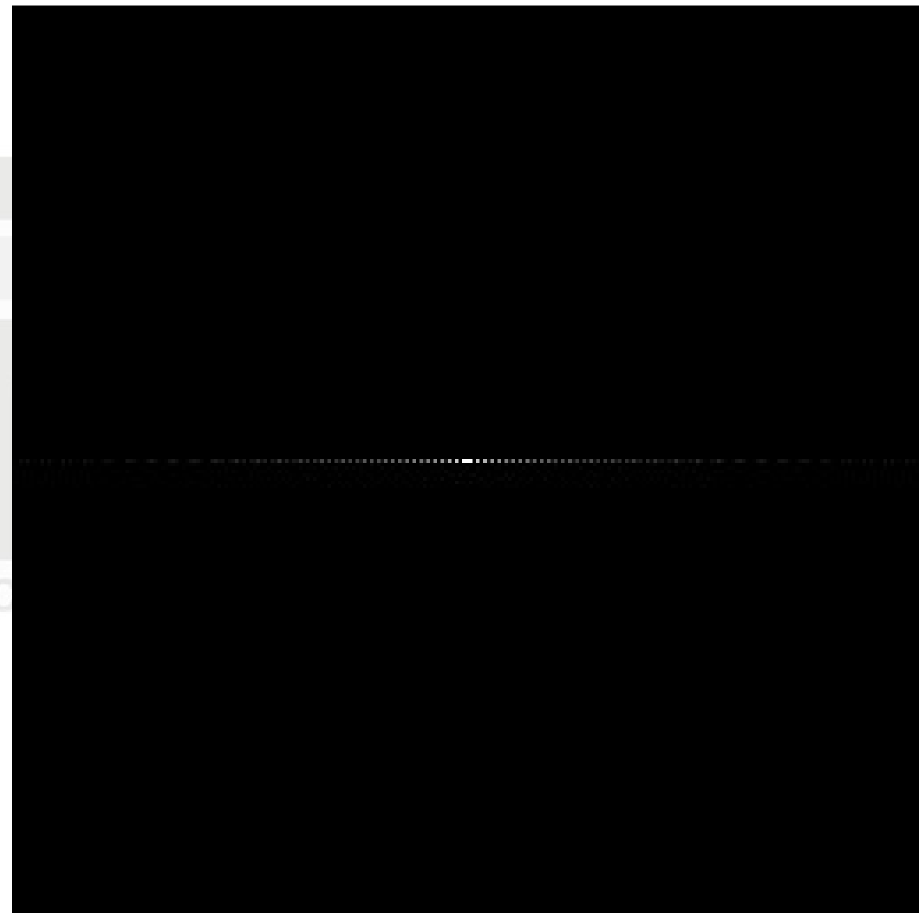
$$g(x,y)=f(x,y)\cdot w(x)\cdot w(y)$$

Janela de Hann

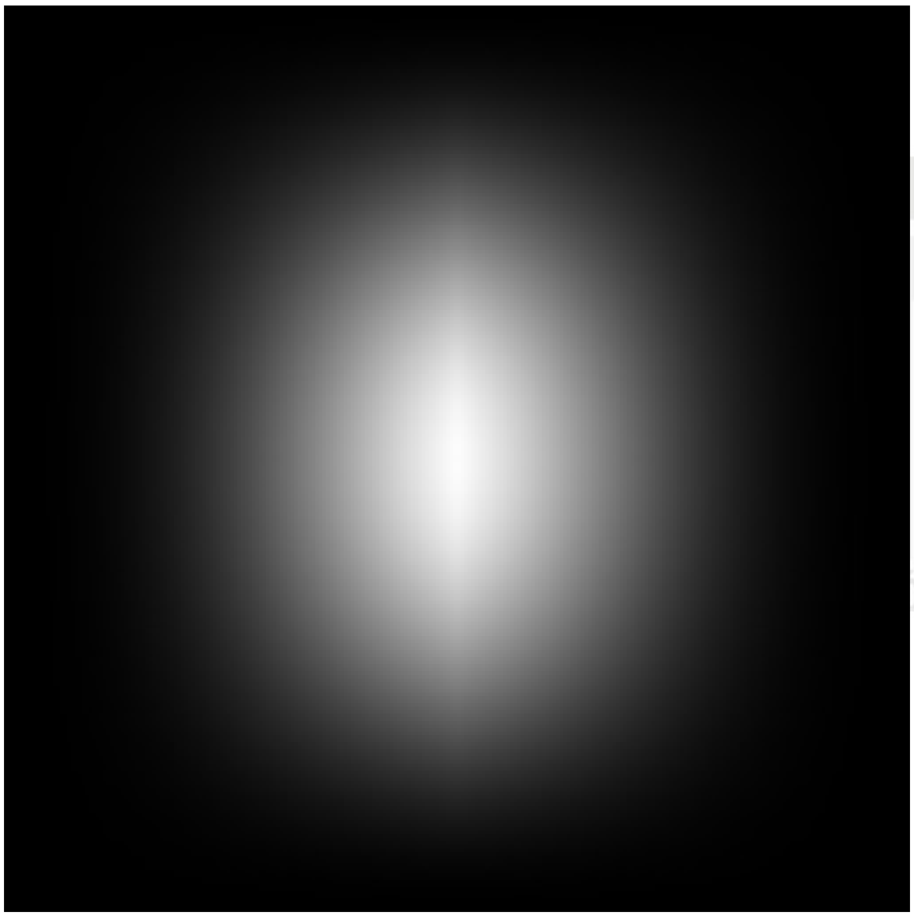


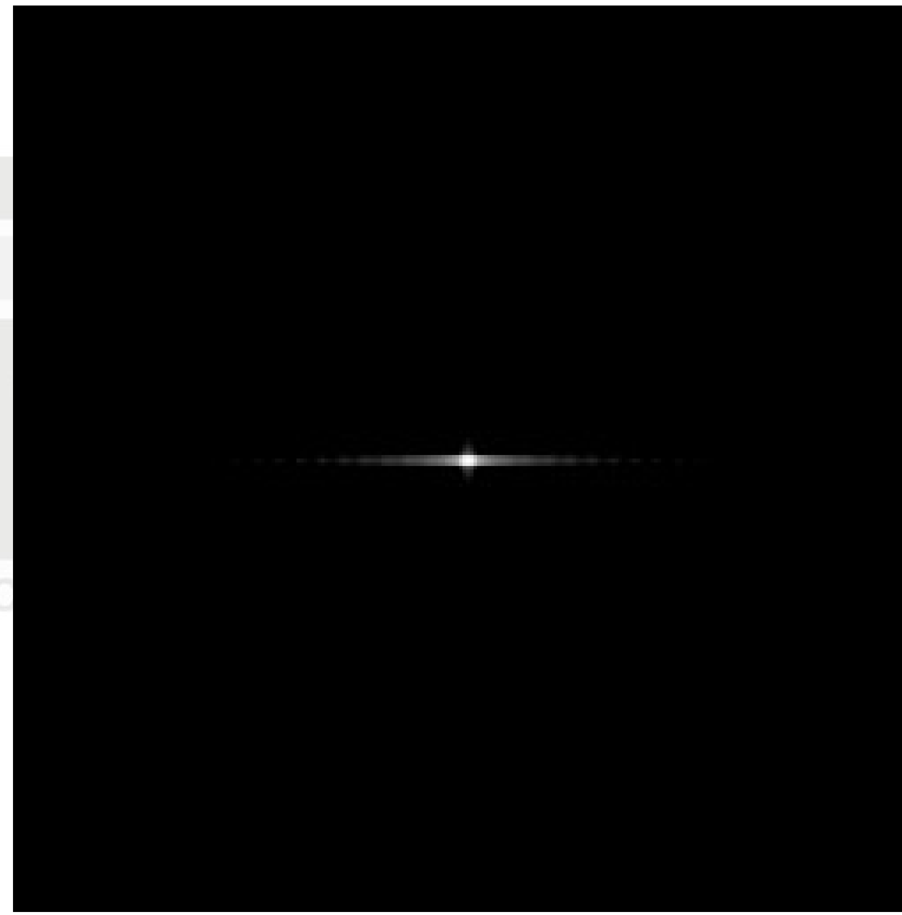
Janela de Hann (sem janela)



