



**OPPA European Social Fund
Prague & EU: We invest in your future.**

Frequency analysis in images

2D Fourier Transform

Revision: 1.9, dated: April 30, 2008

Tomáš Svoboda

Czech Technical University, Faculty of Electrical Engineering
Center for Machine Perception, Prague, Czech Republic

`svoboda@cmp.felk.cvut.cz`

`http://cmp.felk.cvut.cz/~svoboda`

Frequency Analysis

Decomposition of signals into sequences of sinusoids \cos, \sin .

Why?

Frequency Analysis

Decomposition of signals into sequences of sinusoids \cos , \sin .

Why?

- ◆ Speech, music analysis.

Frequency Analysis

Decomposition of signals into sequences of sinusoids \cos , \sin .

Why?

- ◆ Speech, music analysis.
- ◆ **Filtering**: boosting or attenuating of specific frequencies.

Frequency Analysis

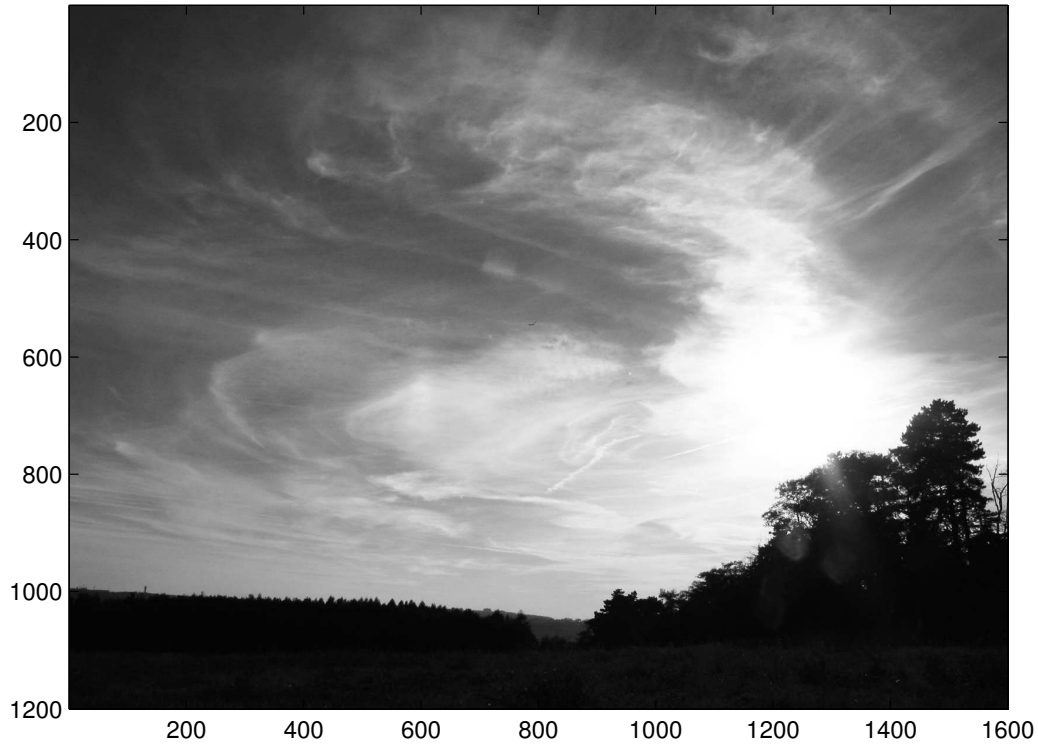
Decomposition of signals into sequences of sinusoids \cos , \sin .

Why?

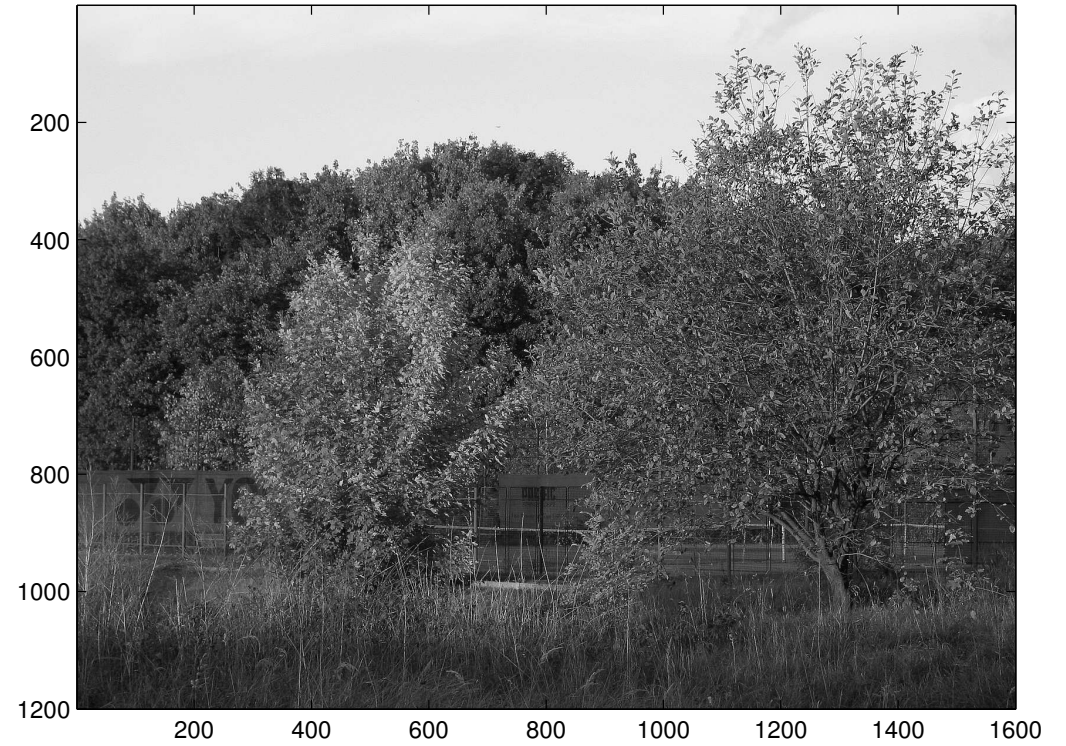
- ◆ Speech, music analysis.
- ◆ **Filtering**: boosting or attenuating of specific frequencies.
- ◆ Some image distortions may be characterized well in frequency domain.

What are frequencies in real images

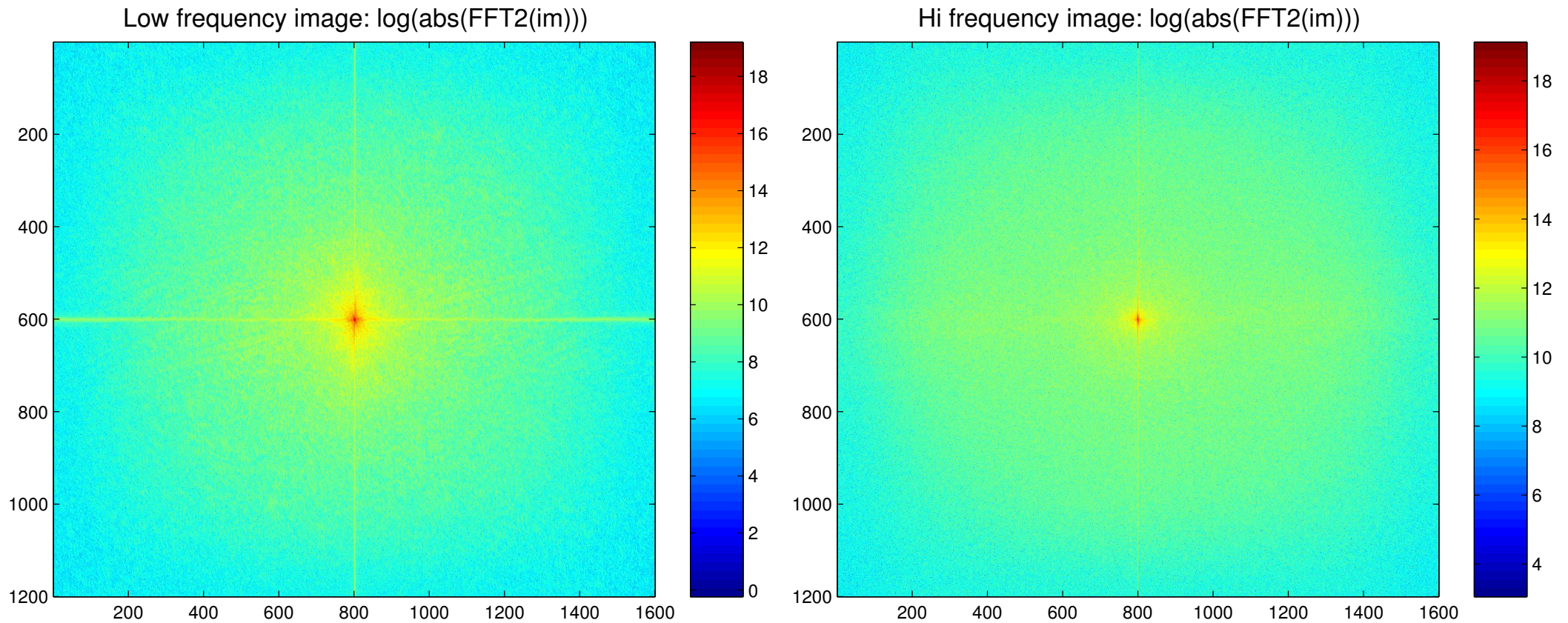
Low frequency image in gray scales



Hi frequency image in gray scales



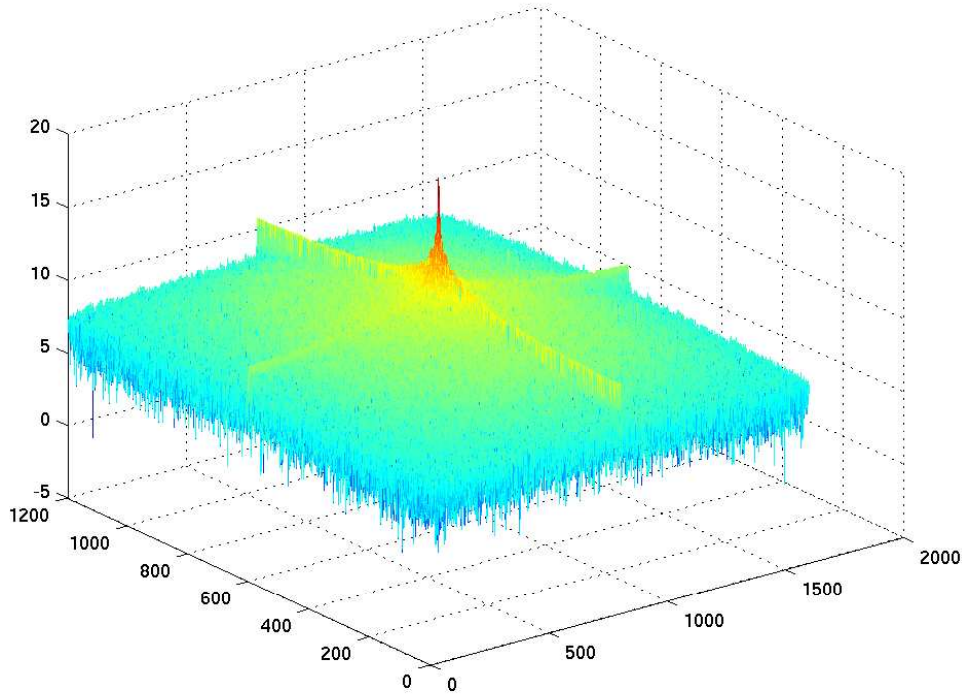
What are frequencies in real images



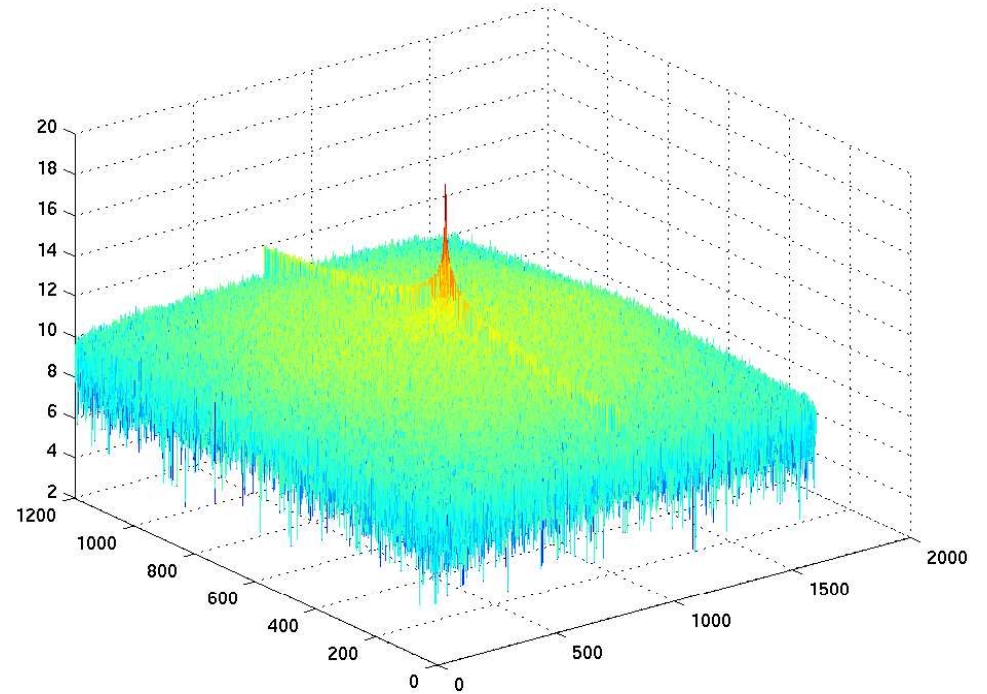
The further from image center, $[600, 800]$, the higher frequency. Do not care about details we will come to this later.

What are frequencies in real images

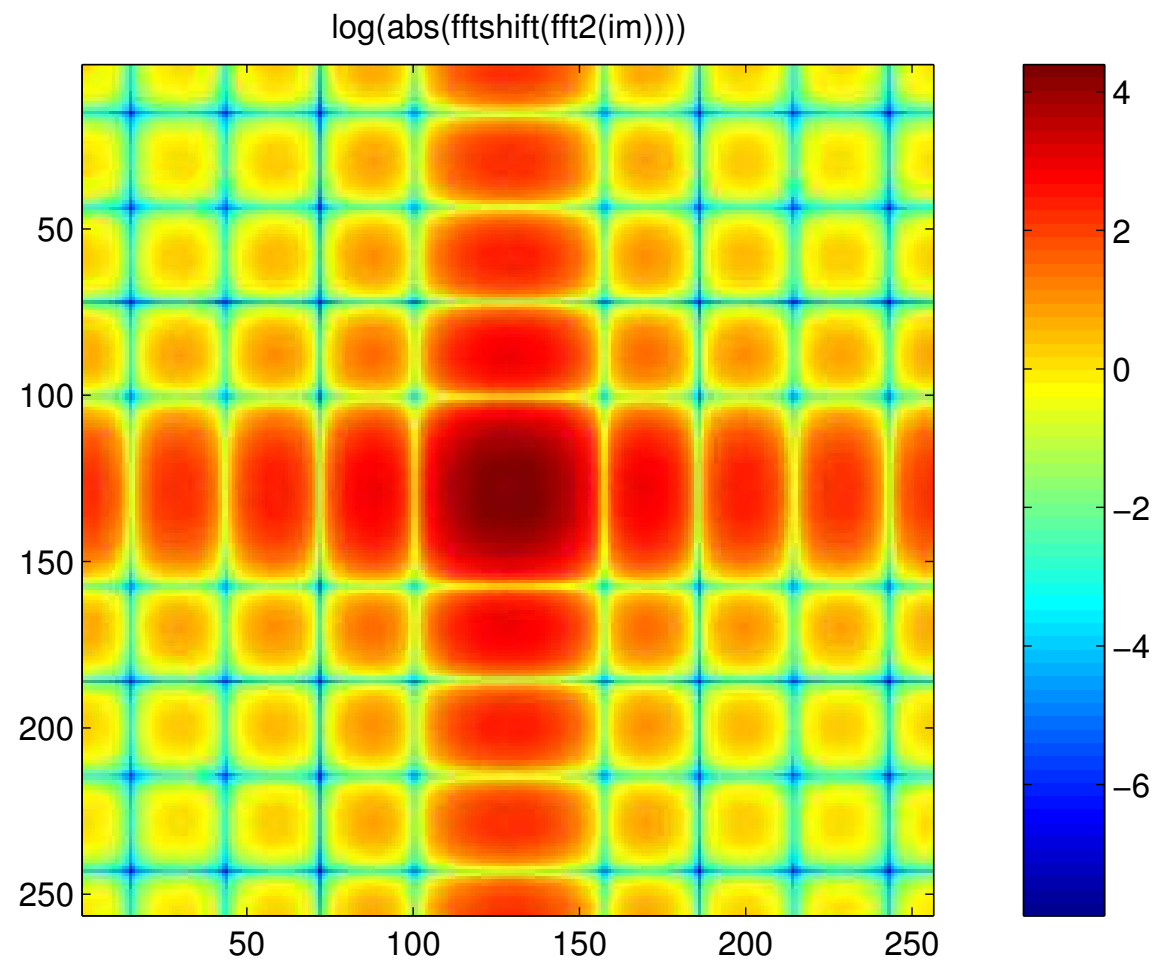
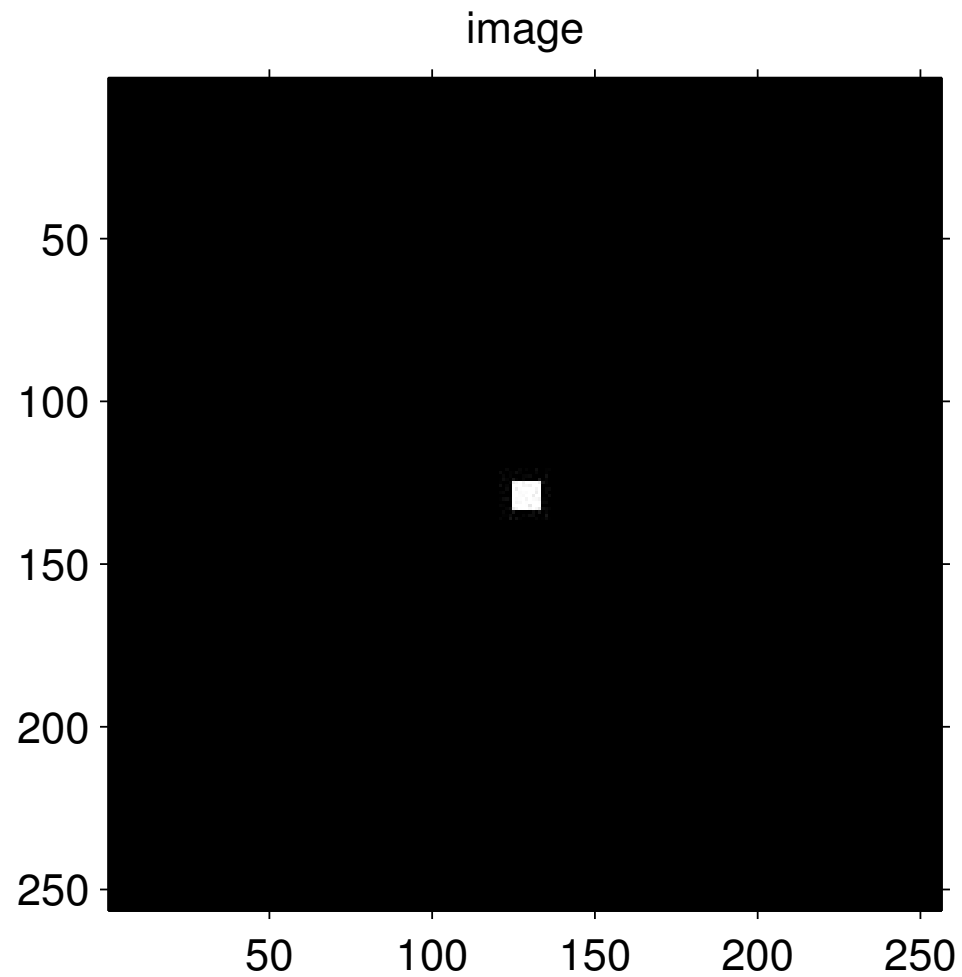
Low frequency image: mesh print of $\log(\text{abs}(\text{FFT2}(\text{im})))$



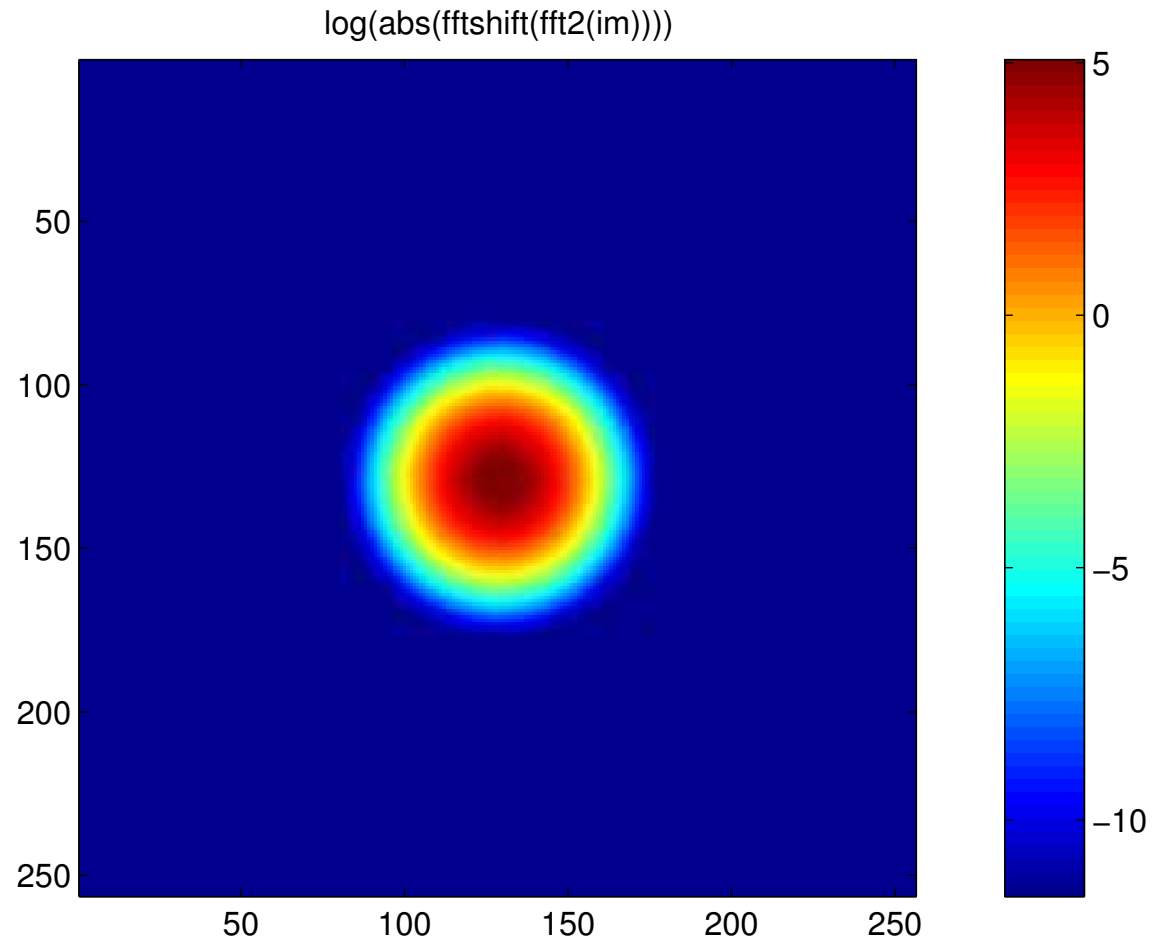
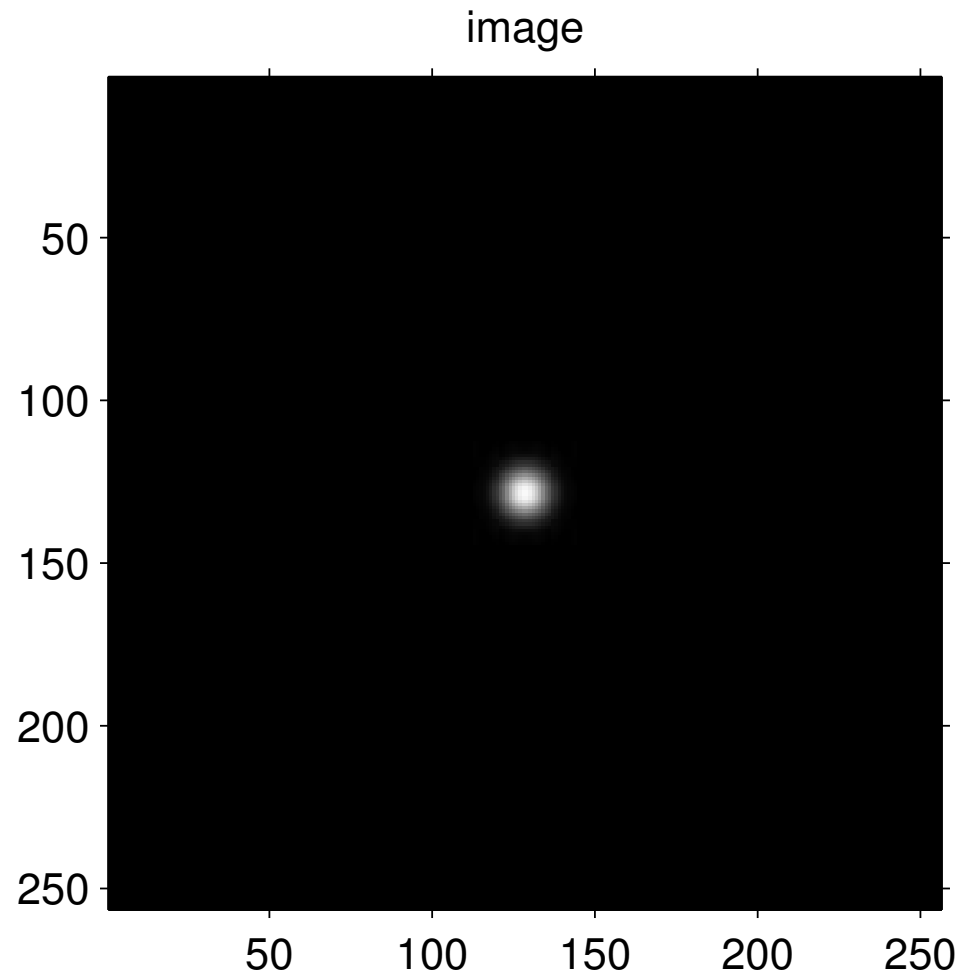
Hi frequency image: mesh print of $\log(\text{abs}(\text{FFT2}(\text{im})))$



What are frequencies in artificial images



What are frequencies in artificial images



Fourier transform — Decomposition into sinusoids



Mathematical procedure for computing how is each frequency u in the input signal $f(x)$.

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi ux} dx$$

Fourier transform — Decomposition into sinusoids

Mathematical procedure for computing how is each frequency u in the input signal $f(x)$.

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

and back to image/time space

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi xu} du$$

Fourier transform — Decomposition into sinusoids

Mathematical procedure for computing how is each frequency u in the input signal $f(x)$.

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

and back to image/time space

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi xu} du$$

do not forget that

$$\begin{aligned} e^{-i2\pi ux} &= \cos(2\pi ux) - i \sin(2\pi ux), \\ e^{i2\pi ux} &= \cos(2\pi ux) + i \sin(2\pi ux). \end{aligned}$$

2D Fourier transform

Straightforward generalization of the 1D transform

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(xu+yv)} du dv$$

Remark: f has a fourier image F if it is integrable, i.e. has finite energy.

2D FT — Corrugation viewpoint

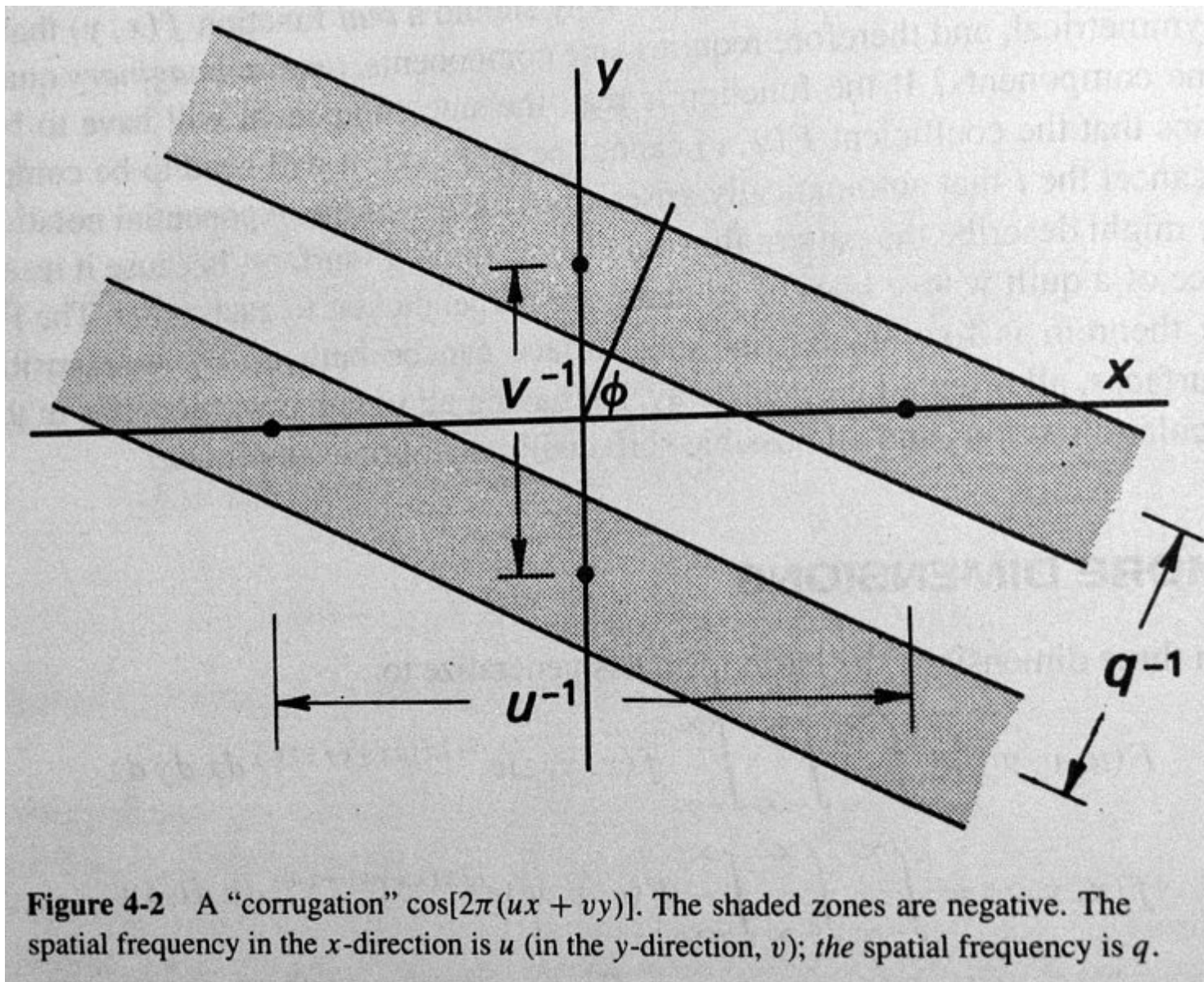


Illustration taken from [1].

2D FT — Corrugation viewpoint

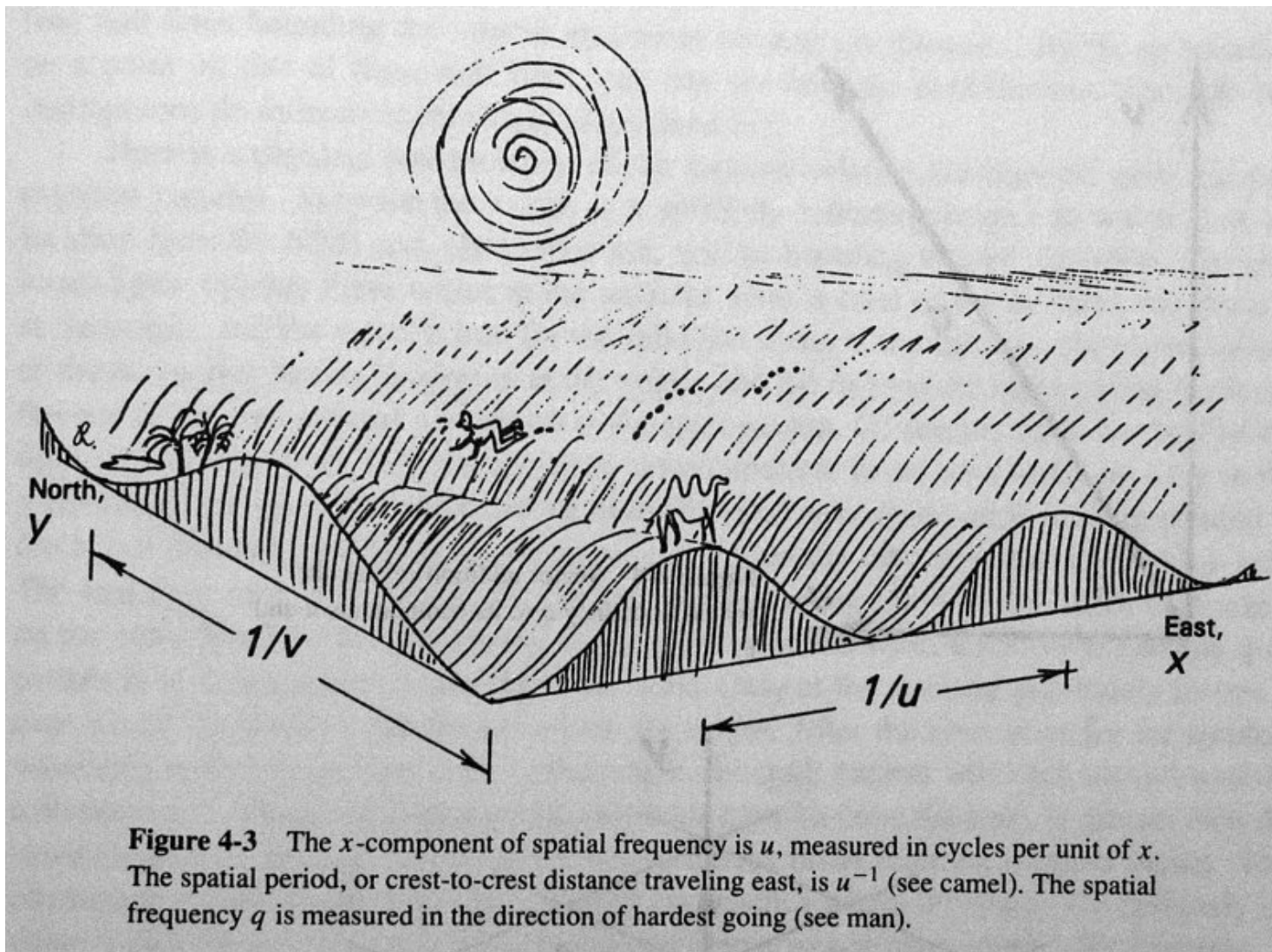


Illustration taken from [1].

Discrete Fourier transform

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left(-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right)$$

Discrete Fourier transform

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left(-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left(i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right)$$

Discrete Fourier transform

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left(-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left(i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right)$$

Discrete and finite range of $f(x, y)$, and $F(u, v)$ has some important consequences . . .

Discrete Fourier transform — DC part

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp\left(-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right)$$

x is an abbreviation for $x_0 + x\delta_x$.

u is $u_0 + u\delta_u$, where $\delta_u = \frac{1}{M\delta_x}$.

Discrete Fourier transform — DC part

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp\left(-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right)$$

x is an abbreviation for $x_0 + x\delta_x$.

u is $u_0 + u\delta_u$, where $\delta_u = \frac{1}{M\delta_x}$.

DC part:

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

which is the **mean** of f . But beware of actual implementation! Sometimes it may be only proportional to the mean.

Remember the DC part, we will need it to understand some filtering results.

Discrete Fourier transform — Periodicity

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \exp\left(-i2\pi\left(\frac{ux}{M}\right)\right)$$

x is an abbreviation for $x_0 + x\delta_x$. Hence, $x = 0, 1, \dots, M - 1$

u is $u_0 + u\delta_u$, where $\delta_u = \frac{1}{M\delta_x}$.

Discrete Fourier transform — Periodicity

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \exp\left(-i2\pi\left(\frac{ux}{M}\right)\right)$$

x is an abbreviation for $x_0 + x\delta_x$. Hence, $x = 0, 1, \dots, M - 1$

u is $u_0 + u\delta_u$, where $\delta_u = \frac{1}{M\delta_x}$.

Periodicity:

We know that: $\exp\left(-i2\pi\left(\frac{ux}{M}\right)\right) = \cos(2\pi ux/M) - i \sin(2\pi ux/M)$.

Discrete Fourier transform — Periodicity

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \exp\left(-i2\pi\left(\frac{ux}{M}\right)\right)$$

x is an abbreviation for $x_0 + x\delta_x$. Hence, $x = 0, 1, \dots, M - 1$

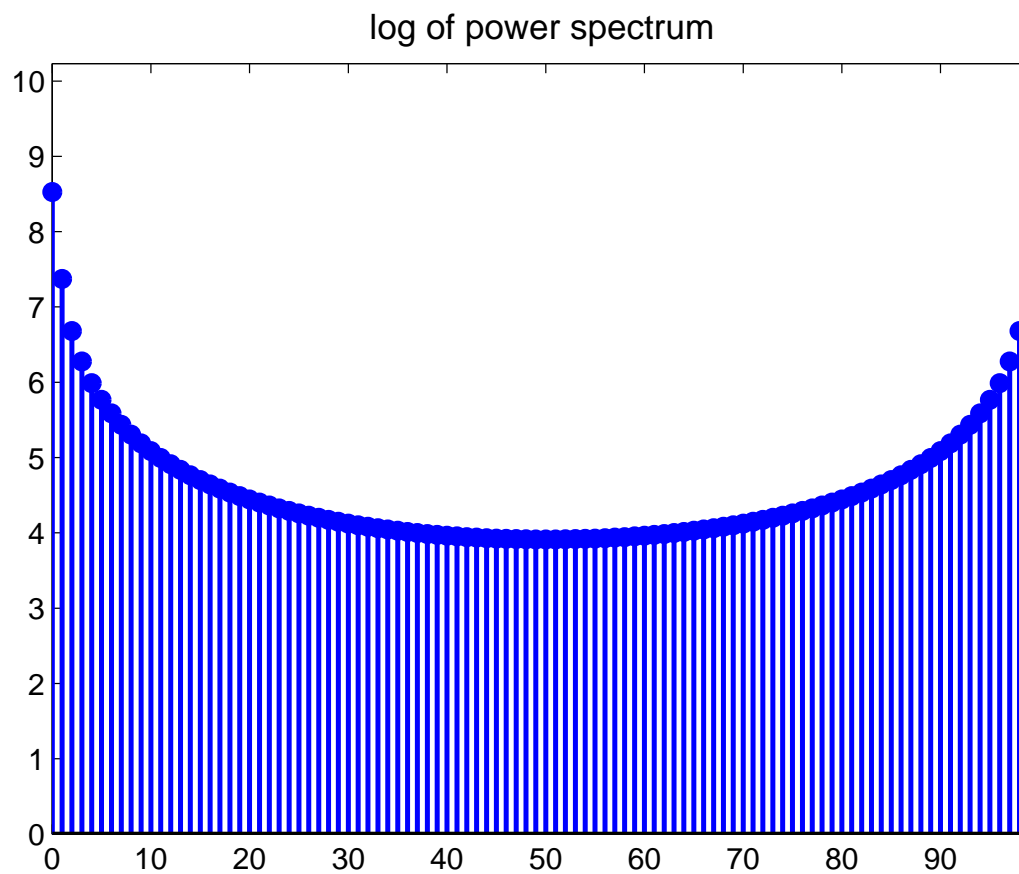
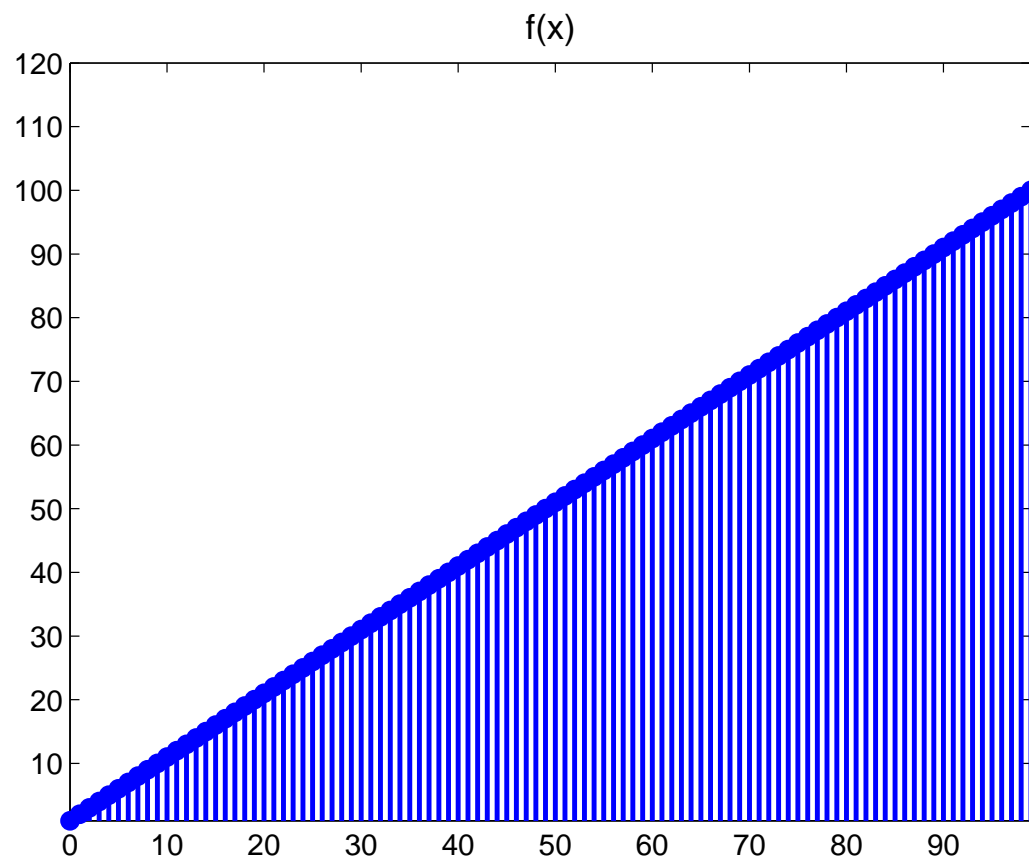
u is $u_0 + u\delta_u$, where $\delta_u = \frac{1}{M\delta_x}$.

Periodicity:

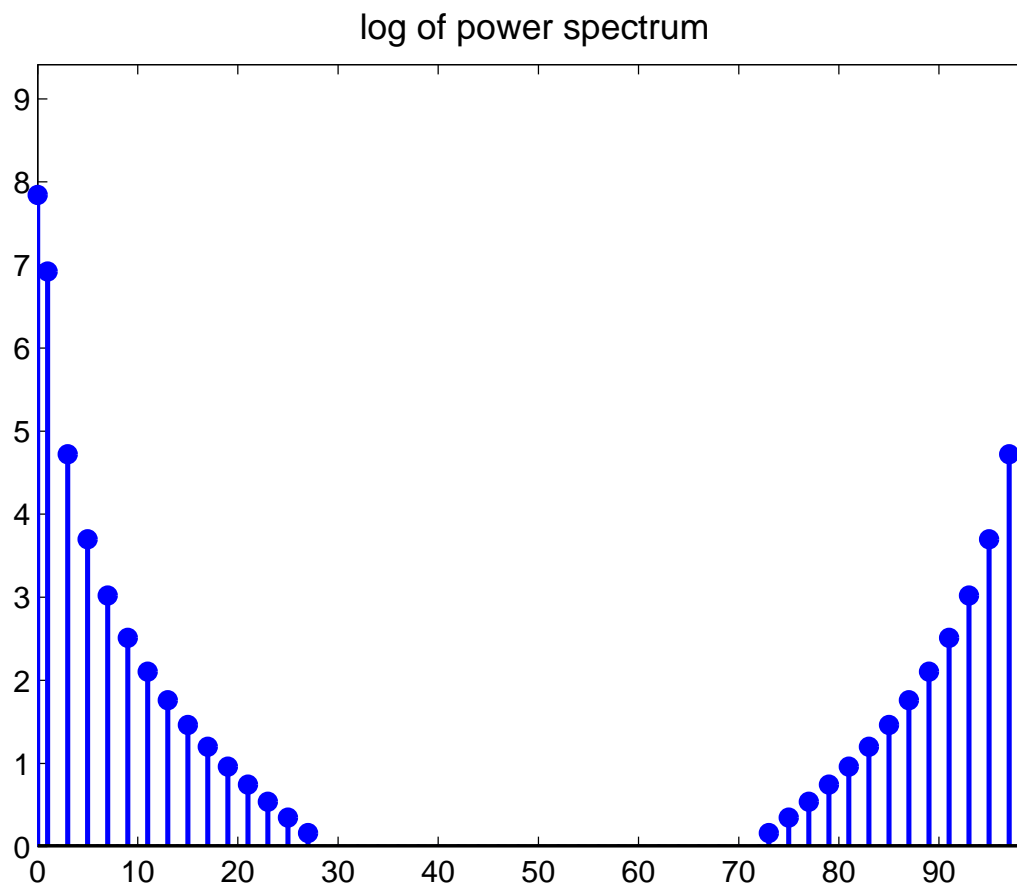
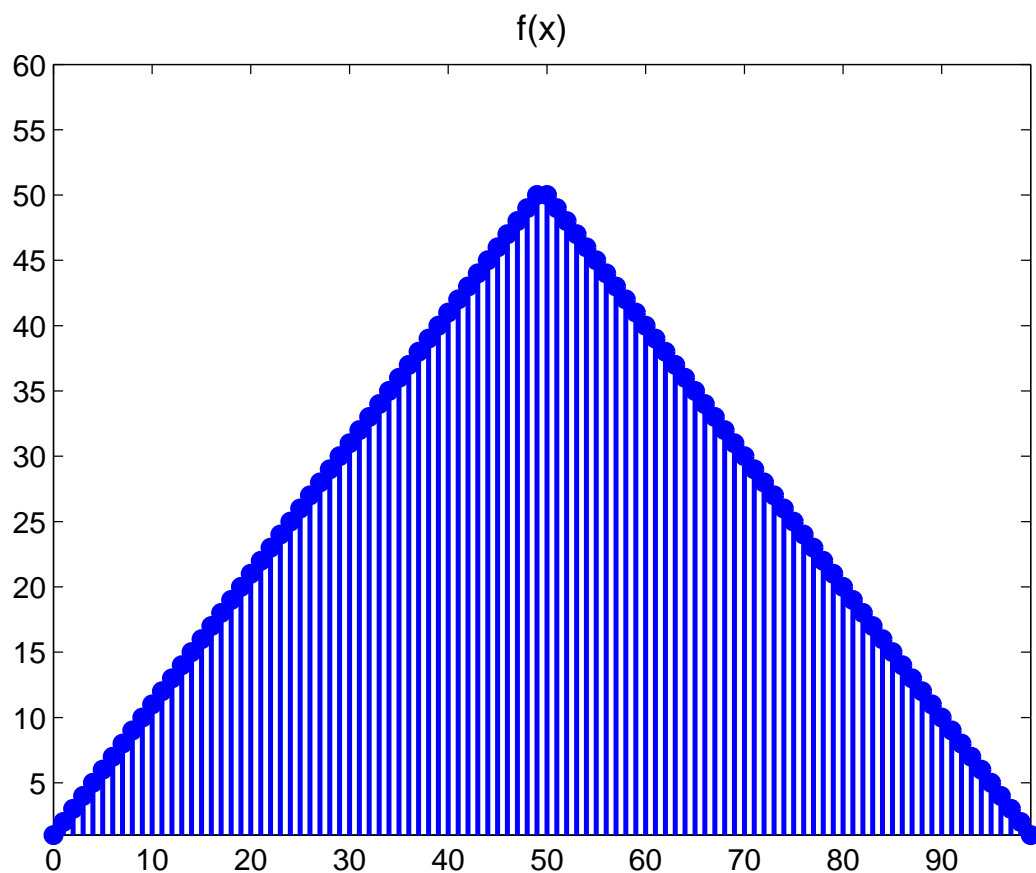
We know that: $\exp\left(-i2\pi\left(\frac{ux}{M}\right)\right) = \cos(2\pi ux/M) - i \sin(2\pi ux/M)$.