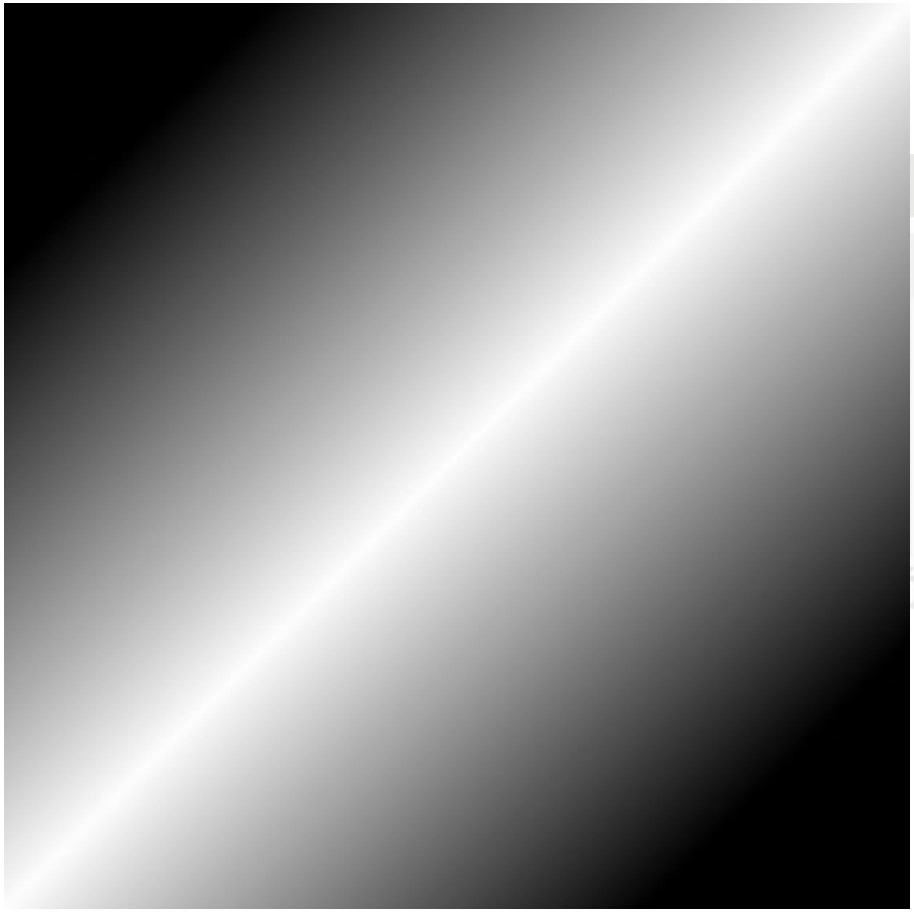


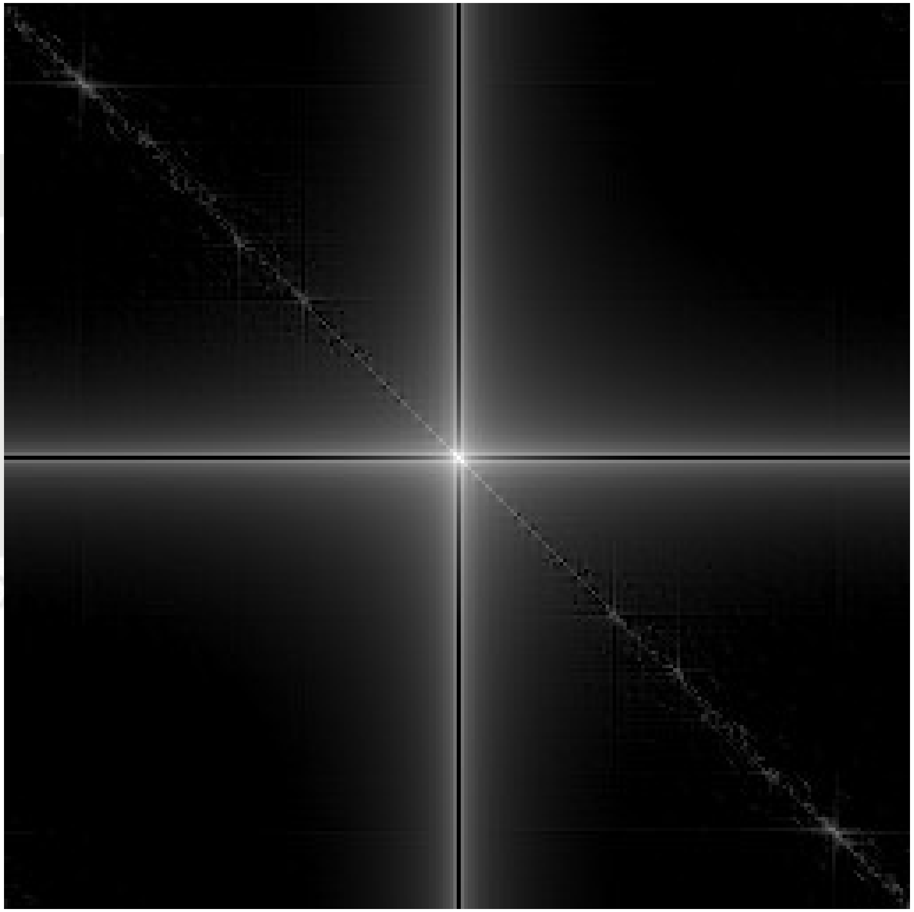
Janela de Hann (com janela)



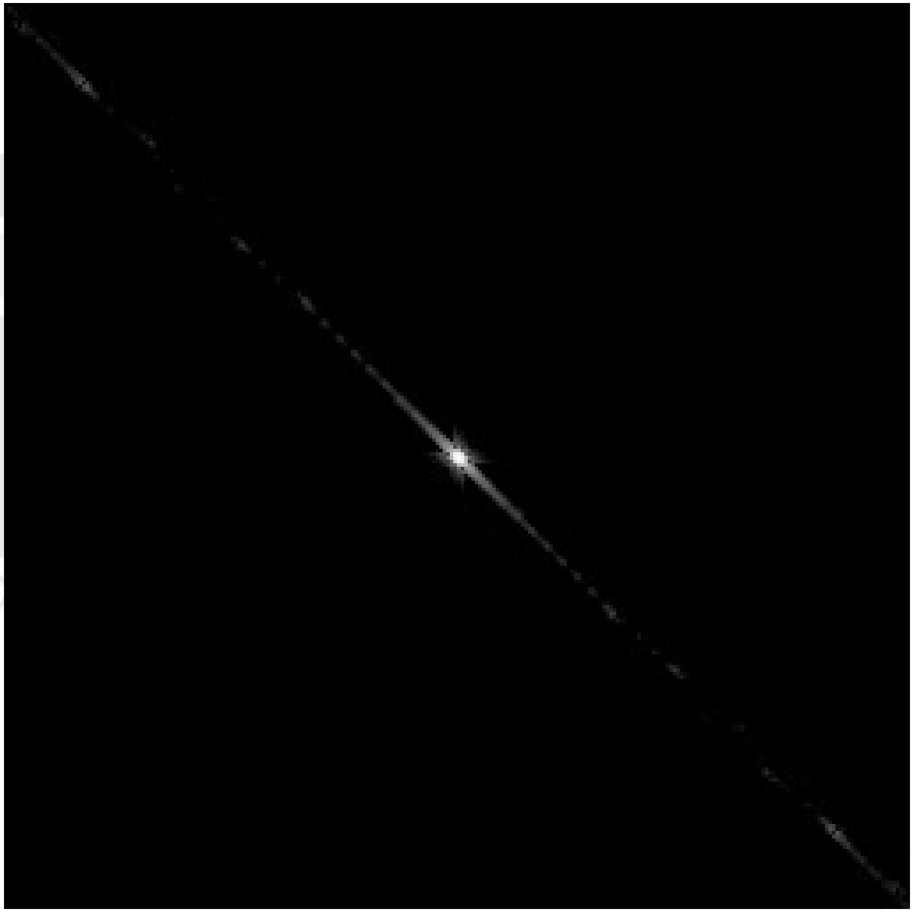


Janela de Hann (sem janela)





Janela de Hann (com janela)



UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Janela de Tukey

- Em funções como a janela de Hann, os dados têm sua importância reduzida conforme a distância para o centro aumenta.
  - Mesmo dados próximos ao centro são alterados.
- Existem alternativas que afetam somente as regiões próximas às margens.
  - Exemplo: janela de Tukey.