It means that $F(u)$ is periodic with the period M .

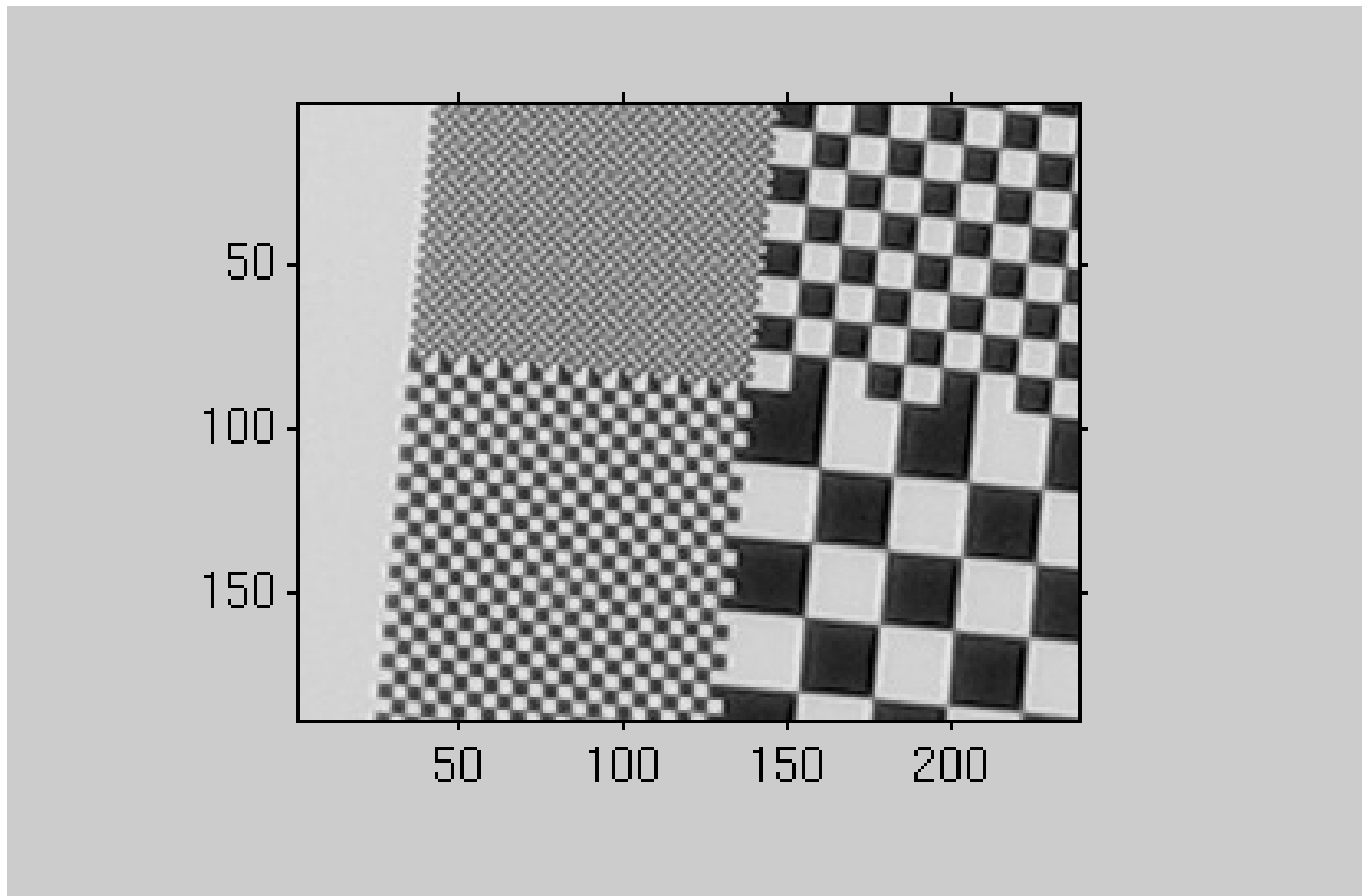
Periodicity of Fourier Transform I



Periodicity of Fourier Transform II

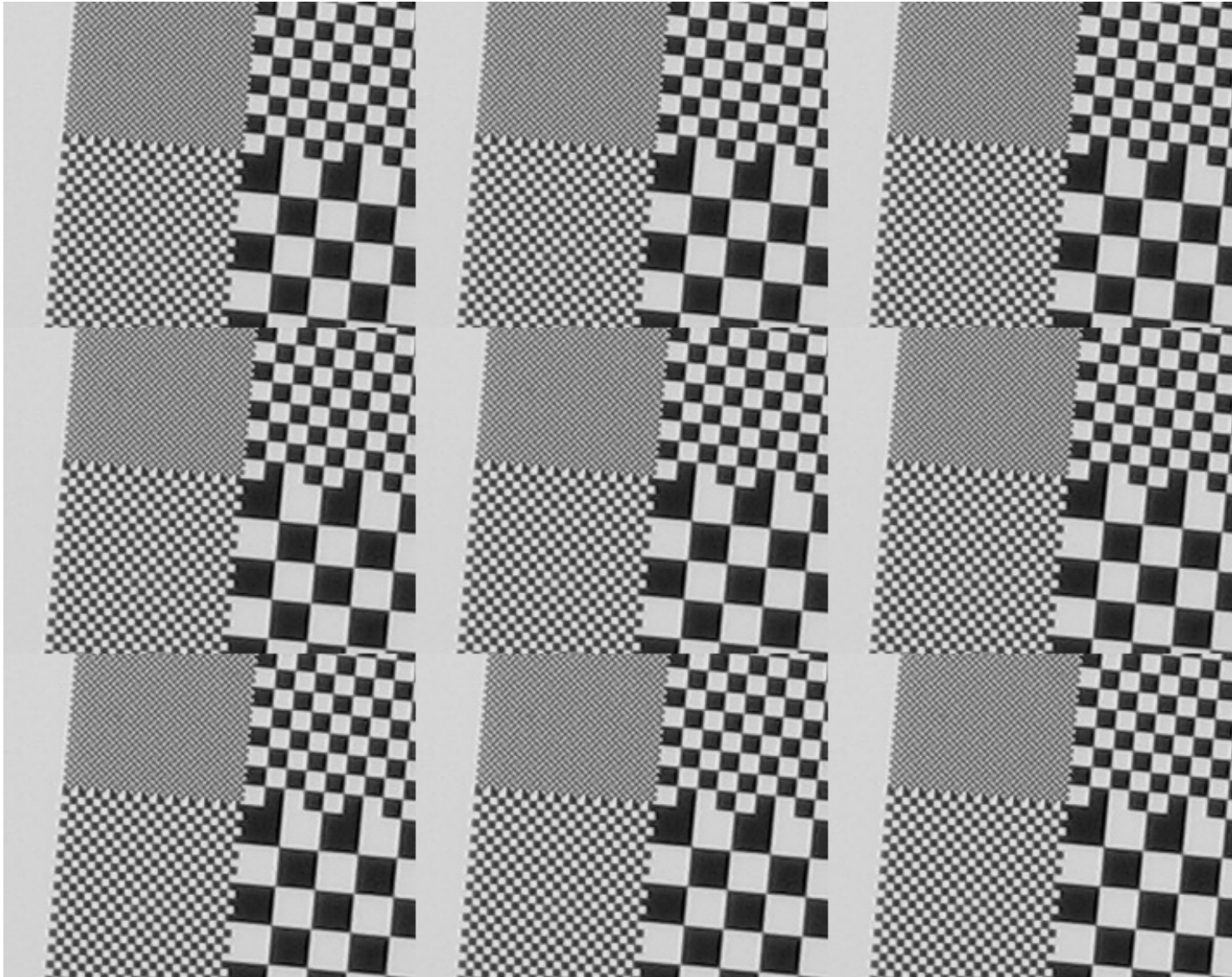


Periodicity of Fourier Transform in Images



Input image. However, DFT assumes . . .

Periodicity of Fourier Transform in Images – cont.

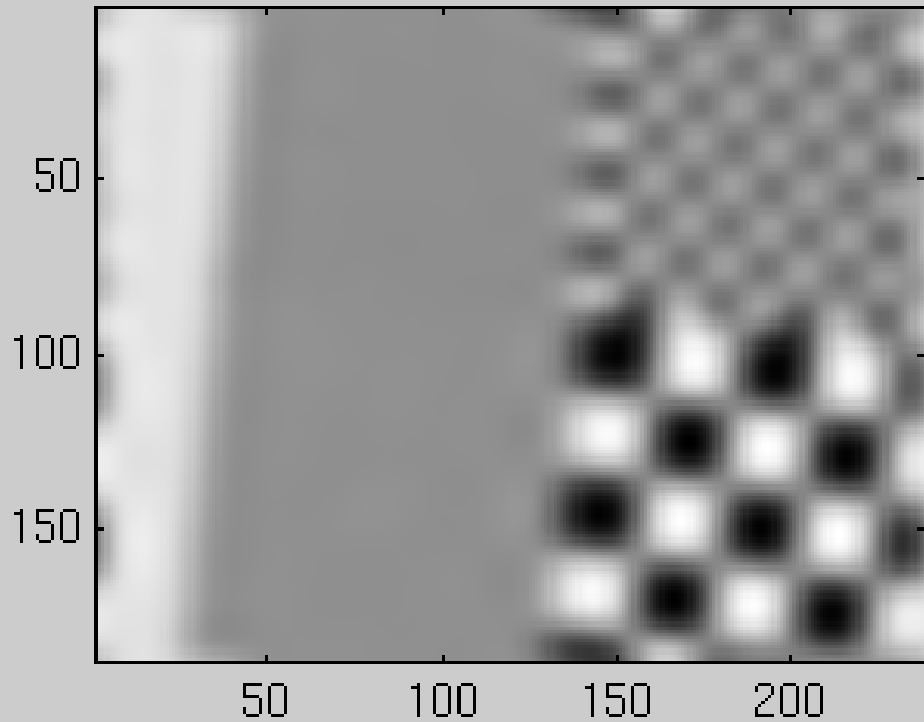


. . . periodicity of the image.

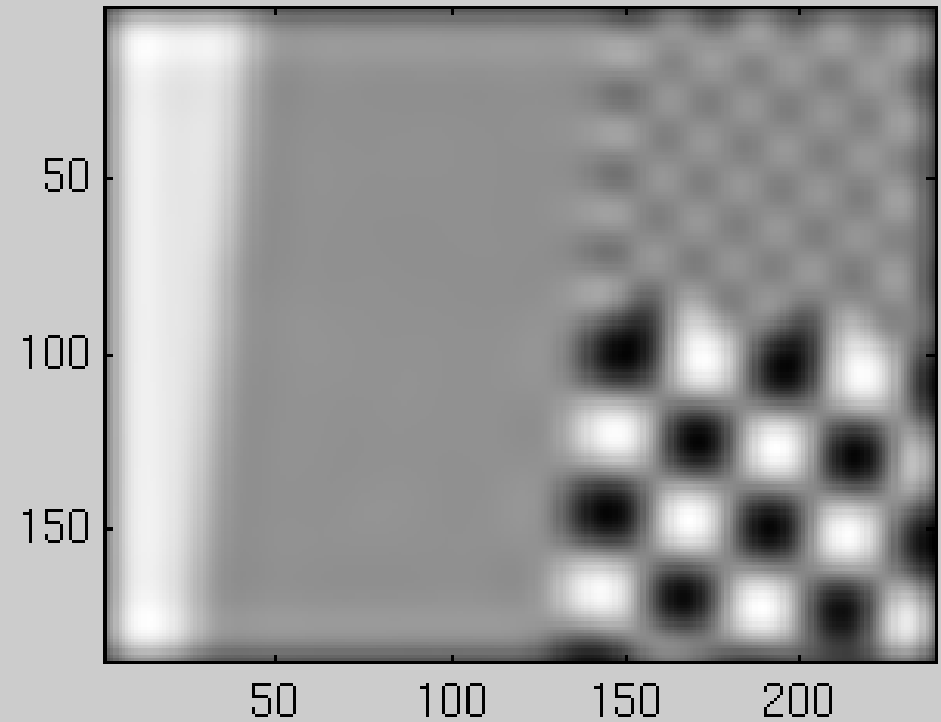
Periodicity of Fourier Transform in Images – cont.



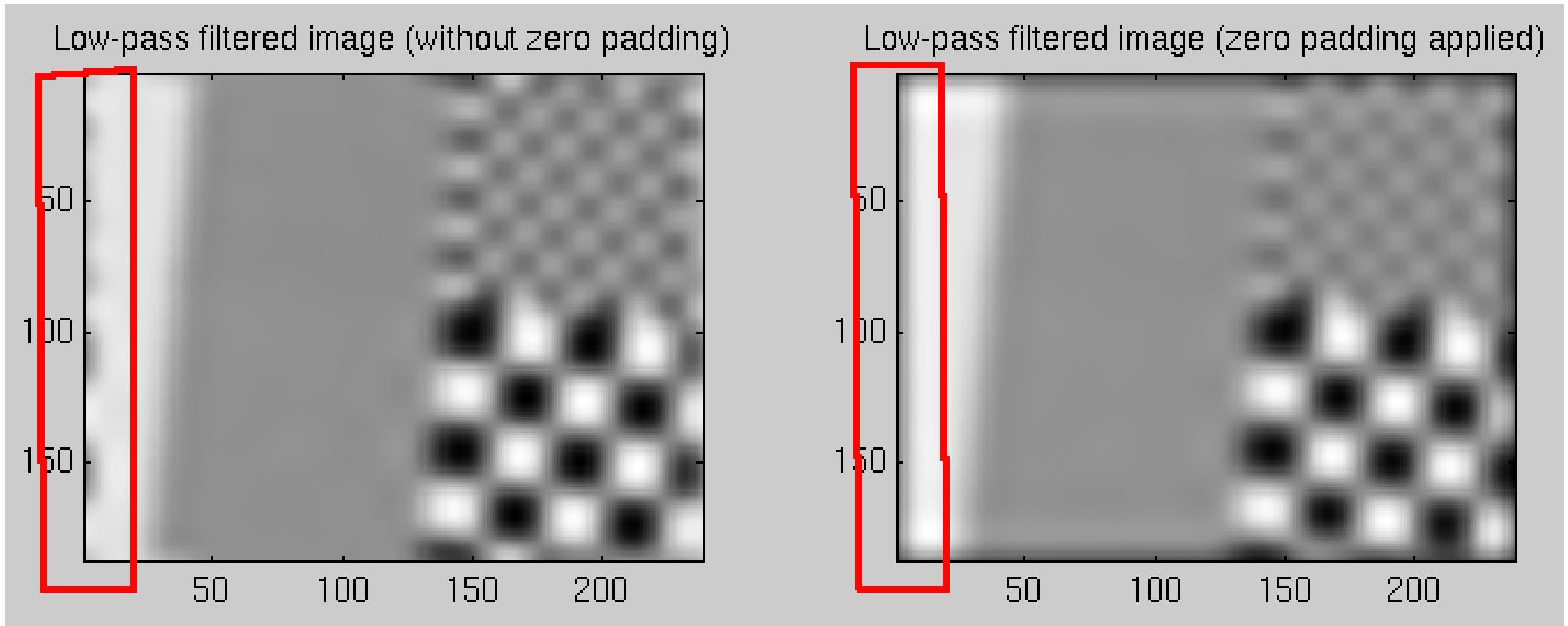
Low-pass filtered image (without zero padding)



Low-pass filtered image (zero padding applied)



Periodicity of Fourier Transform in Images – cont.



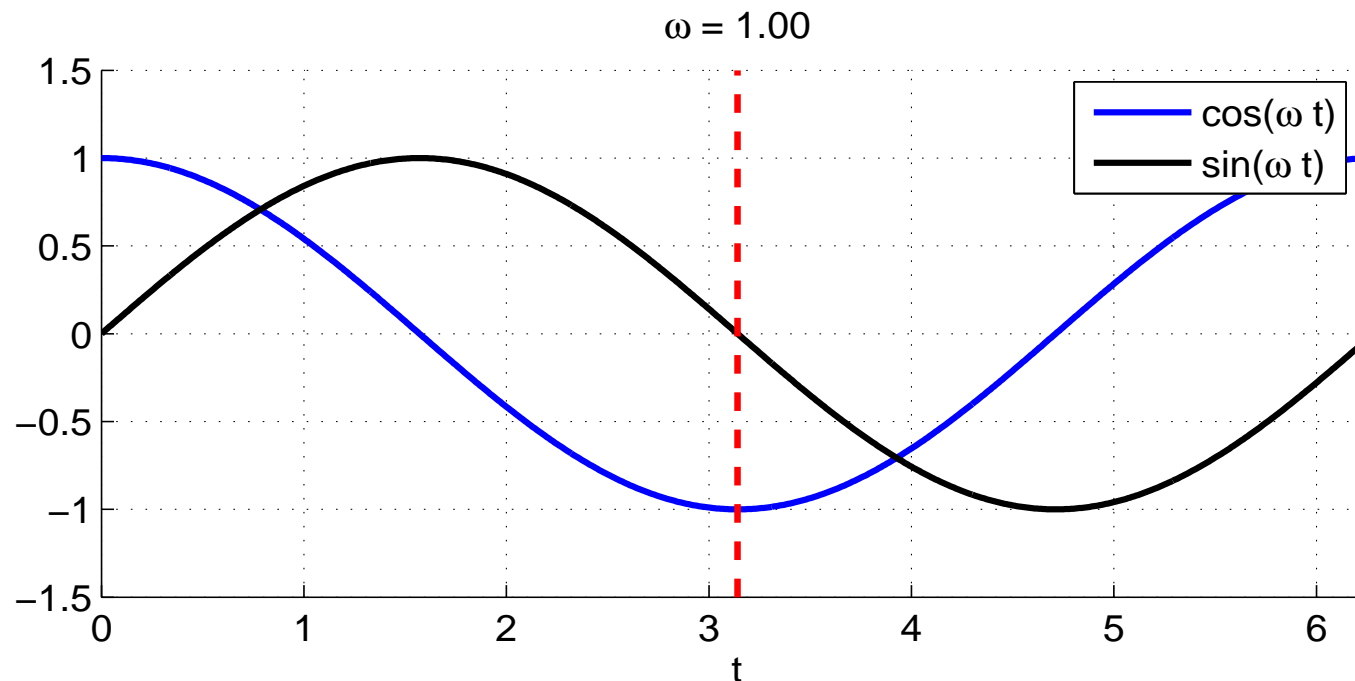
The periodicity is sometimes undesirable, apply zero-padding.