Janela de Tukey

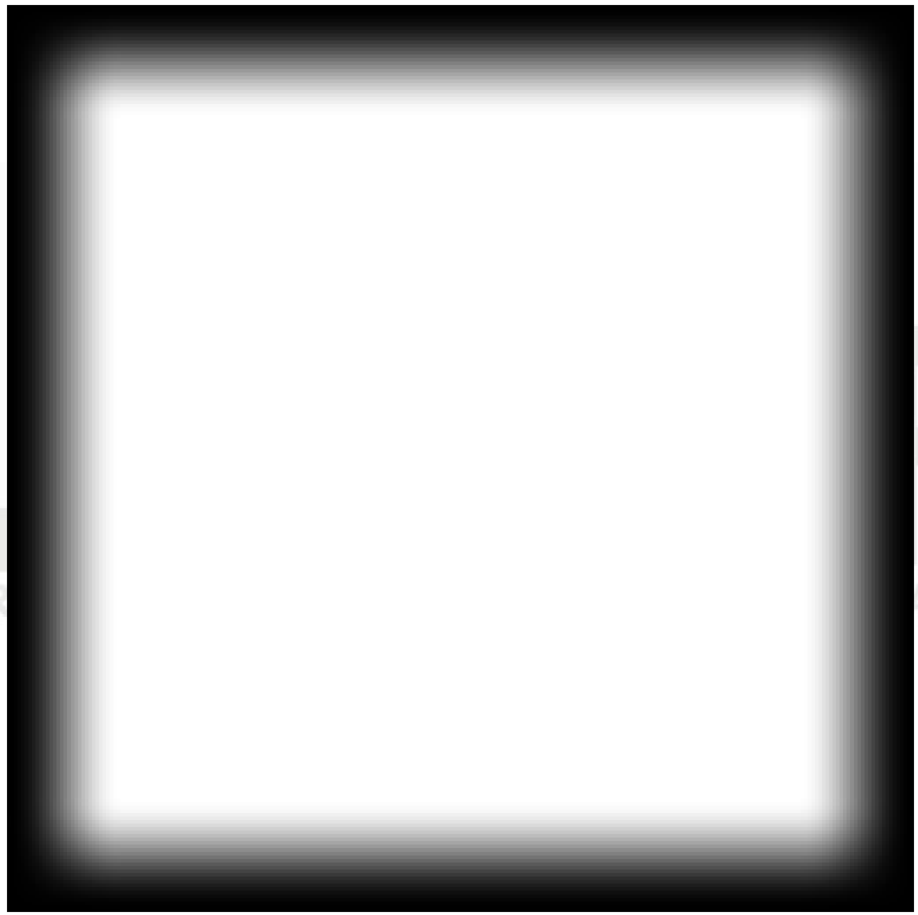
$$w(x)=\frac{1+\cos\left(\pi\left(\frac{2x}{\alpha(W-1)}-1\right)\right)}{2}\, se\, 0\leq x<\frac{\alpha(W-1)}{2}$$

$$w(x)=\frac{1+\cos\left(\pi\left(\frac{2x}{\alpha(W-1)}-\frac{2}{\alpha}+1\right)\right)}{2}\, se\,(W-1)(1-\frac{\alpha}{2})<x\leq(W-1)$$