Discrete Fourier transform — conjugate symmetry

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \exp\left(-i2\pi\frac{ux}{M}\right) \text{ for real } f(x).$$

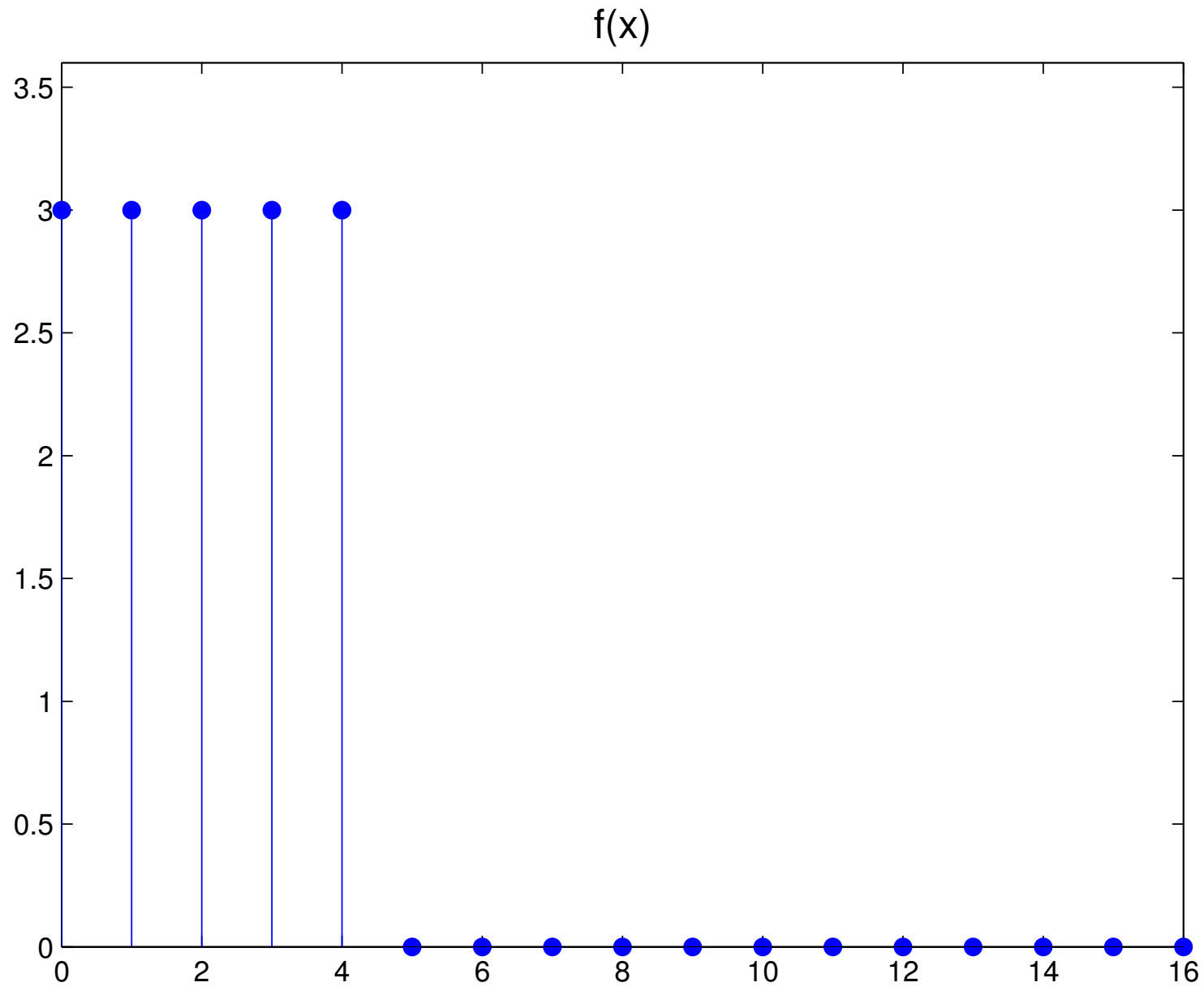
Conjugate symmetry: $F(M - u) = F^*(u)$, $u = 0, \dots, M - 1$.

Remind $e^{-i2\pi ux/M} = \cos(2\pi ux/M) + i \sin(2\pi ux/M)$.

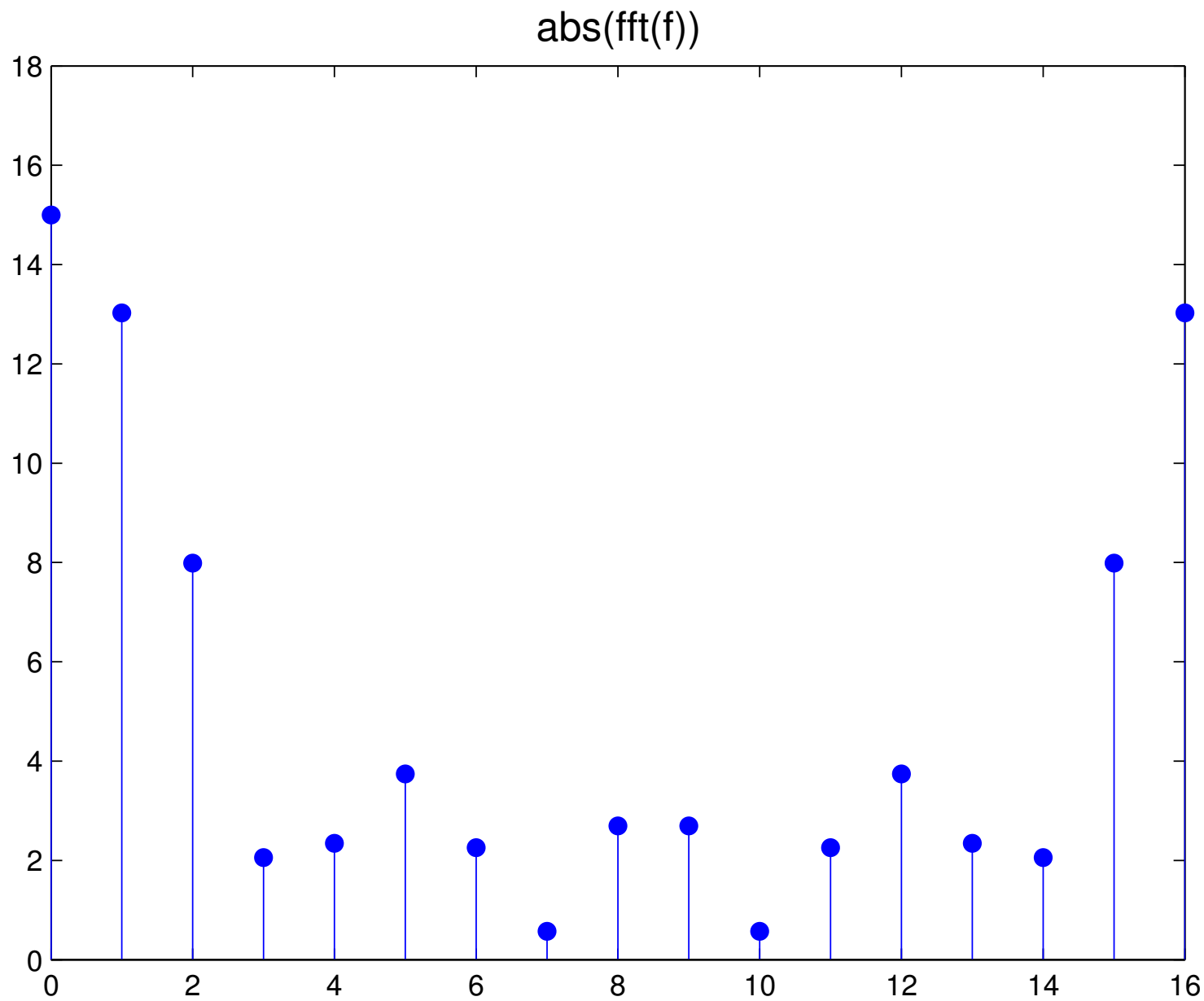
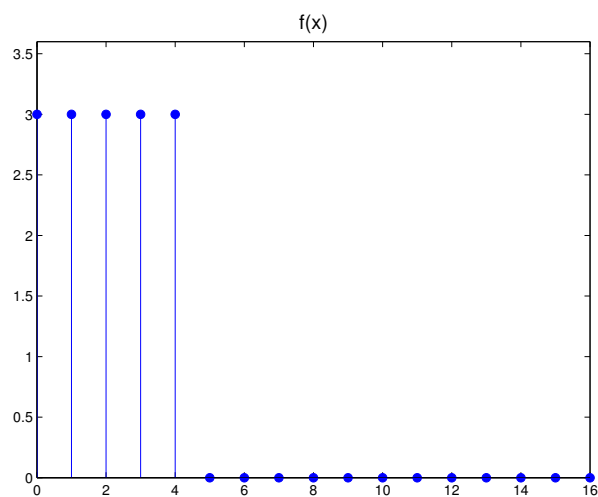


Note that conjugate symmetry implies $|F(M - u)| = |F(u)|$.

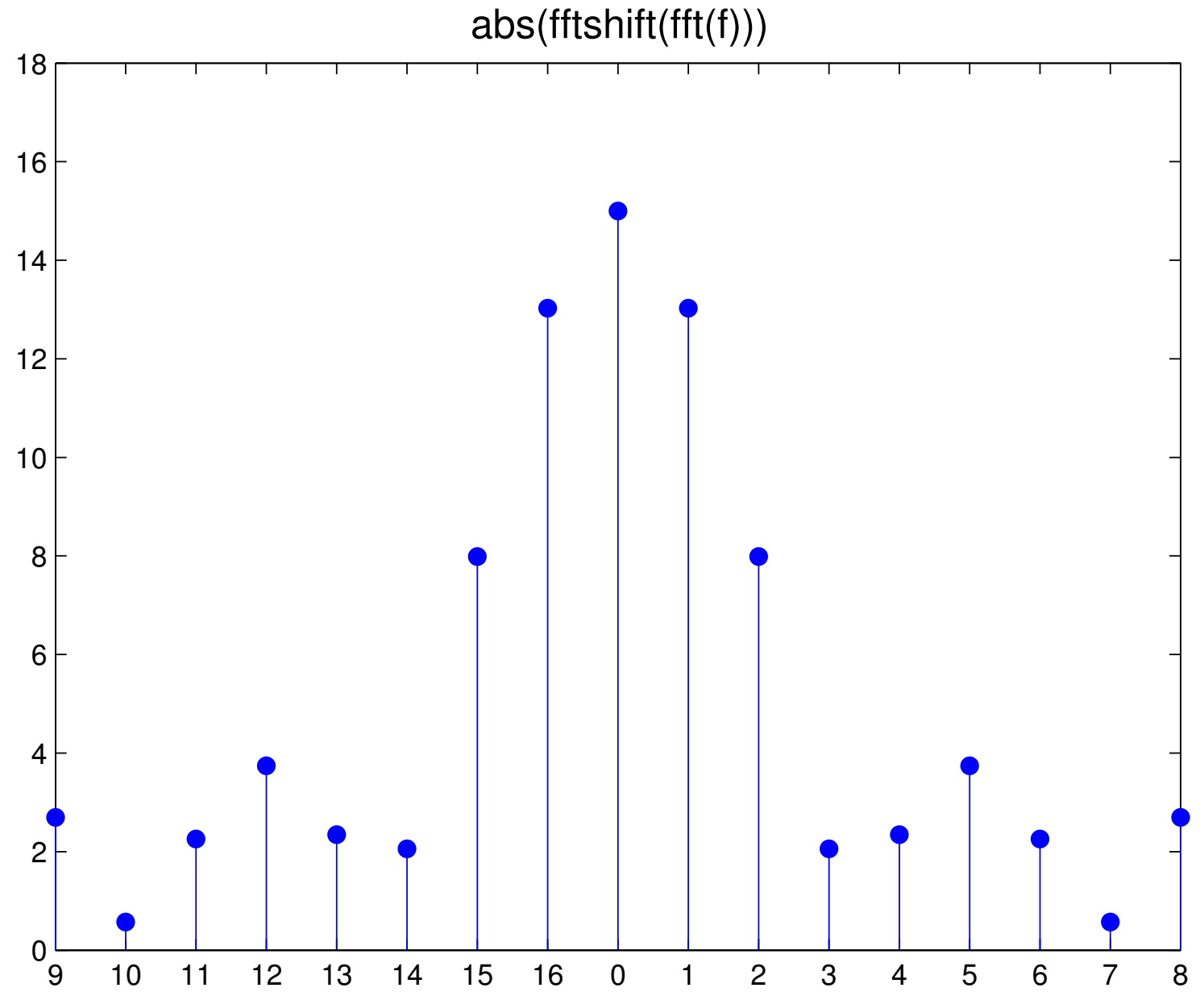
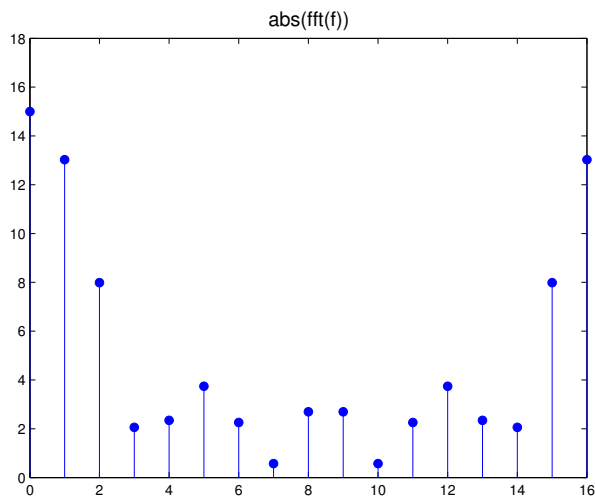
Conjugate symmetry in practice – Input



Conjugate symmetry in practice – Fourier image

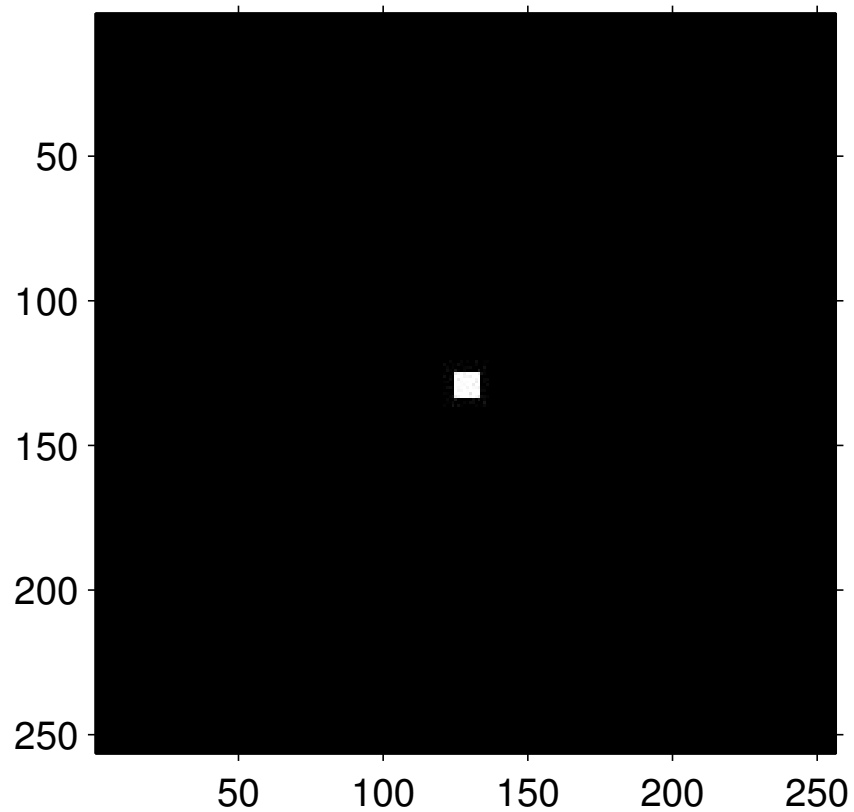


Conjugate symmetry in practice – F with centered origin

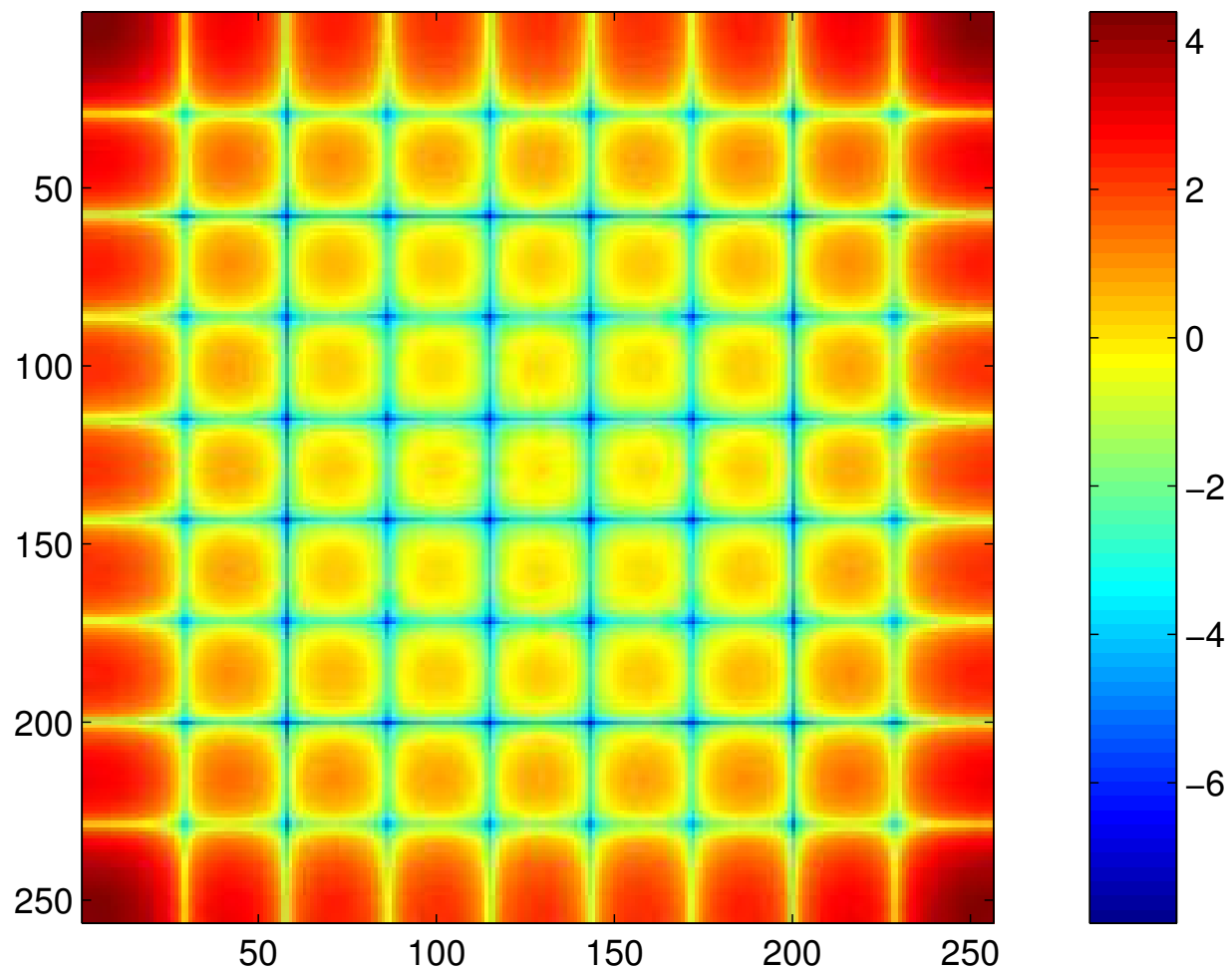


Conjugate symmetry in images

image

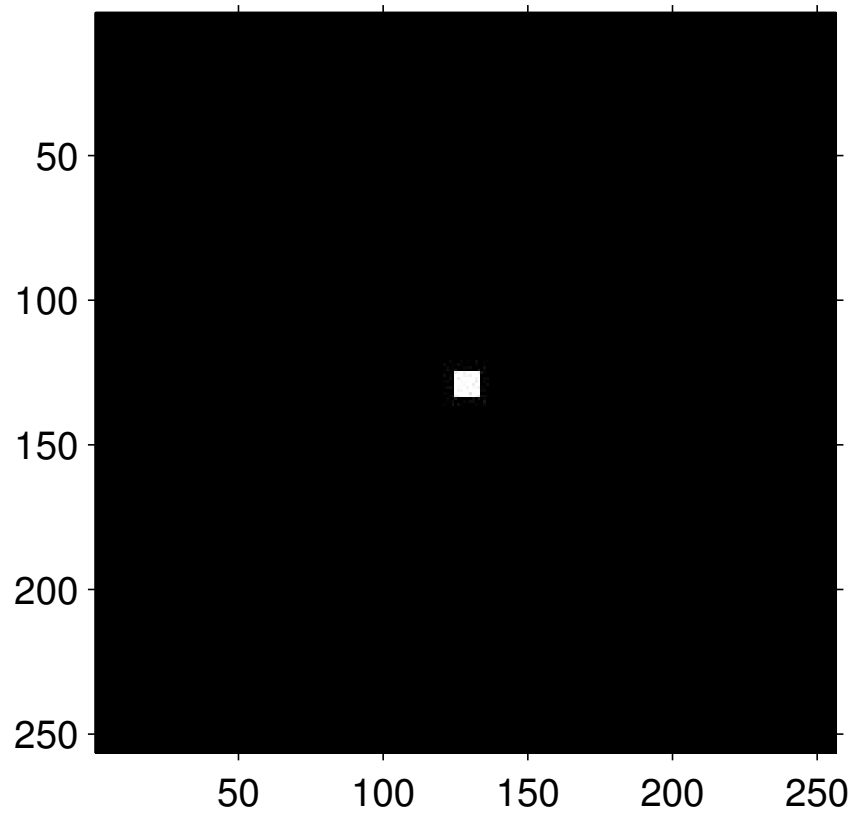


$\log(\text{abs}(\text{fft2}(\text{im})))$

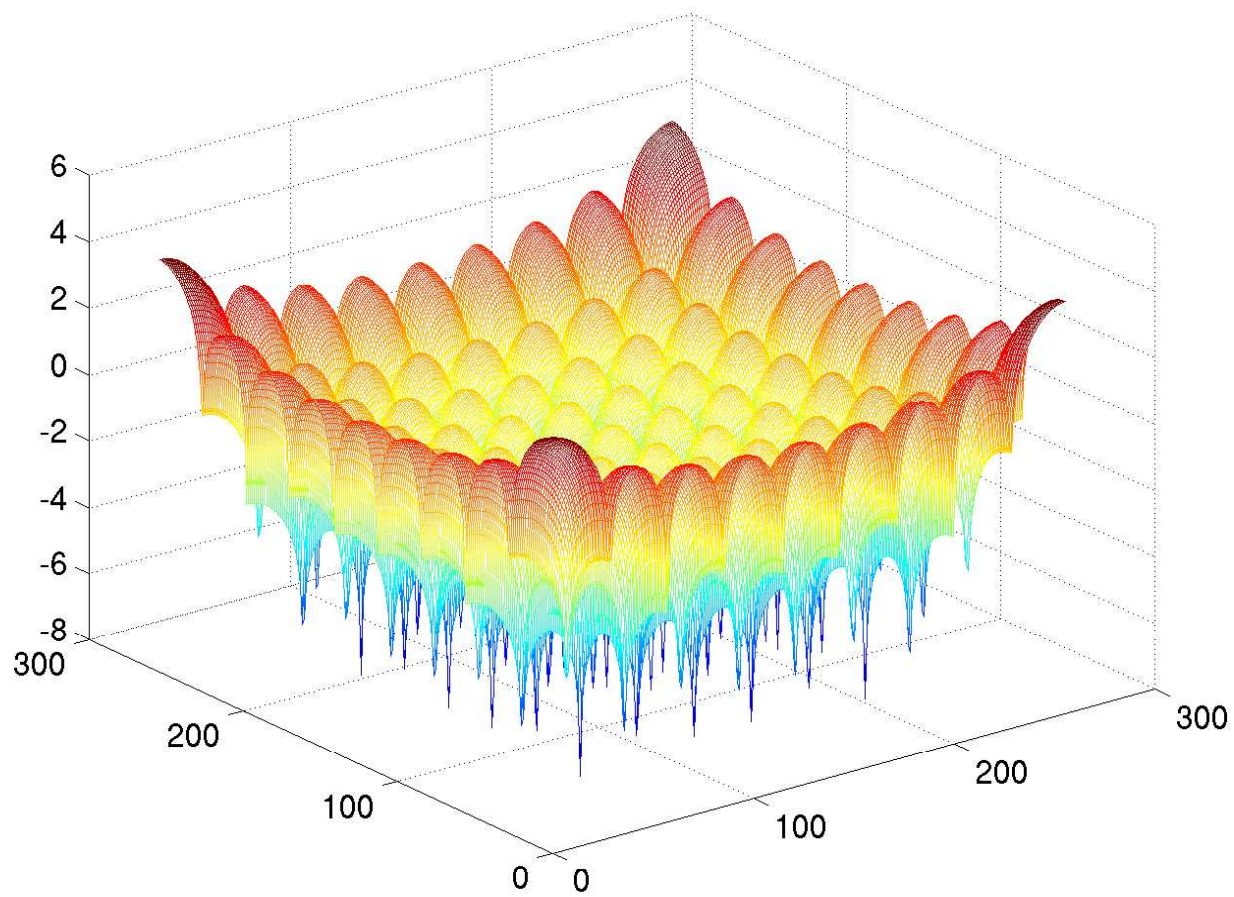


Conjugate symmetry in images

image

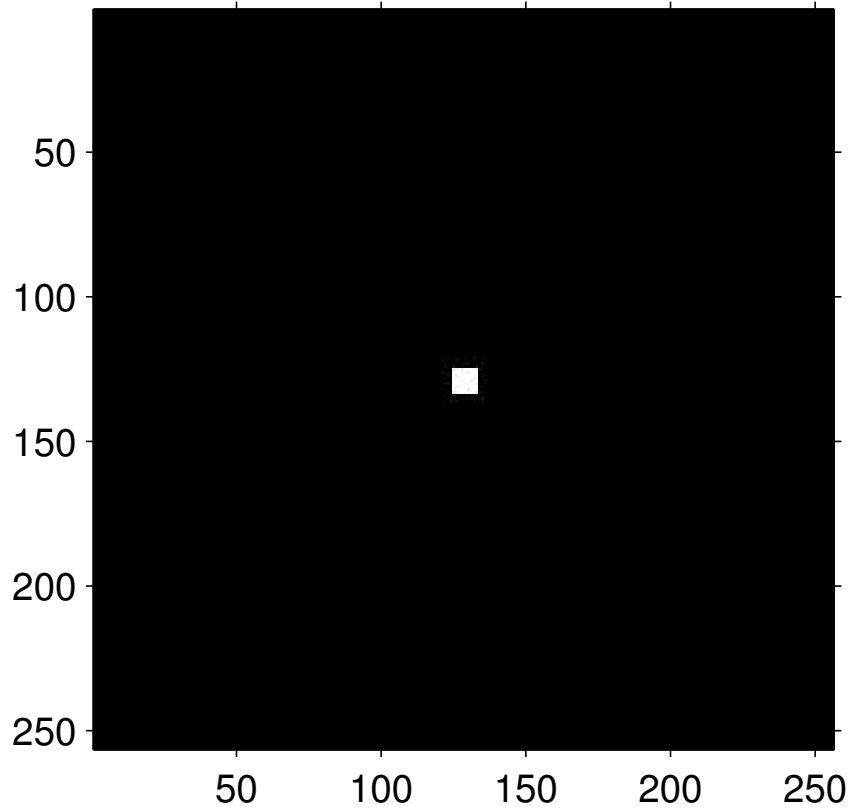


$\log(\text{abs}(\text{fft2}(\text{im})))$

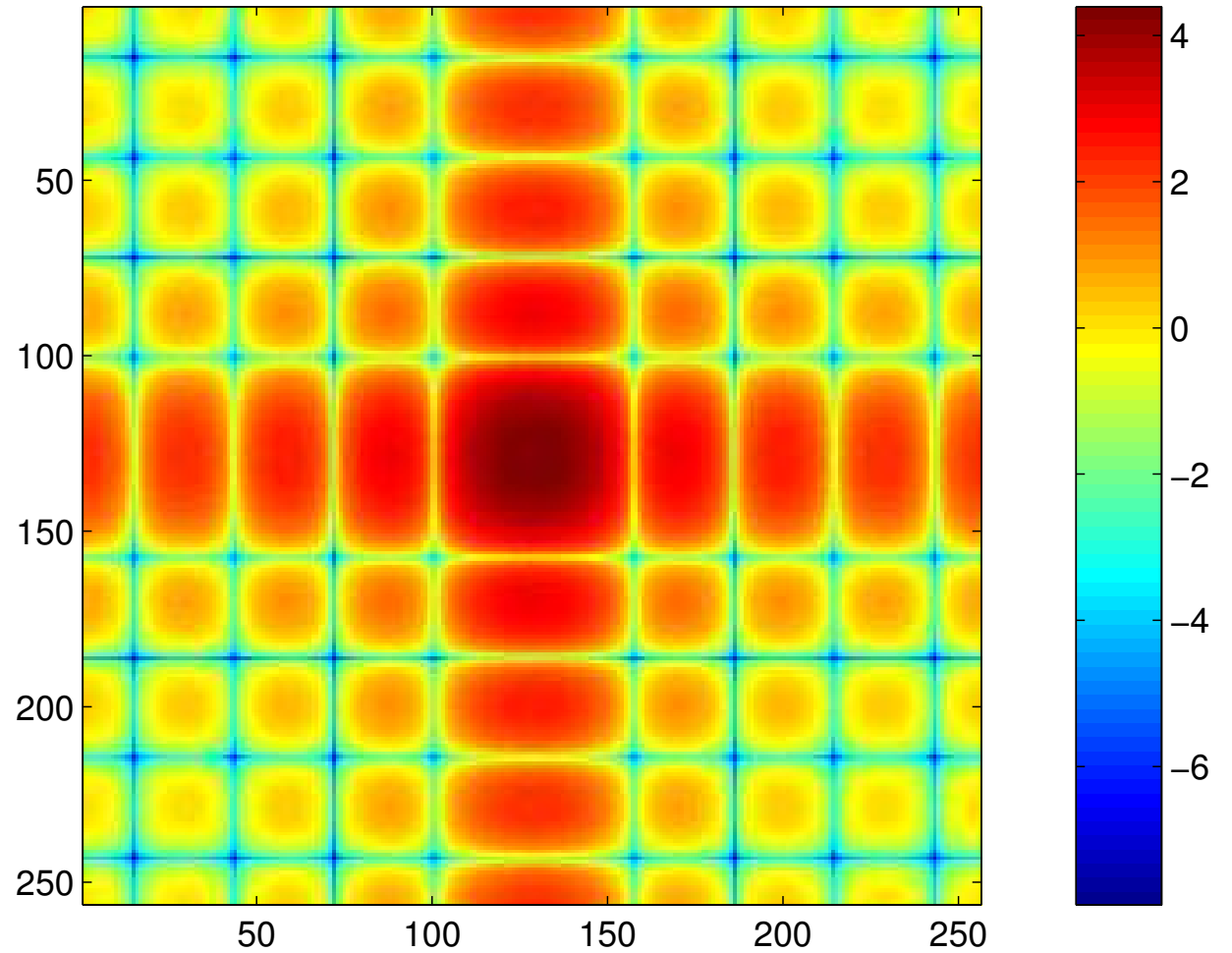


Conjugate symmetry in images – centered origin

image



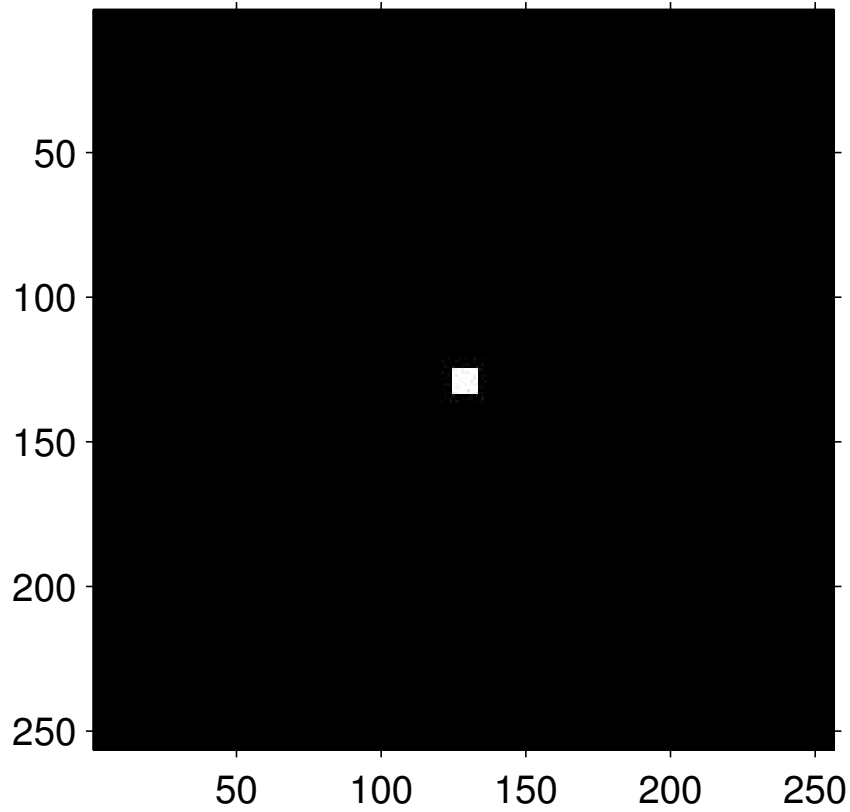
$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$



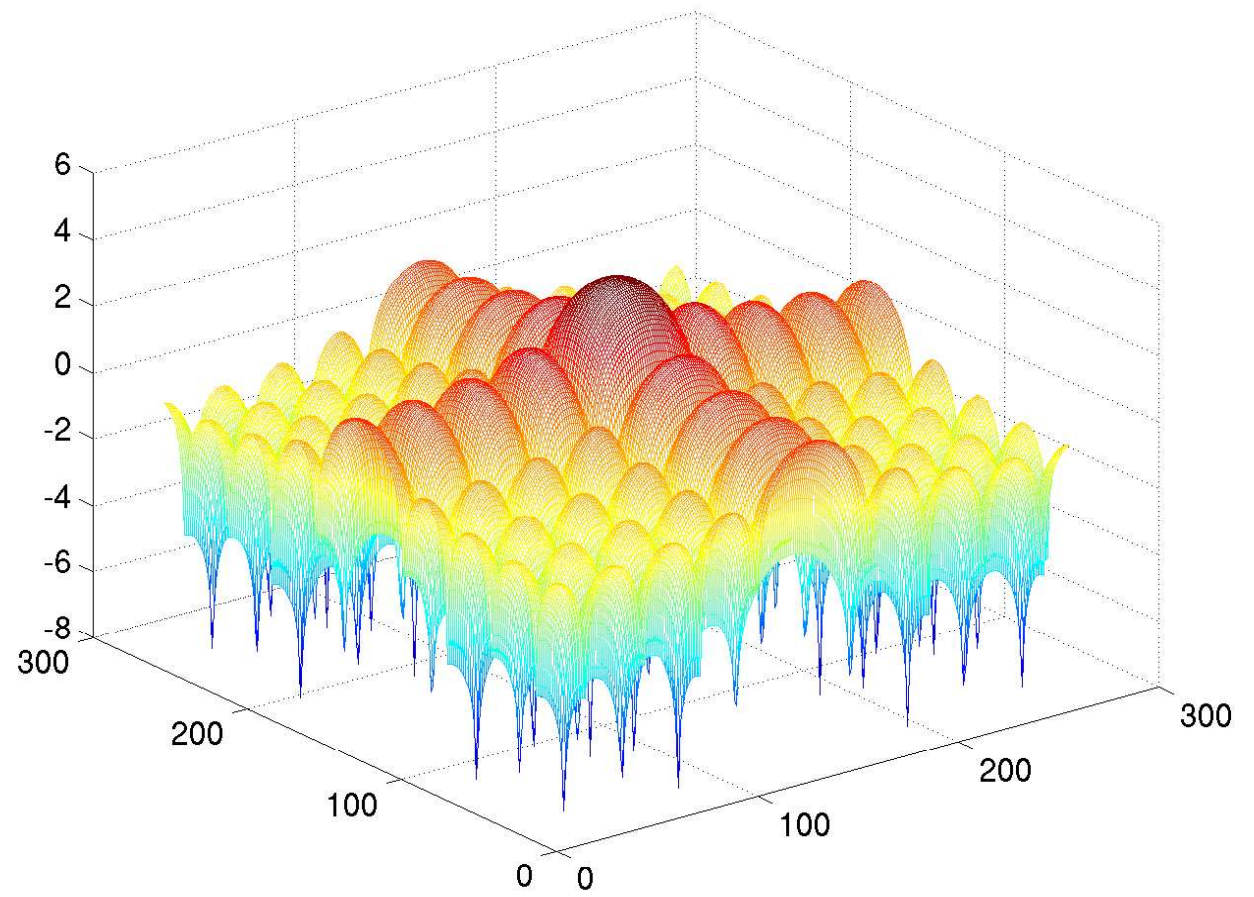
Conjugate symmetry in images – centered origin



image



$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$



Fourier pair of rectangle function

$f_A(x) = A$ for $0 \leq x \leq X$ and $f_A(x) = 0$ elsewhere.

For the derivation we will need:

Fourier pair of rectangle function

$f_A(x) = A$ for $0 \leq x \leq X$ and $f_A(x) = 0$ elsewhere.

For the derivation we will need:

$$\sin(\omega x) = \frac{1}{2i} (e^{i\omega x} - e^{-i\omega x})$$

Fourier pair of rectangle function

$f_A(x) = A$ for $0 \leq x \leq X$ and $f_A(x) = 0$ elsewhere.

For the derivation we will need:

$$\sin(\omega x) = \frac{1}{2i} (e^{i\omega x} - e^{-i\omega x})$$

$$\text{sinc}(\omega x) = \frac{\sin(\omega x)}{\omega x}$$

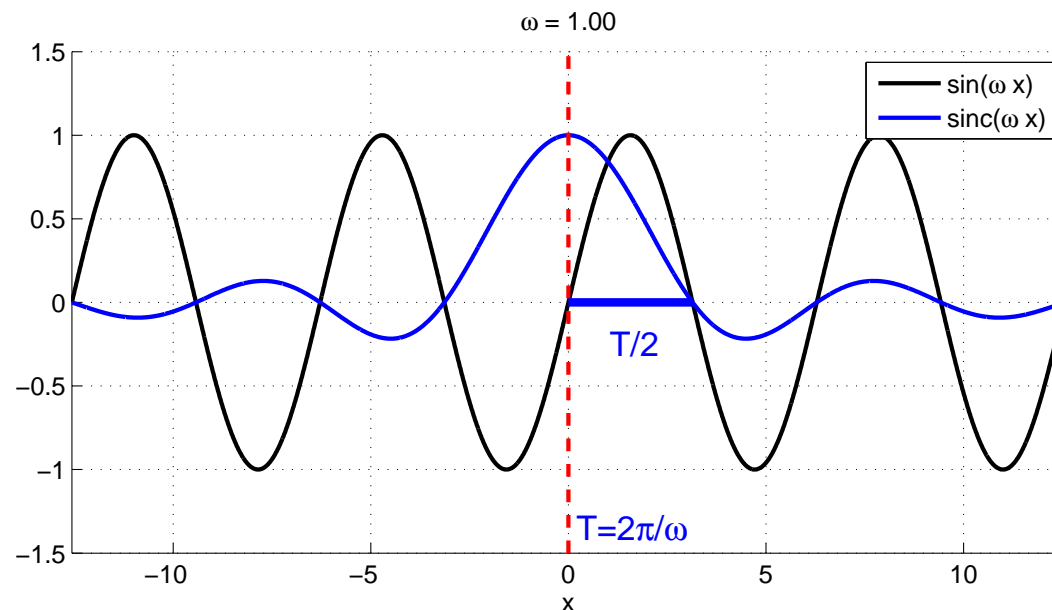
Fourier pair of rectangle function

$f_A(x) = A$ for $0 \leq x \leq X$ and $f_A(x) = 0$ elsewhere.

For the derivation we will need:

$$\sin(\omega x) = \frac{1}{2i} (e^{i\omega x} - e^{-i\omega x})$$

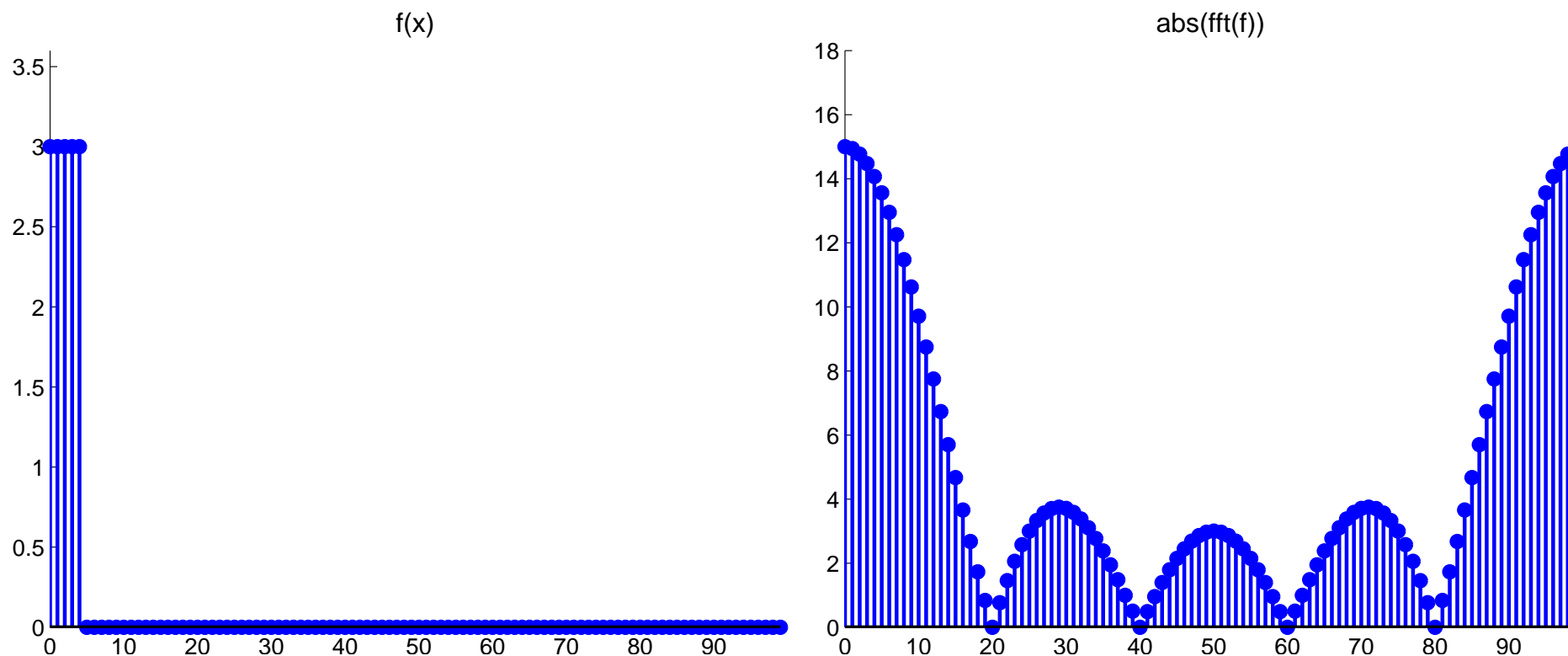
$$\text{sinc}(\omega x) = \frac{\sin(\omega x)}{\omega x}$$



Fourier pair of rectangle function

$$\begin{aligned}
 F(u) &= \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx &&= \frac{A}{\pi u} e^{-i\pi Xu} \sin \pi Xu \\
 &= A \int_0^X e^{-i2\pi ux} dx &&= AX e^{-i\pi Xu} \frac{\sin \pi Xu}{\pi Xu} \\
 &= A \left[\frac{e^{-i2\pi ux}}{-i2\pi u} \right]_0^X &&= AX e^{-i\pi Xu} \operatorname{sinc}(\pi Xu) \\
 &= \frac{A}{-i2\pi u} (e^{-i2\pi Xu} - 1) && \text{which implies} \\
 &= \frac{A}{-i2\pi u} (-e^{-i\pi Xu}) (e^{i\pi Xu} - e^{-i\pi Xu}) && |F(u)| = AX |\operatorname{sinc}(\pi Xu)| \\
 &= \frac{A}{\pi u} e^{-i\pi Xu} \left(\frac{e^{i\pi Xu} - e^{-i\pi Xu}}{2i} \right)
 \end{aligned}$$

Fourier pair of discrete rectangle function



$A = 3, X = 5, M = 100$ makes $T/2$ of sinc equal to 20.

$$\mathcal{F}\{f_A(x) = A \text{ for } 0 \leq x \leq X\} = AX e^{-i\pi Xu} \text{sinc}\left(\frac{\pi X}{M}u\right)$$

Remember that $T = 2\pi/\omega$, hence

$$X = \frac{2M}{T}$$

We can find the frequency of $|F(u)|$ perhaps even simpler . . . ¹

¹Blackboard only for the moment.