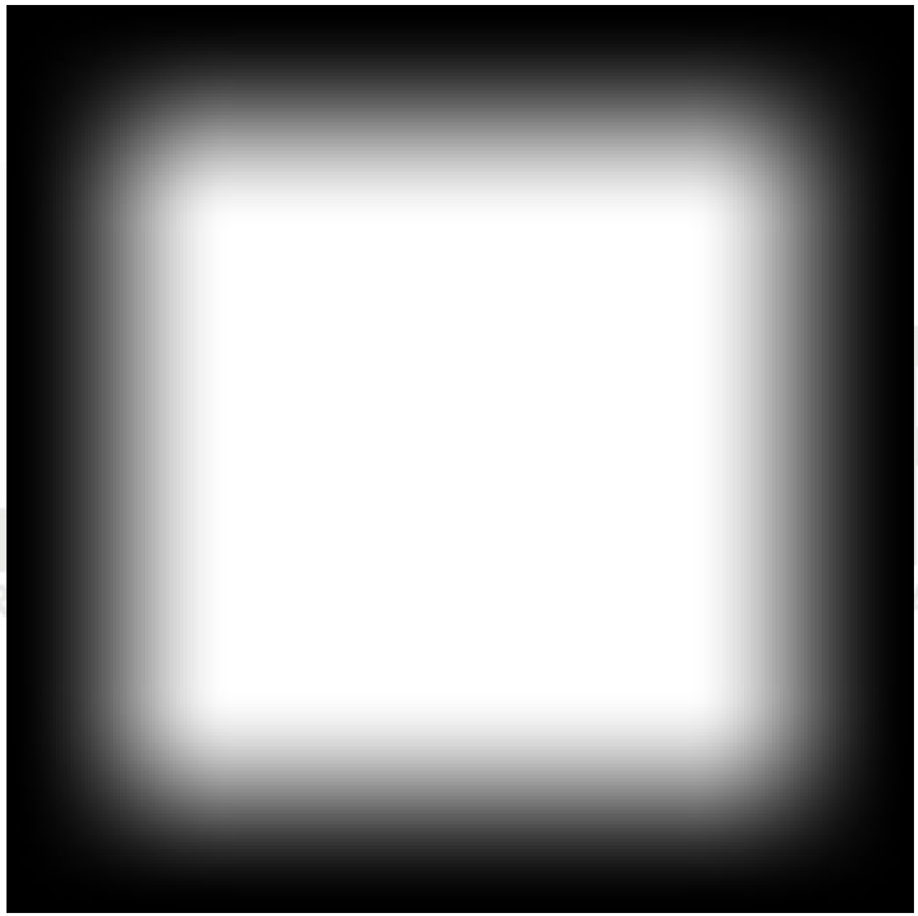
$$w(x)=1, do\, contrário$$

α é o parâmetro que diz o quanto a janela “entra” na imagem. Quanto maior, mais pixels são afetados.

Janela de Tukey (α = 0.25)

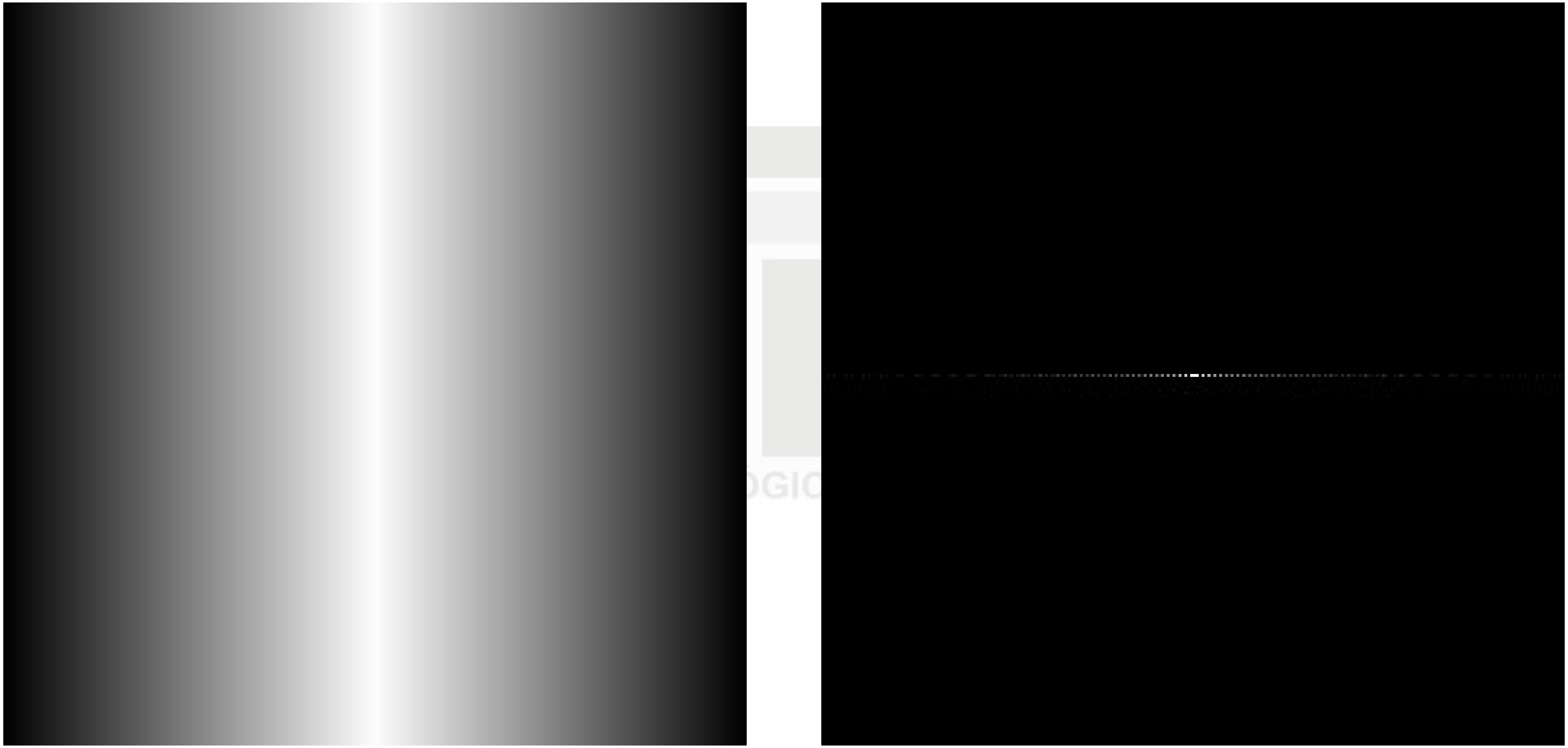


Janela de Tukey ($\alpha = 0.5$)