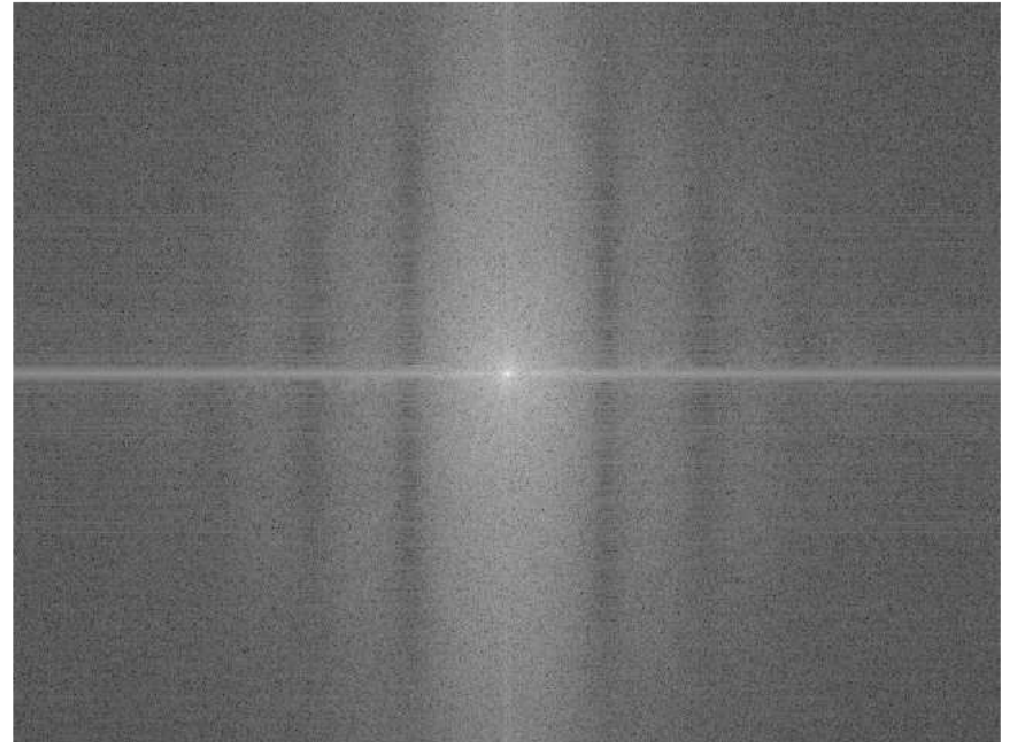
Use of Rectangle function — Image Restoration



Use of Rectangle function — Image Restoration



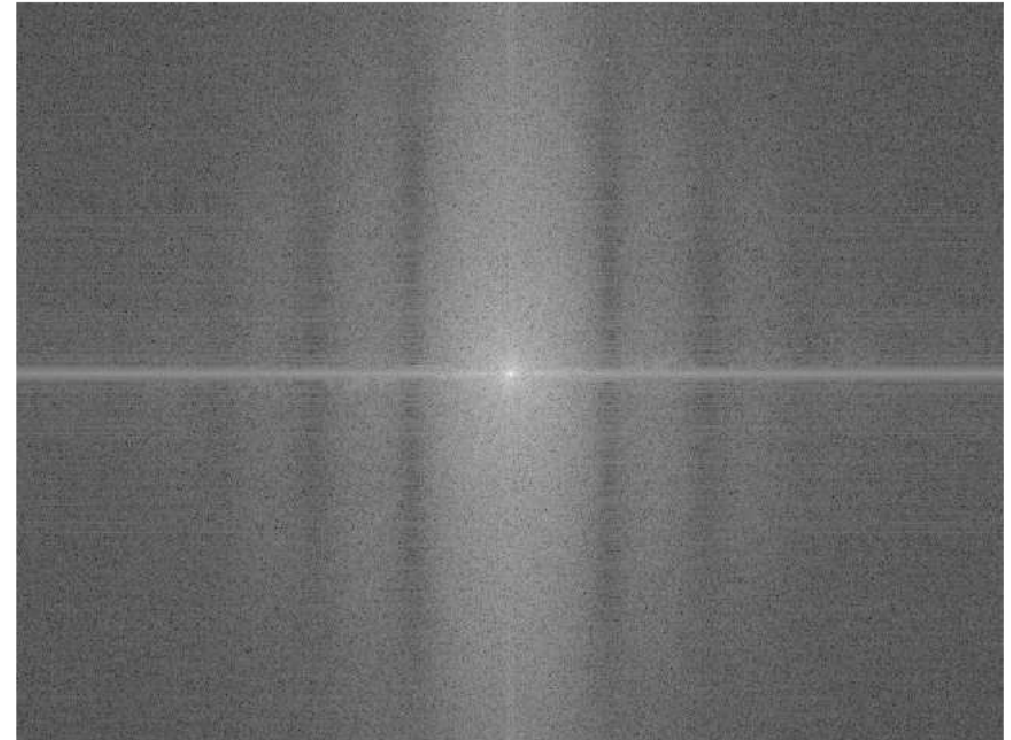
spectrum of blurred image



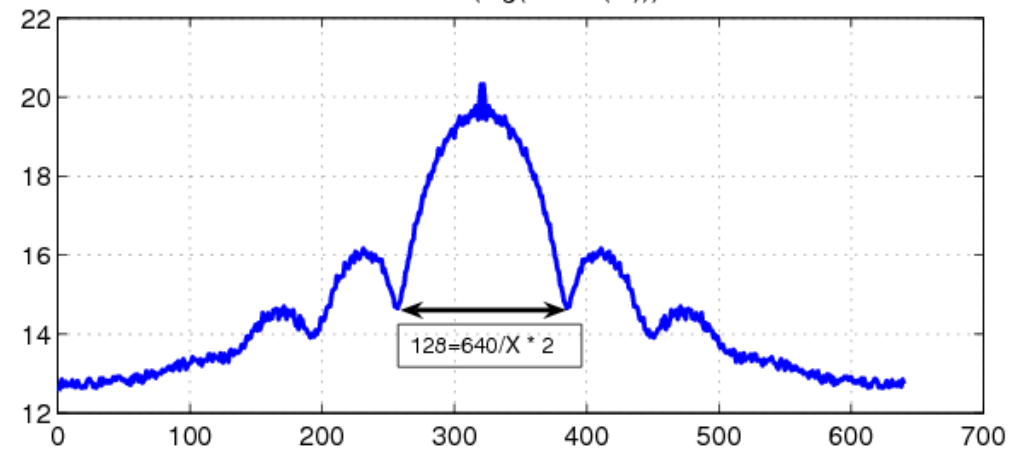
Use of Rectangle function — Image Restoration



spectrum of blurred image



median(log(1+abs(fft)))



Use of Rectangle function — Image Restoration



Blurred image.

Use of Rectangle function — Image Restoration



Blurred image.



Restored image.

Image processing in frequency domain — Filtering



Image processing in frequency domain — Filtering

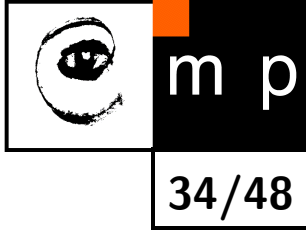


1. $F(u, v) = \mathcal{F}\{f(x, y)\}$

Image processing in frequency domain — Filtering

1. $F(u, v) = \mathcal{F}\{f(x, y)\}$
2. $G(u, v) = H(u, v) \cdot * F(u, v)$, where $\cdot *$ means “per element” multiplication.

Image processing in frequency domain — Filtering



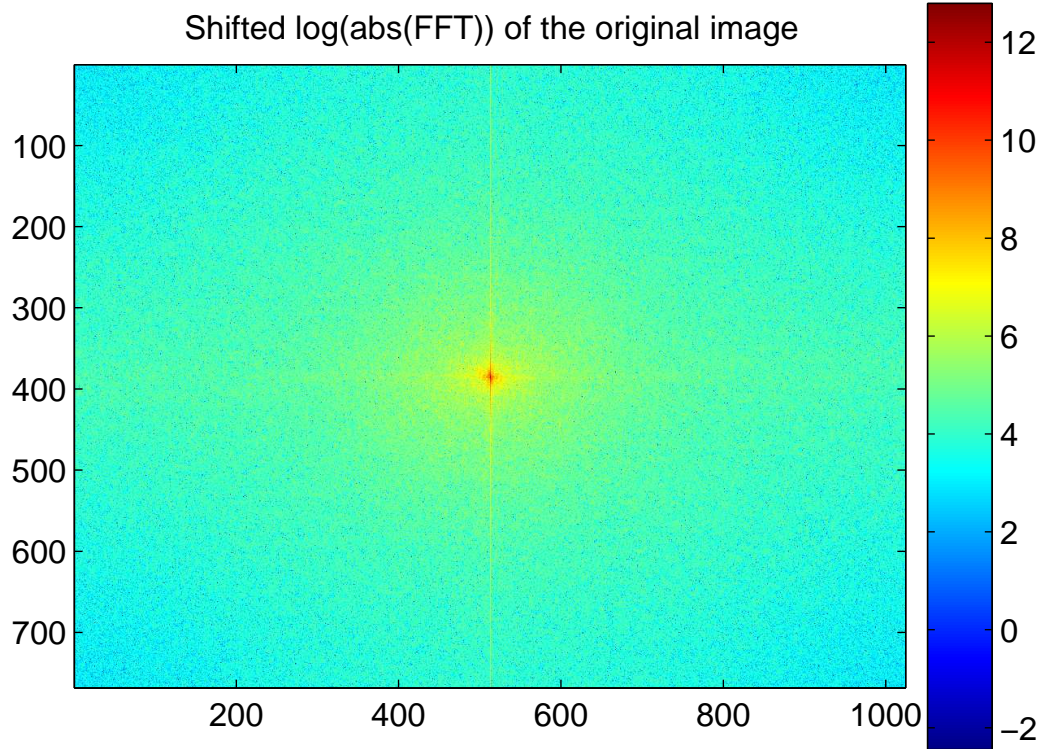
1. $F(u, v) = \mathcal{F}\{f(x, y)\}$
2. $G(u, v) = H(u, v) \cdot * F(u, v)$, where $\cdot *$ means “per element” multiplication.
3. $g(x, y) = \mathcal{F}^{-1}\{G(u, v)\}$

Image processing in frequency domain — Filtering

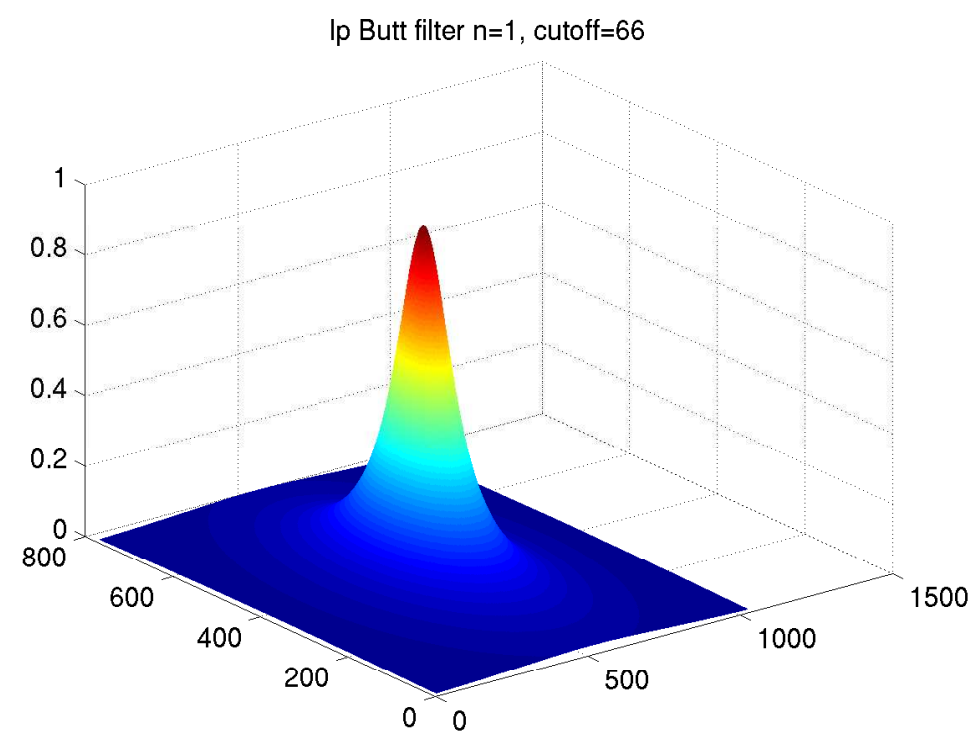
1. $F(u, v) = \mathcal{F}\{f(x, y)\}$
2. $G(u, v) = H(u, v) .* F(u, v)$, where $.*$ means “per element” multiplication.
3. $g(x, y) = \mathcal{F}^{-1}\{G(u, v)\}$

Do not forget: We display $\ln |F(u, v)|$. The filter must be applied to the $F(u, v)$.

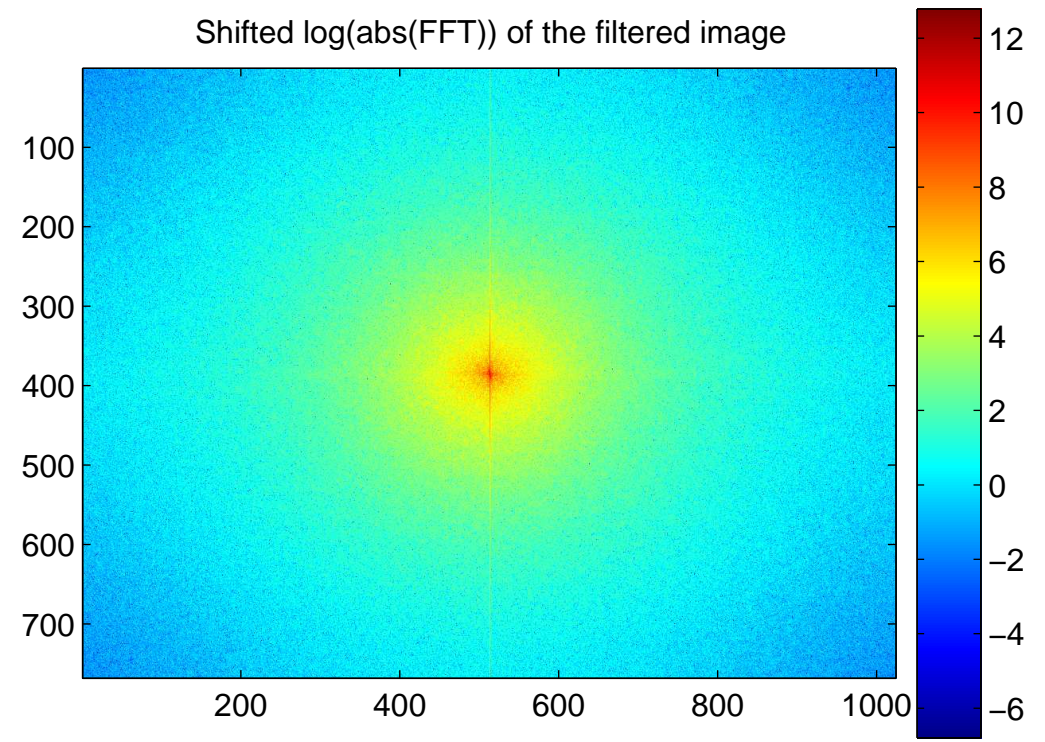
Lowpass filtering — Butterworth filter I



Lowpass filtering — Butterworth filter II



Butterworth lowpass filter



FFT of the filtered image

$$H_{lp}(u, v) = \frac{1}{1+(D(u,v)/D_0)^{2/n}}, \text{ where } D(u, v) = \sqrt{u^2 + v^2}$$

Lowpass filtering — Butterworth filter III

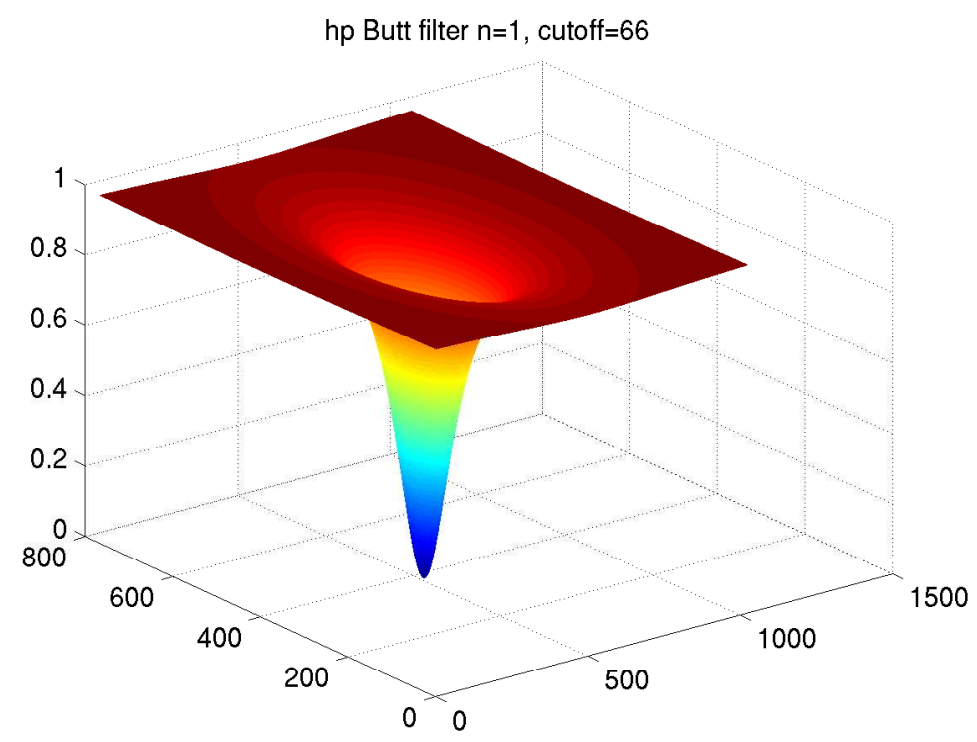


Original image

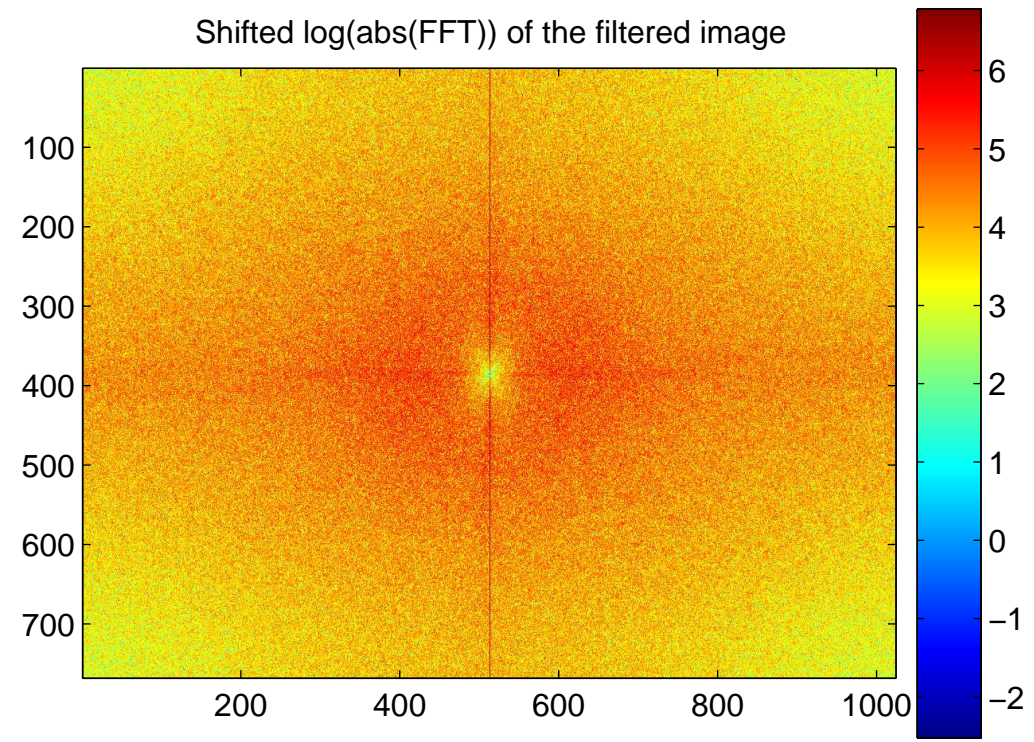


Filtered image

Highpass filtering — Butterworth filter I



Butterworth highpass filter



FFT of the filtered image

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Highpass filtering — Butterworth filter II



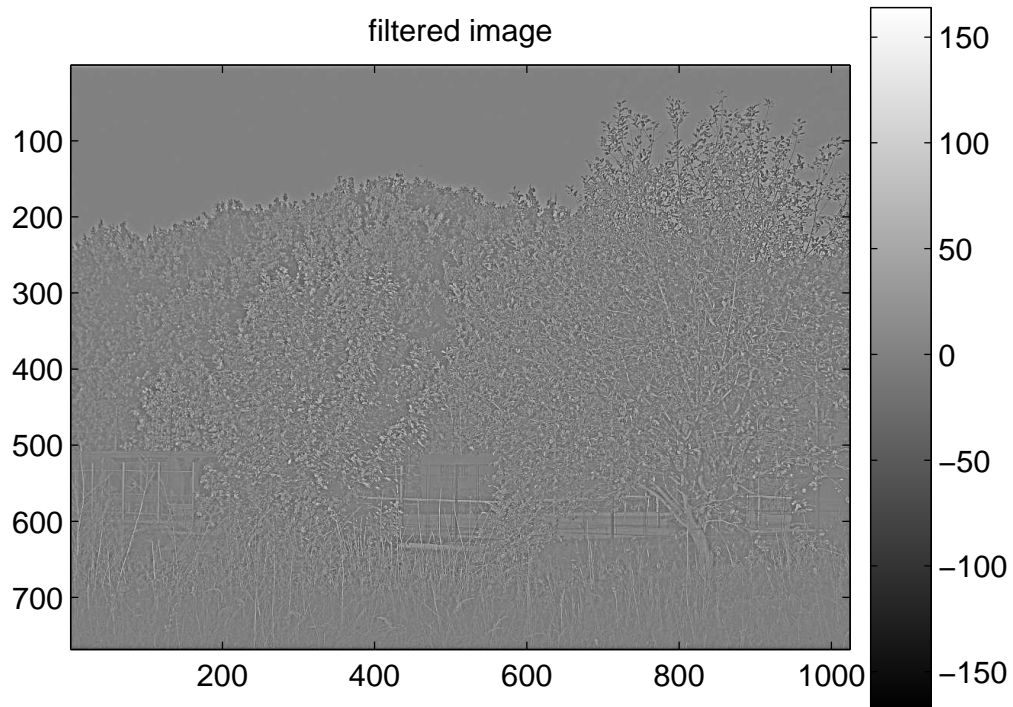
Original image



Filtered image

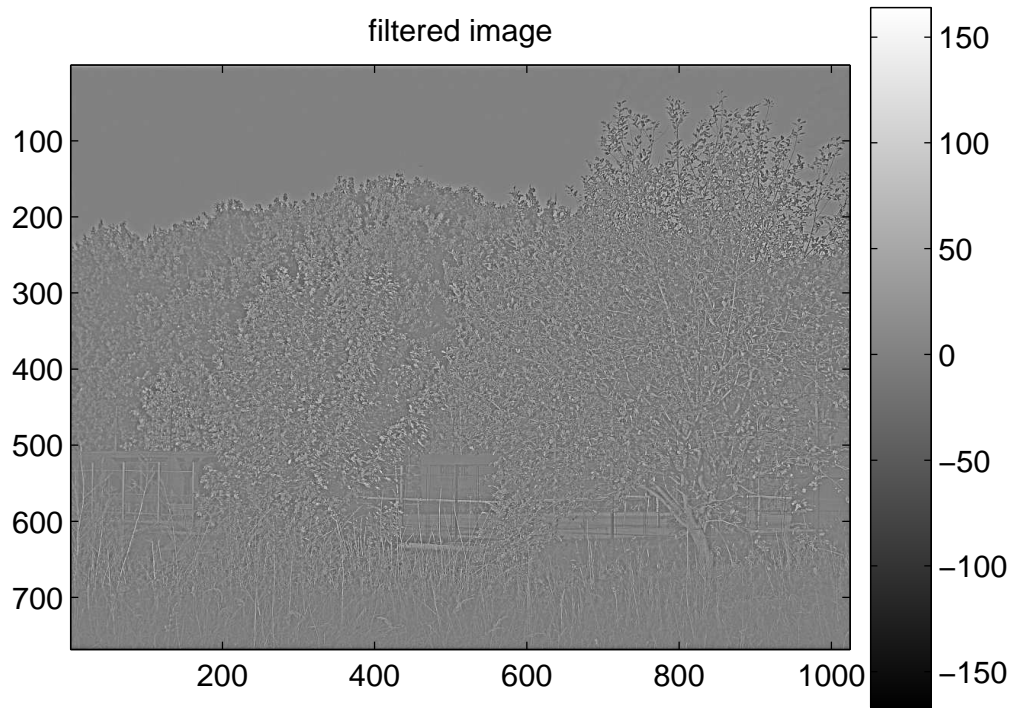
Why do we not see anything in the filtered image?

Highpass filtering — cont.