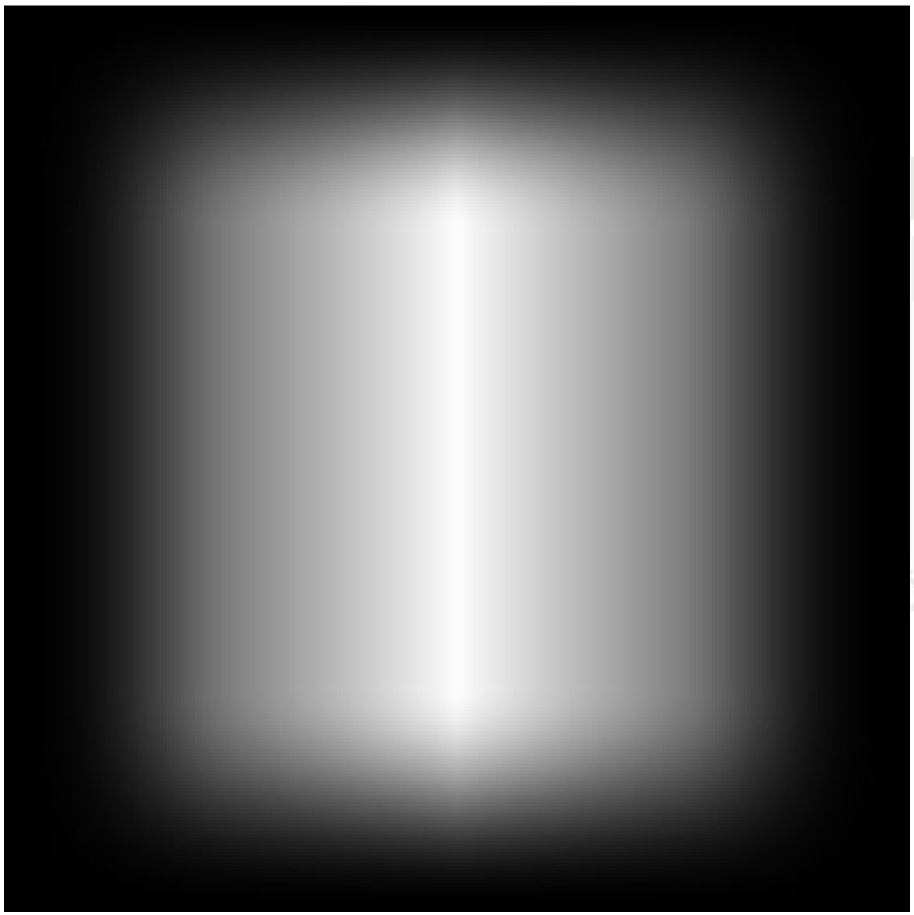


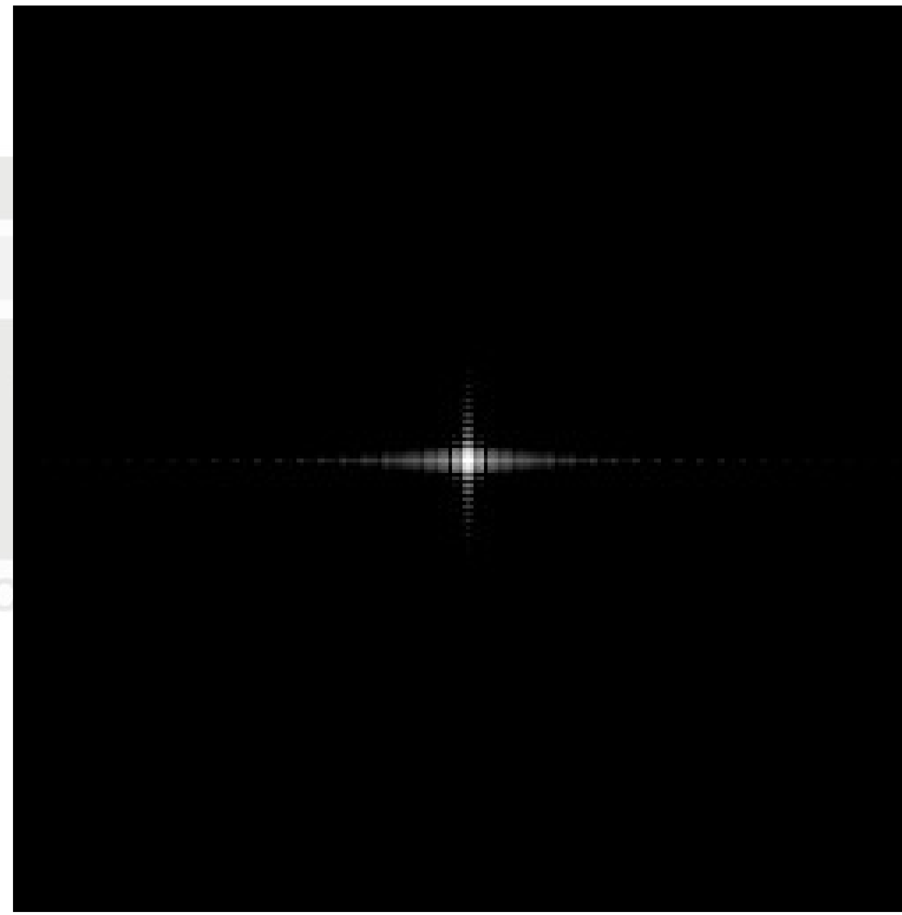
Janela de Tukey ($\alpha = 1$)

Janela de Tukey (sem janela)