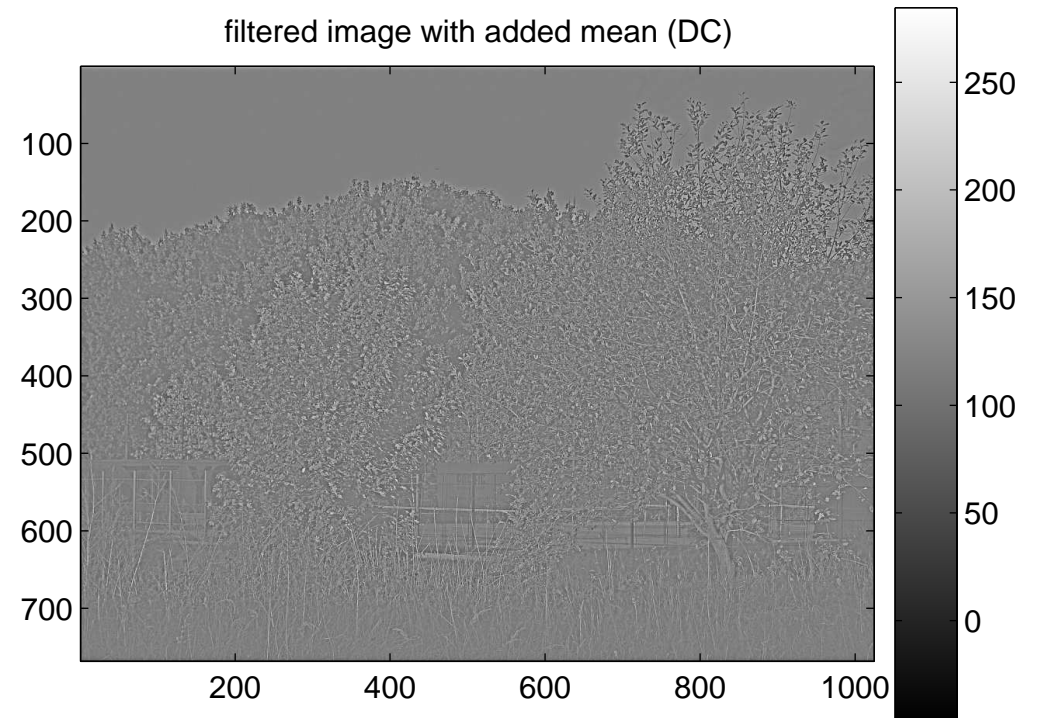


Some values are negative. Why?

Highpass filtering — cont.

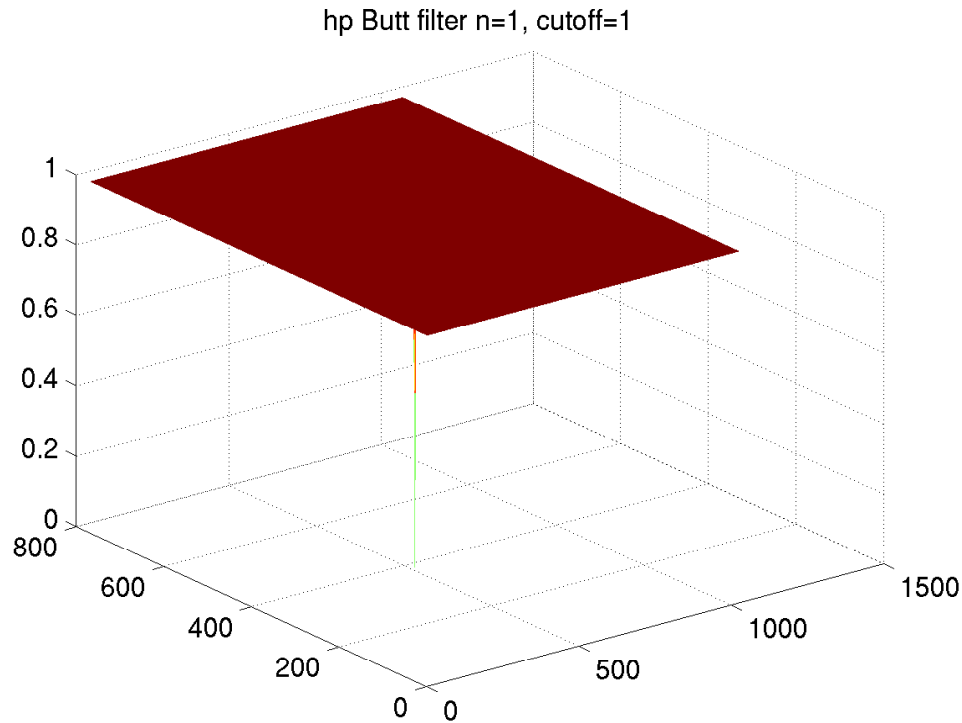


Some values are negative. Why?

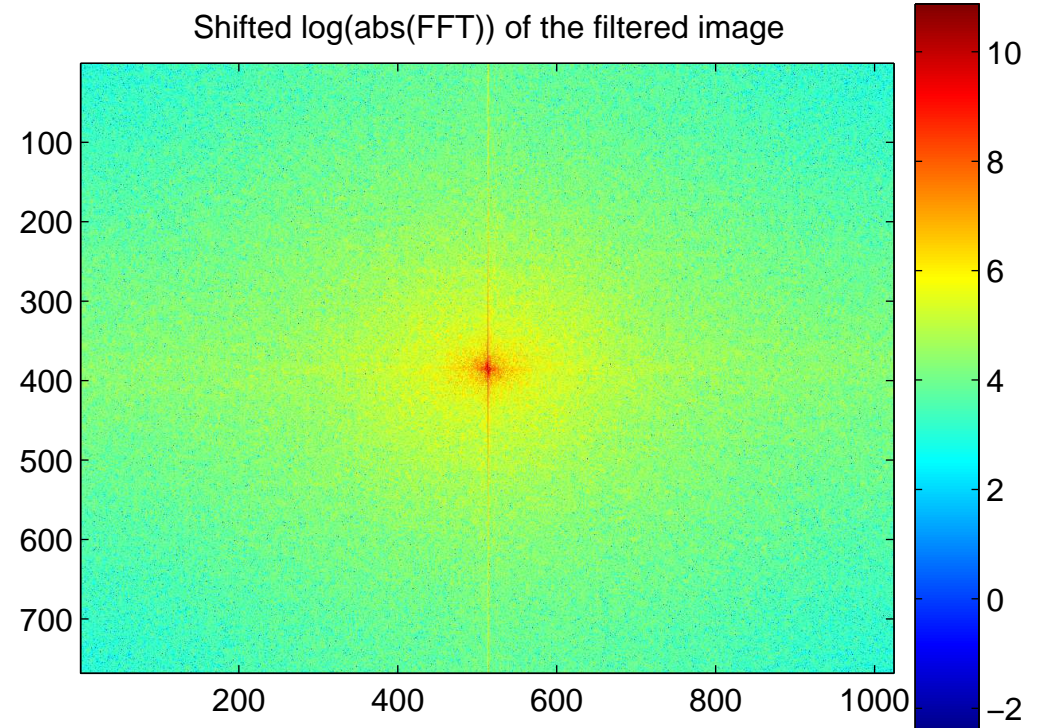


We lost the DC part of the FFT!

Highpass filtering — Loosing DC part I



Butterworth highpass filter



FFT of the filtered image

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Highpass filtering — Loosing DC part II



Original image



Filtered image

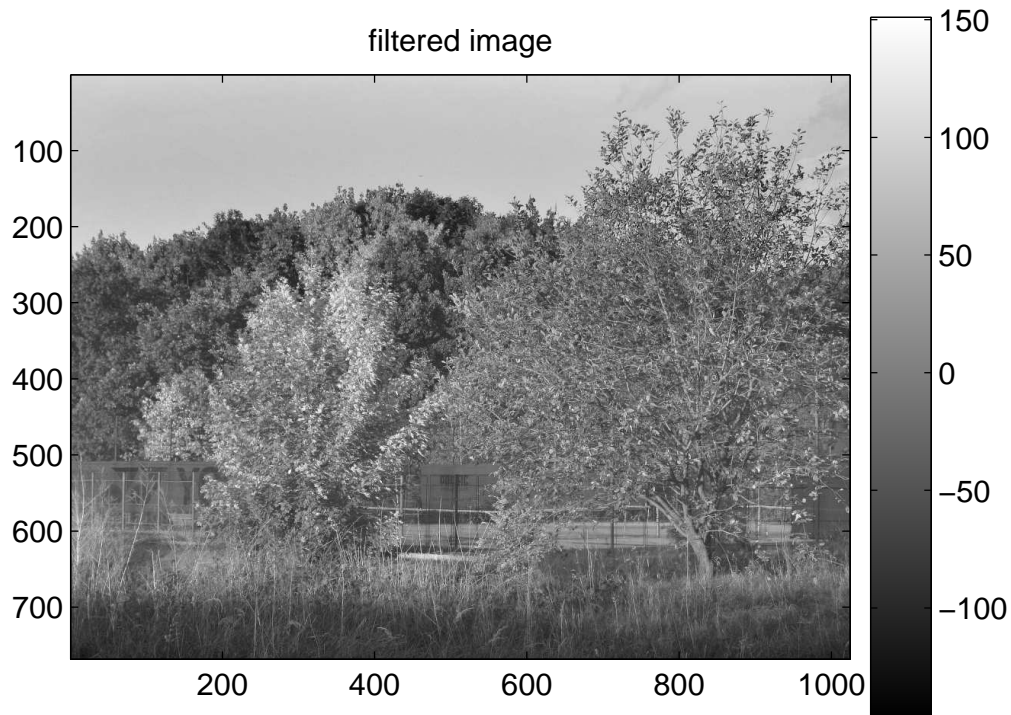
Despite a very gentle high-pass filter the filtered images does not resemble the original one.

Highpass filtering — Loosing DC part III

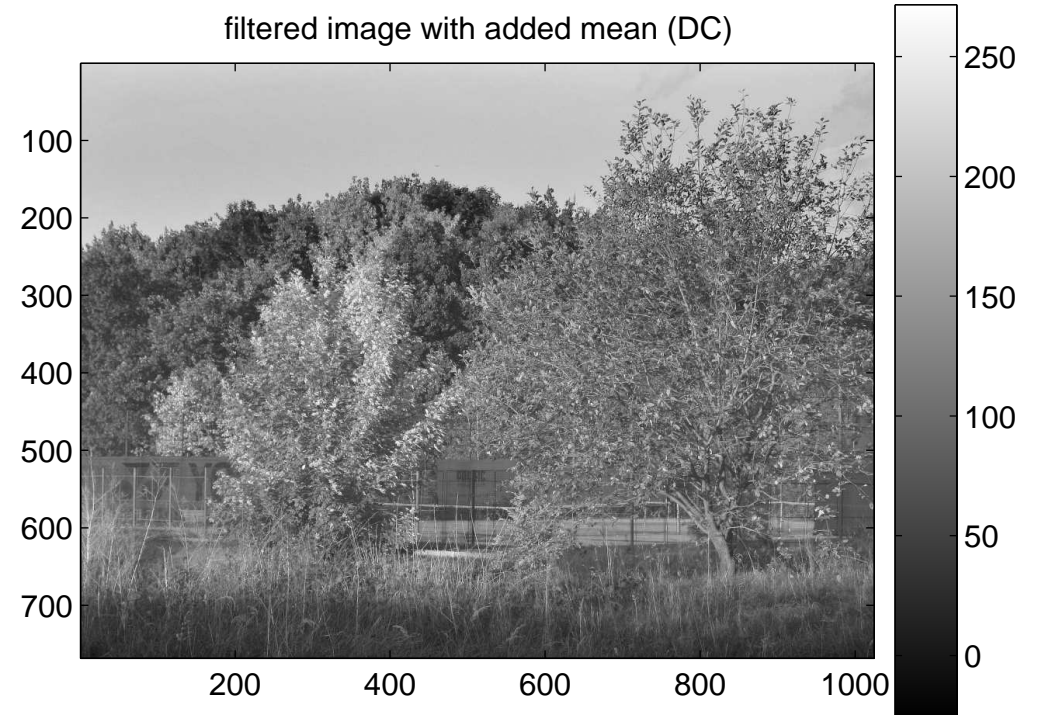


Some values are negative.

Highpass filtering — Loosing DC part III



Some values are negative.



Adding a DC part (mean of the original image) would correct the image.

More advanced filtering — Homomorphic filtering



Idea: simultaneously normalize the brightness across an image and increase contrast.

More advanced filtering — Homomorphic filtering



Idea: simultaneously normalize the brightness across an image and increase contrast.

Image is a product of illumination and reflectance components:

$$f(x, y) = i(x, y)r(x, y)$$

More advanced filtering — Homomorphic filtering



Idea: simultaneously normalize the brightness across an image and increase contrast.

Image is a product of illumination and reflectance components:

$$f(x, y) = i(x, y)r(x, y)$$

Illumination i — slow spatial variations (low frequency)

More advanced filtering — Homomorphic filtering



Idea: simultaneously normalize the brightness across an image and increase contrast.

Image is a product of illumination and reflectance components:

$$f(x, y) = i(x, y)r(x, y)$$

Illumination i — slow spatial variations (low frequency)

Reflectance r — fast variations (dissimilar objects)

More advanced filtering — Homomorphic filtering



Idea: simultaneously normalize the brightness across an image and increase contrast.

Image is a product of illumination and reflectance components:

$$f(x, y) = i(x, y)r(x, y)$$

Illumination i — slow spatial variations (low frequency)

Reflectance r — fast variations (dissimilar objects)

Use logarithm to separate the components and filter the logarithms!

Homomorphic filtering — cont.

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

Homomorphic filtering — cont.

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

Fourier pair

$$Z(u, v) = I(u, v) + R(u, v)$$

Homomorphic filtering — cont.

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

Fourier pair

$$Z(u, v) = I(u, v) + R(u, v)$$

Filtering

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v)$$

Homomorphic filtering — cont.

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

Fourier pair

$$Z(u, v) = I(u, v) + R(u, v)$$

Filtering

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v)$$

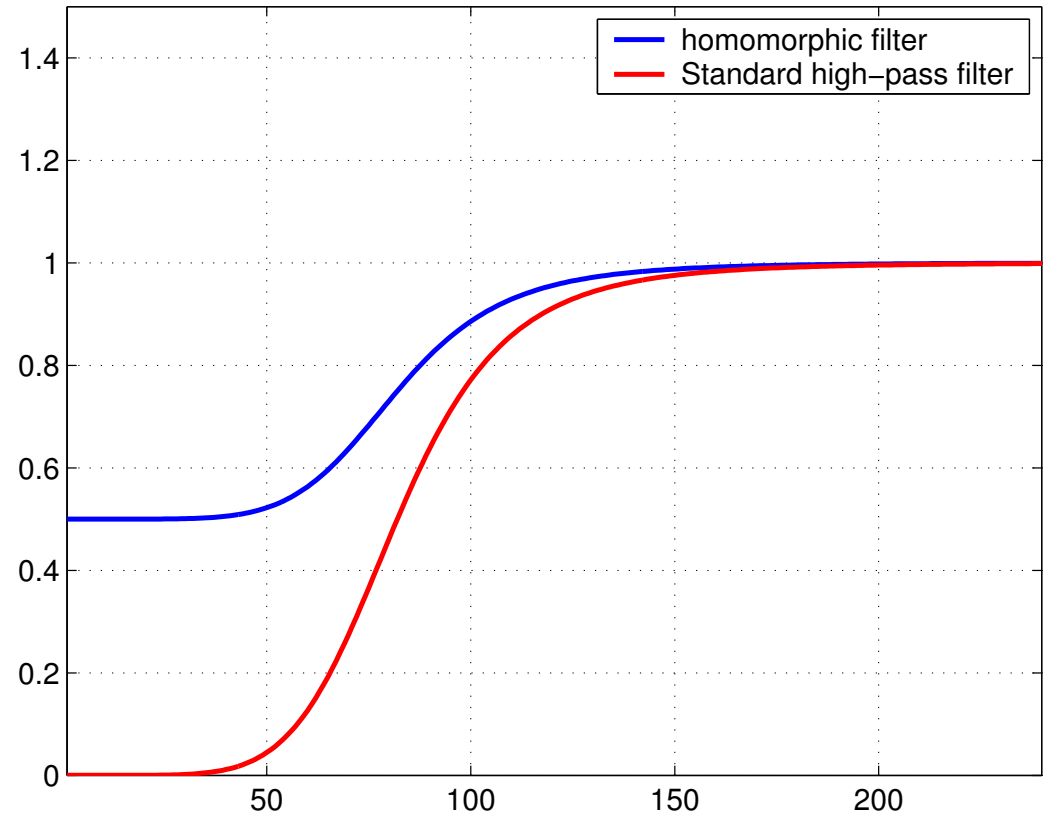
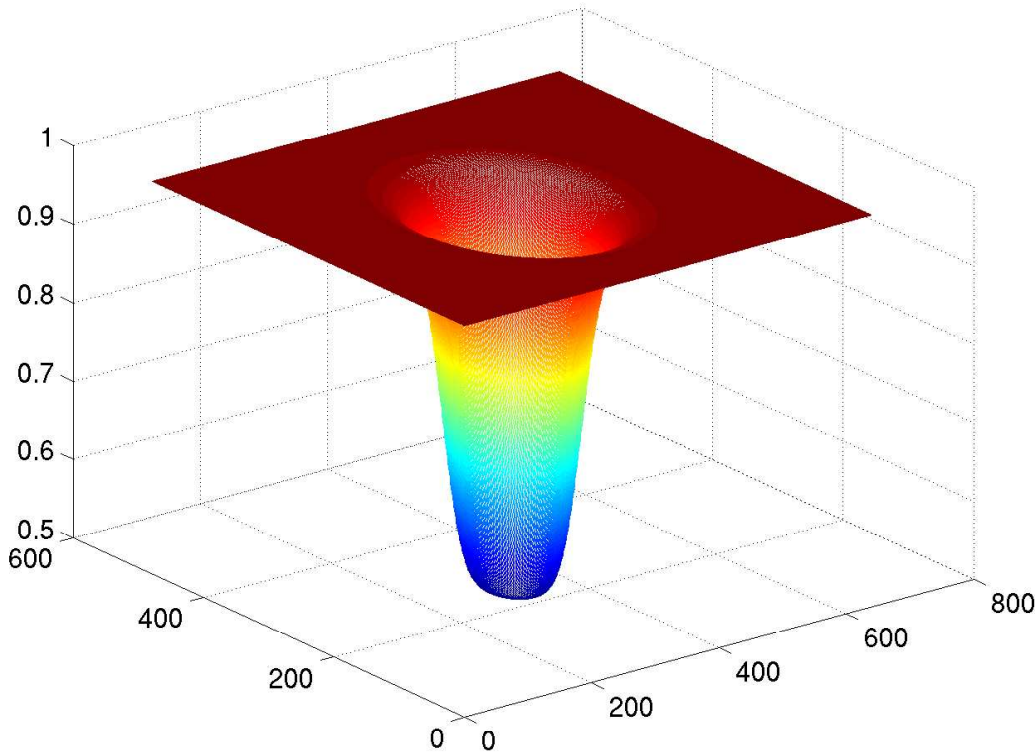
back to space $s(x, y) = \mathcal{F}^{-1}\{S(u, v)\}$ and back from \ln

$$g(x, y) = \exp(s(x, y))$$

So, we can suppress variations in illumination and enhance reflectance component.

Homomorphic filtering — filters

Homomorphic filter made by adaptation of Butterworth highpass



Remember: The filter is applied to $Z(u, v)$. Not to $F(u, v)$!

Homomorphic filtering — results



Original image.

Homomorphic filtering — results



Original image.



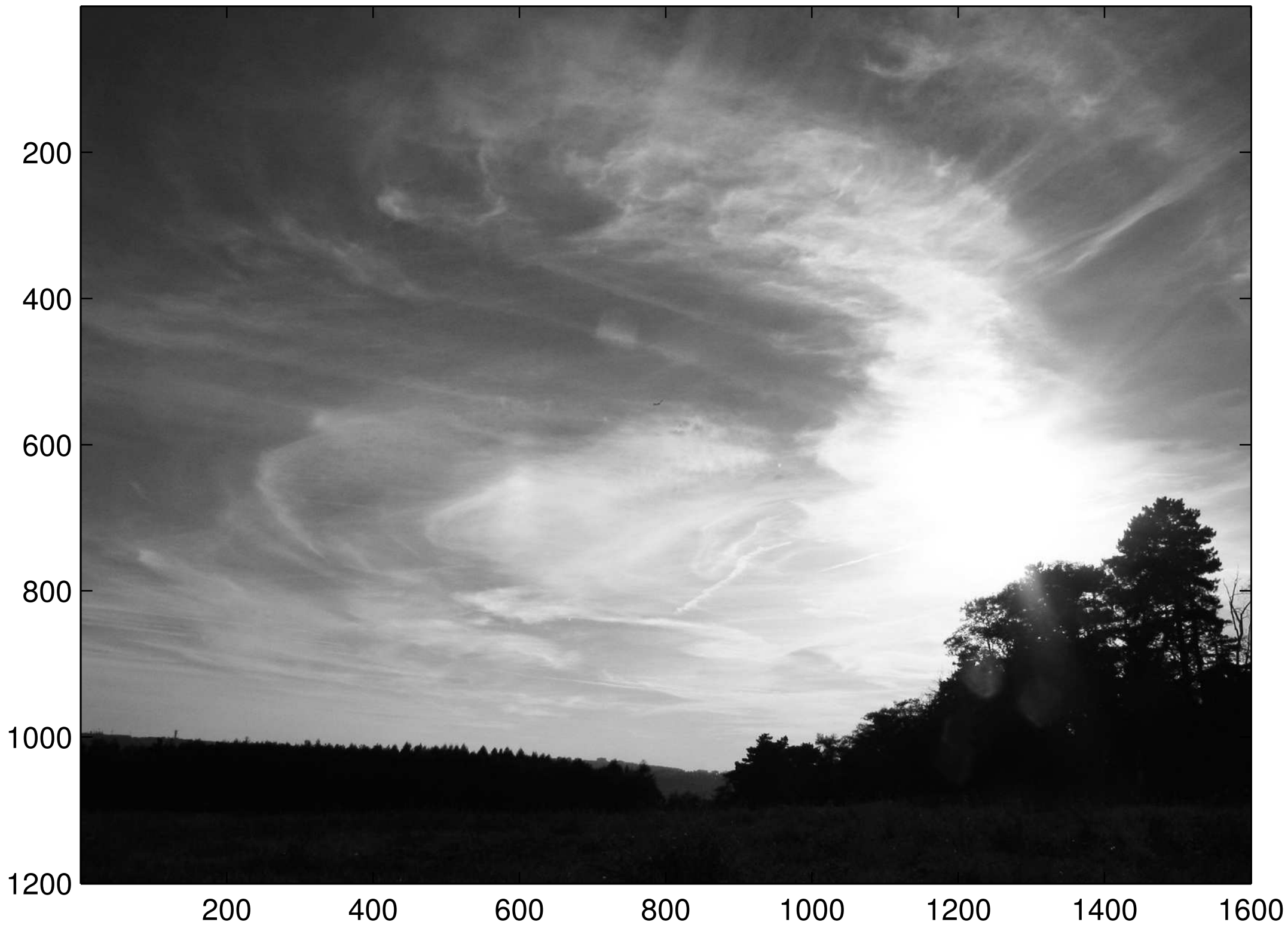
Filtered image.