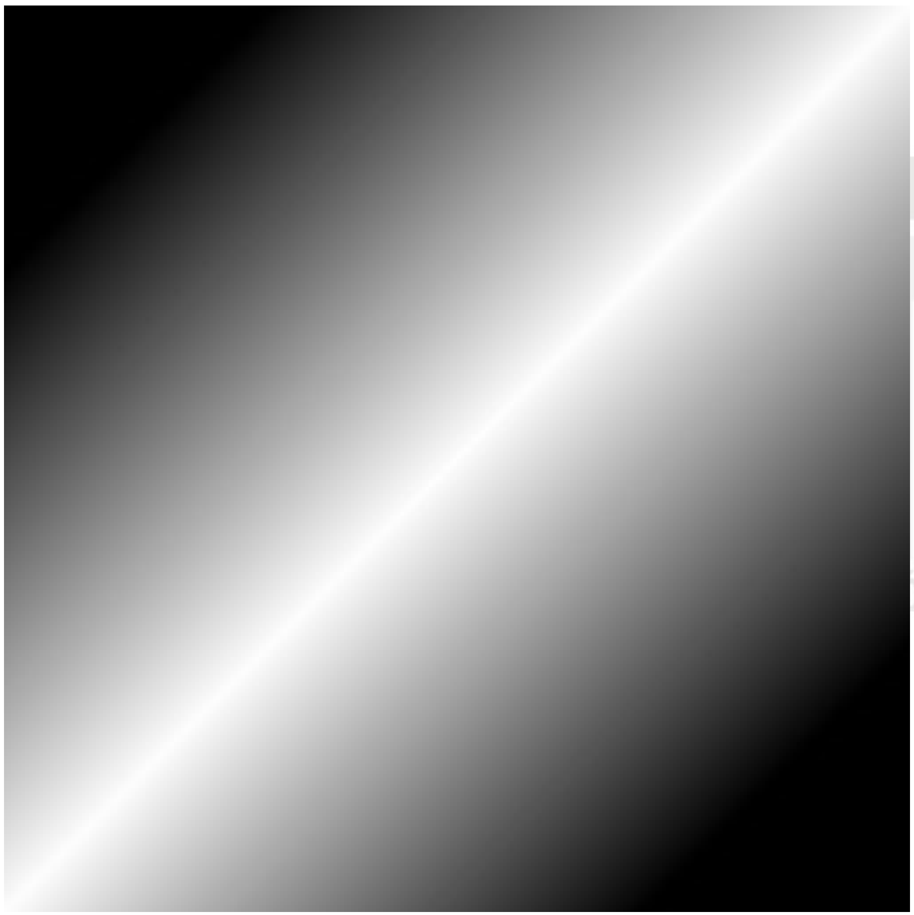


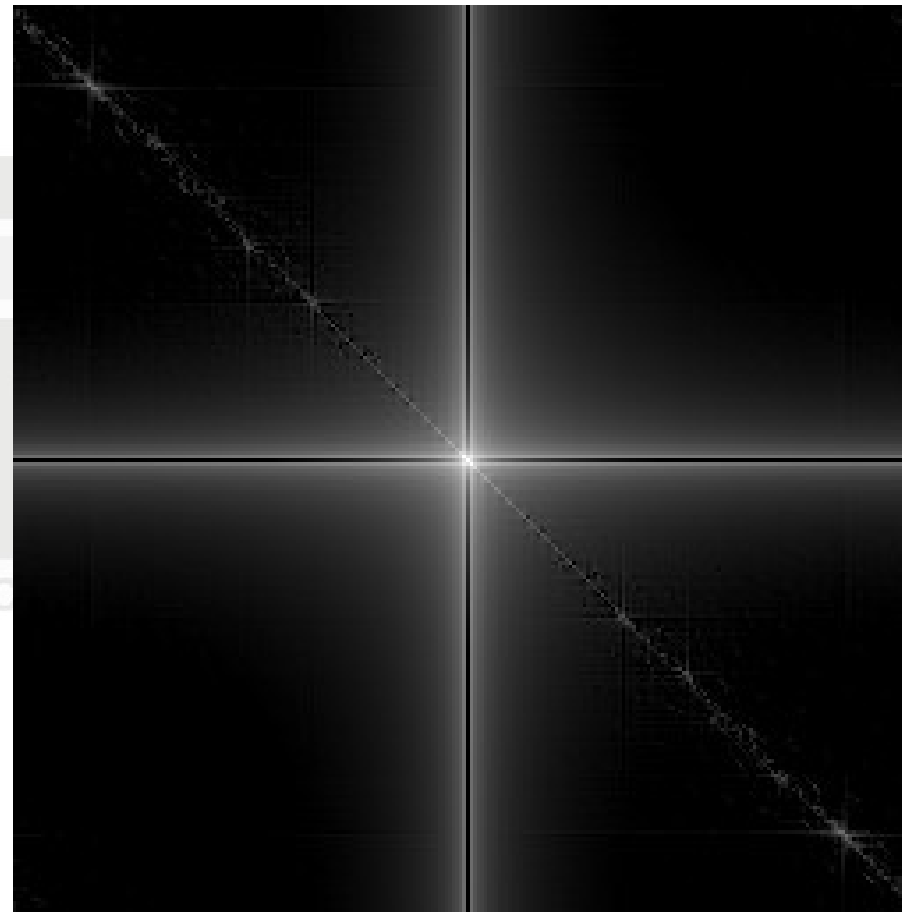
Janela de Tukey (com janela)



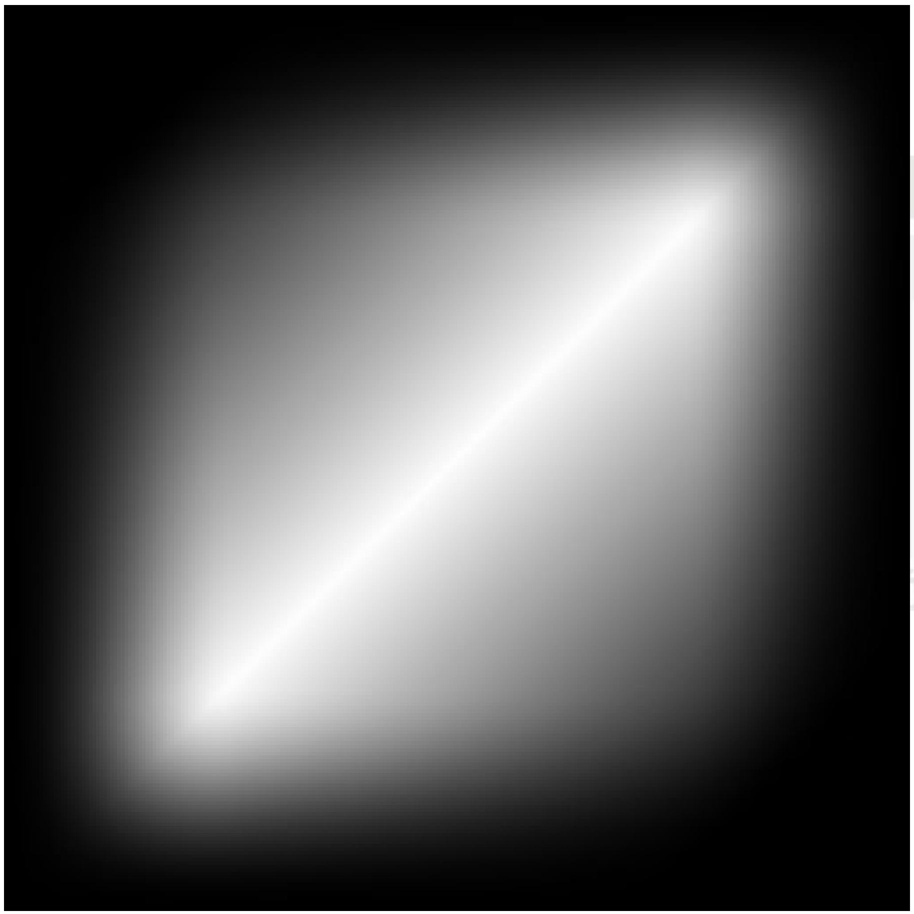


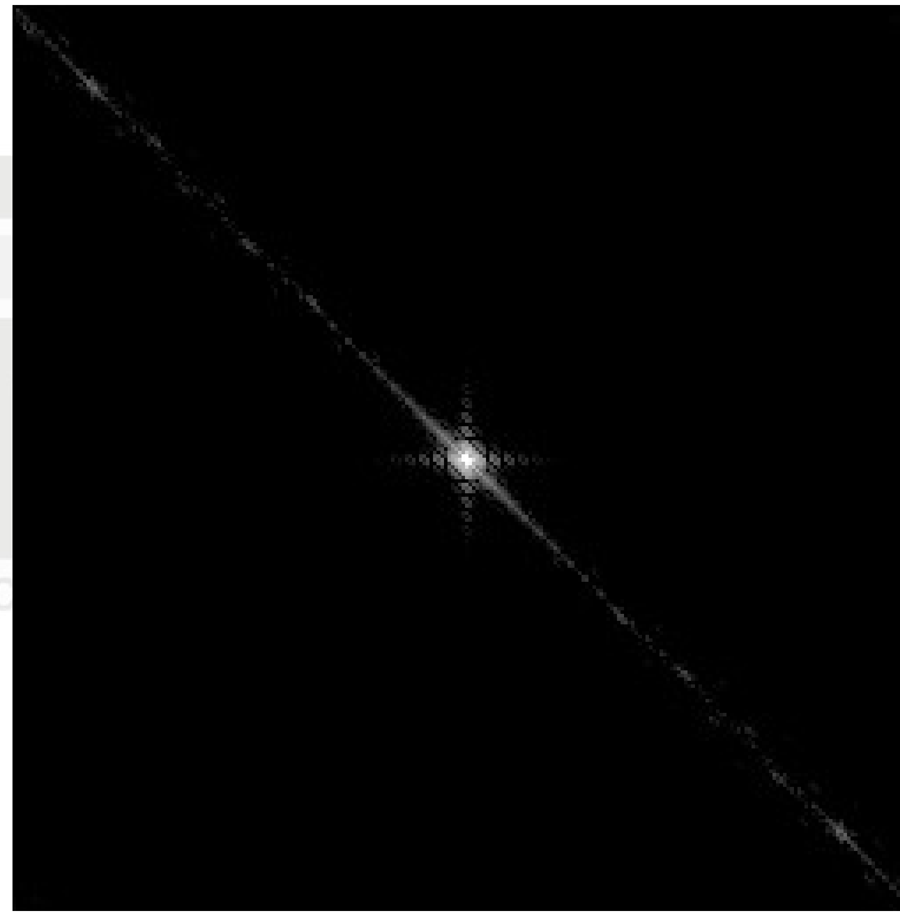
Janela de Tukey (sem janela)





Janela de Tukey (com janela)





Windowing: notas finais

- A aplicação de funções de janela implica em perda de informações.
  - Por outro lado, reduzimos o impacto de artefatos na análise.
- Quando usamos o domínio da frequência para analisar vizinhanças de uma imagem (e não imagens inteiras), é interessante que as vizinhanças se sobreponham.
  - “Janela deslizante” com passo (*stride*) menor que a largura da janela.
- Vejamos um exemplo simples...