Readings

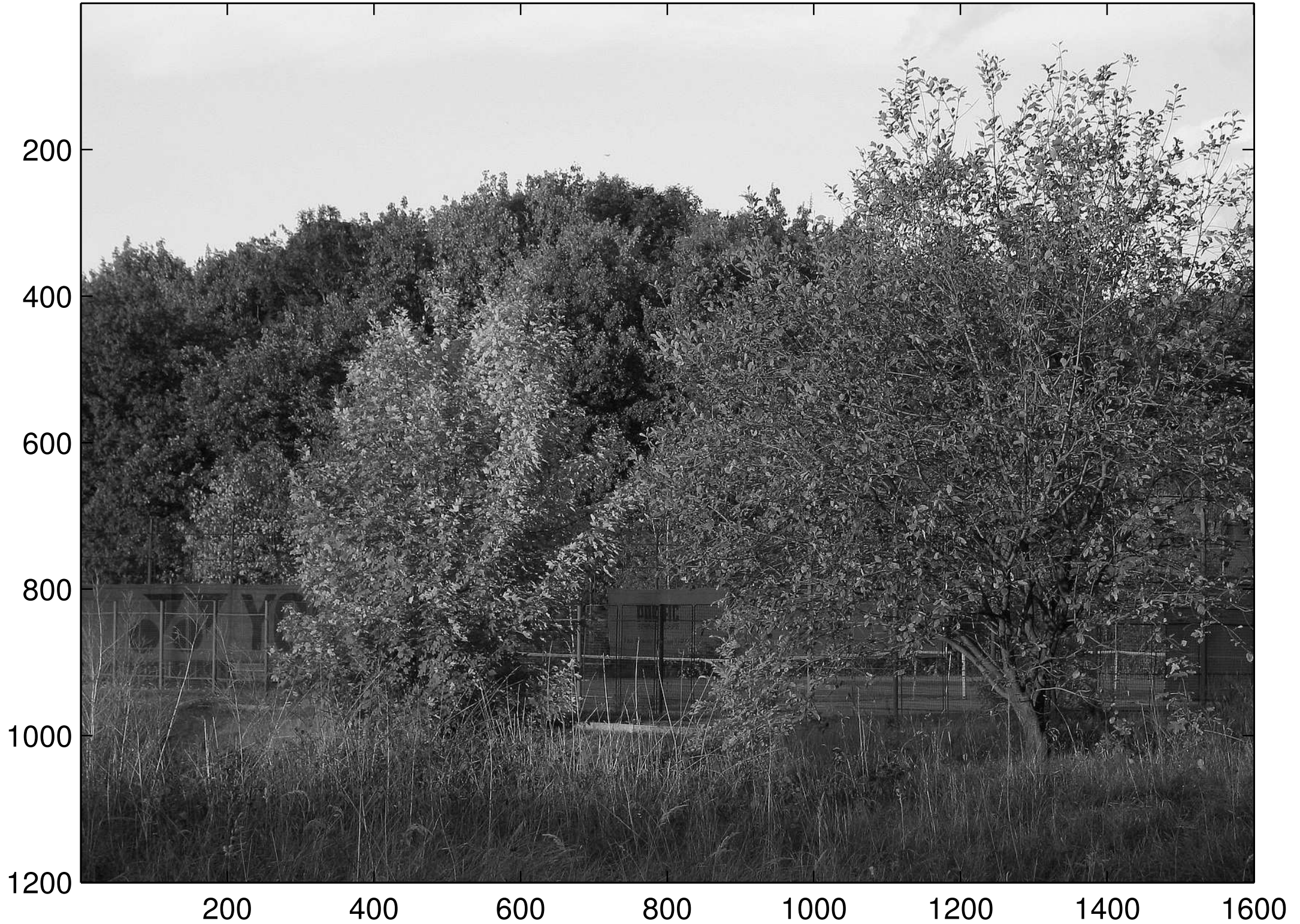
Many images and demos were made by using the codes from chapter 3 of the book [2]. Slightly non-traditional and very valuable insight into Fourier imaging brings the book [1]

- [1] Ronald N. Bracewell. *Fourier analysis and imaging*. Kluwer Academic/Plenum Publishers, New York, USA, 2003.
- [2] Tomáš Svoboda, Jan Kybic, and Václav Hlaváč. *Image Processing, Analysis and Machine Vision. A MATLAB Companion*. Thomson, 2007. Accompanying www site <http://visionbook.felk.cvut.cz>.

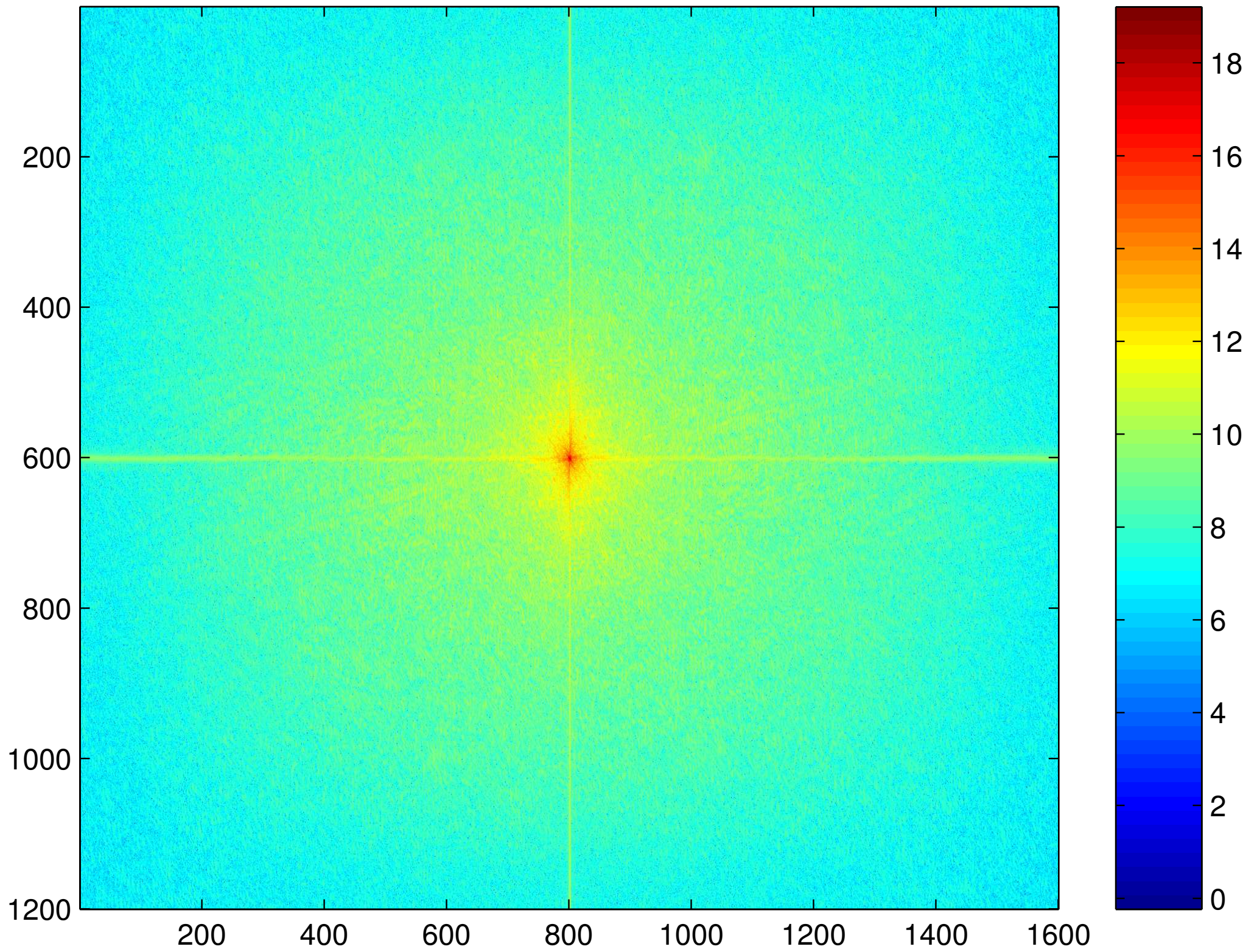
Low frequency image in gray scales



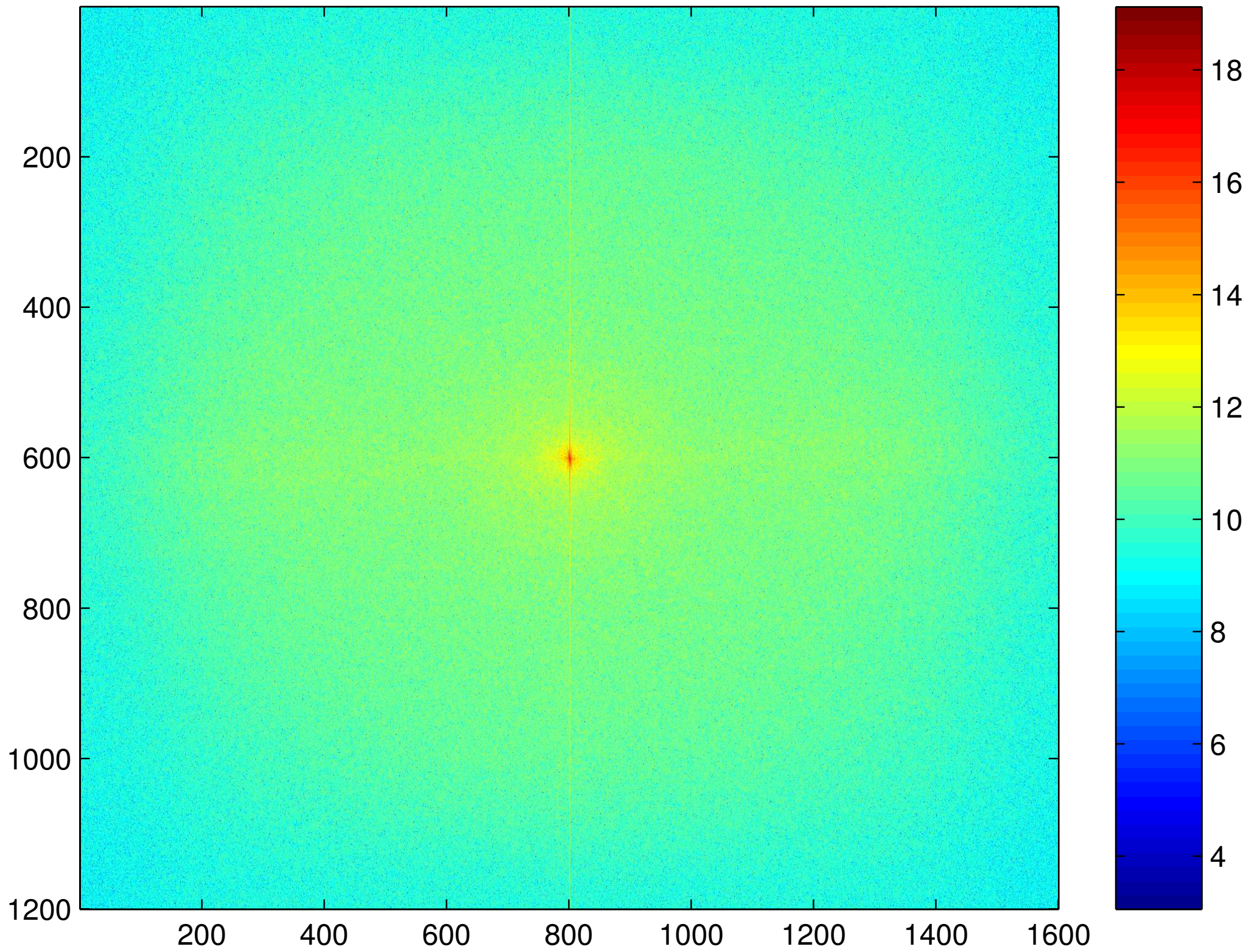
Hi frequency image in gray scales



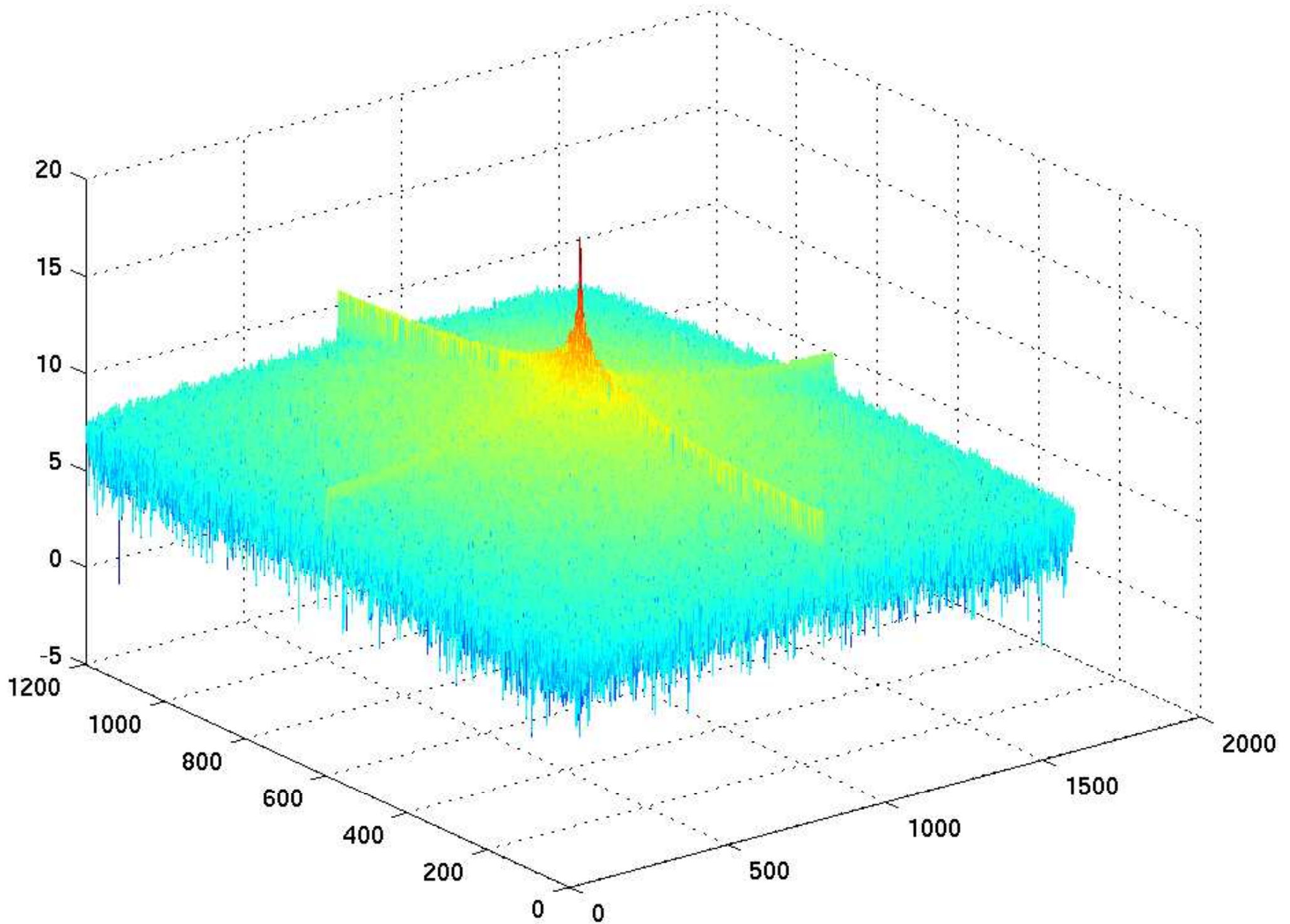
Low frequency image: $\log(\text{abs}(\text{FFT2}(\text{im})))$



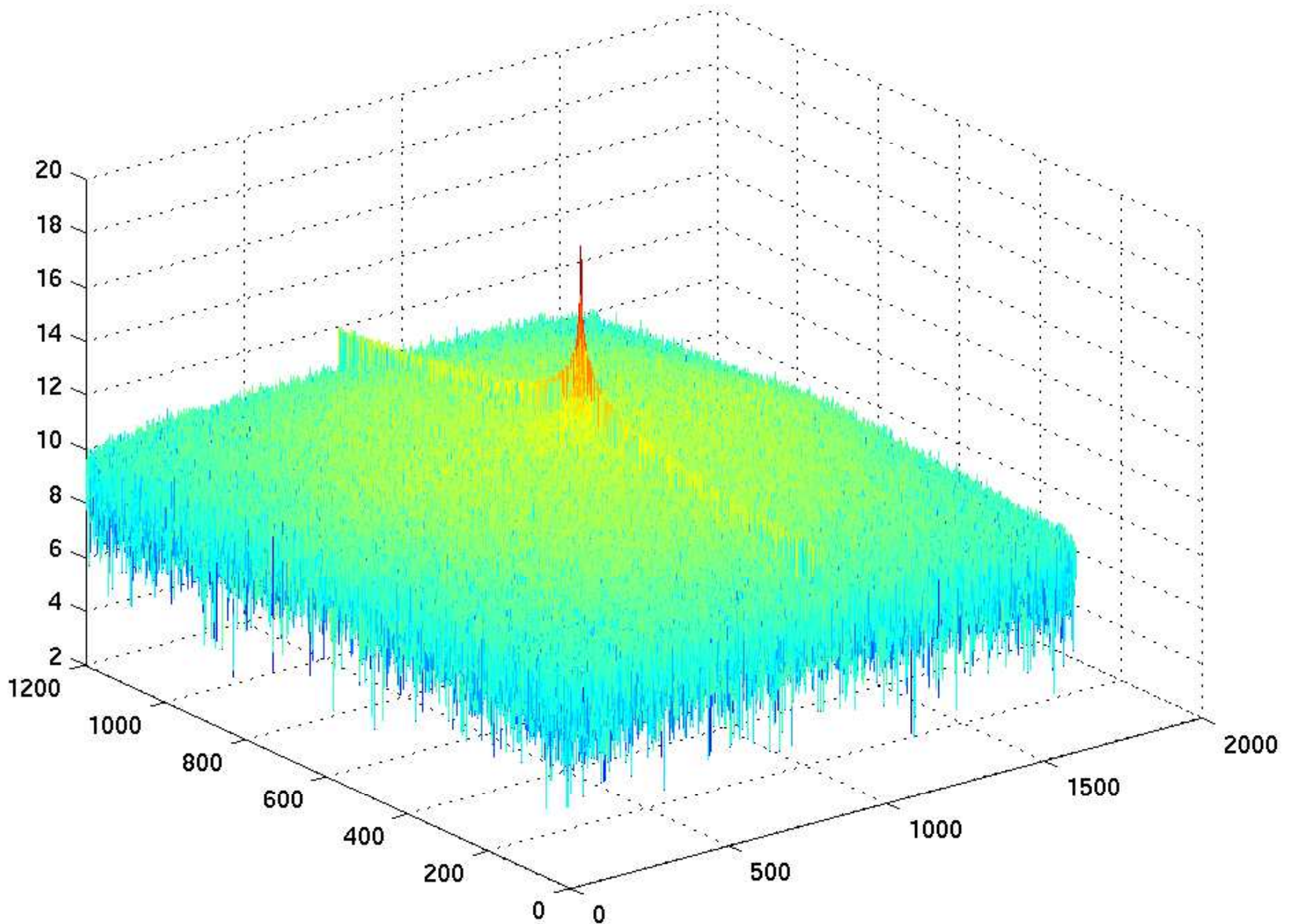
Hi frequency image: $\log(\text{abs}(\text{FFT2}(\text{im})))$



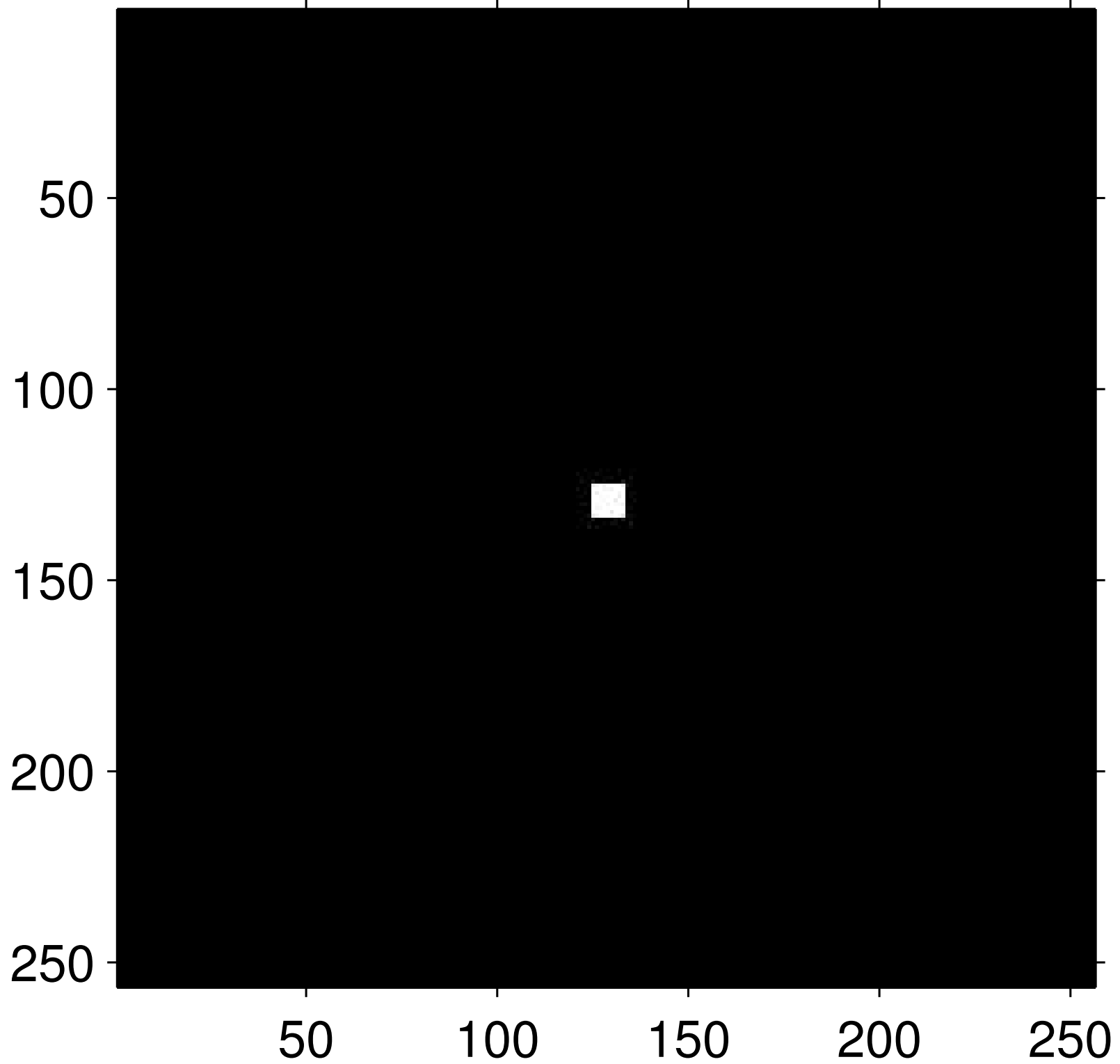
Low frequency image: mesh print of $\log(\text{abs}(\text{FFT2}(\text{im})))$



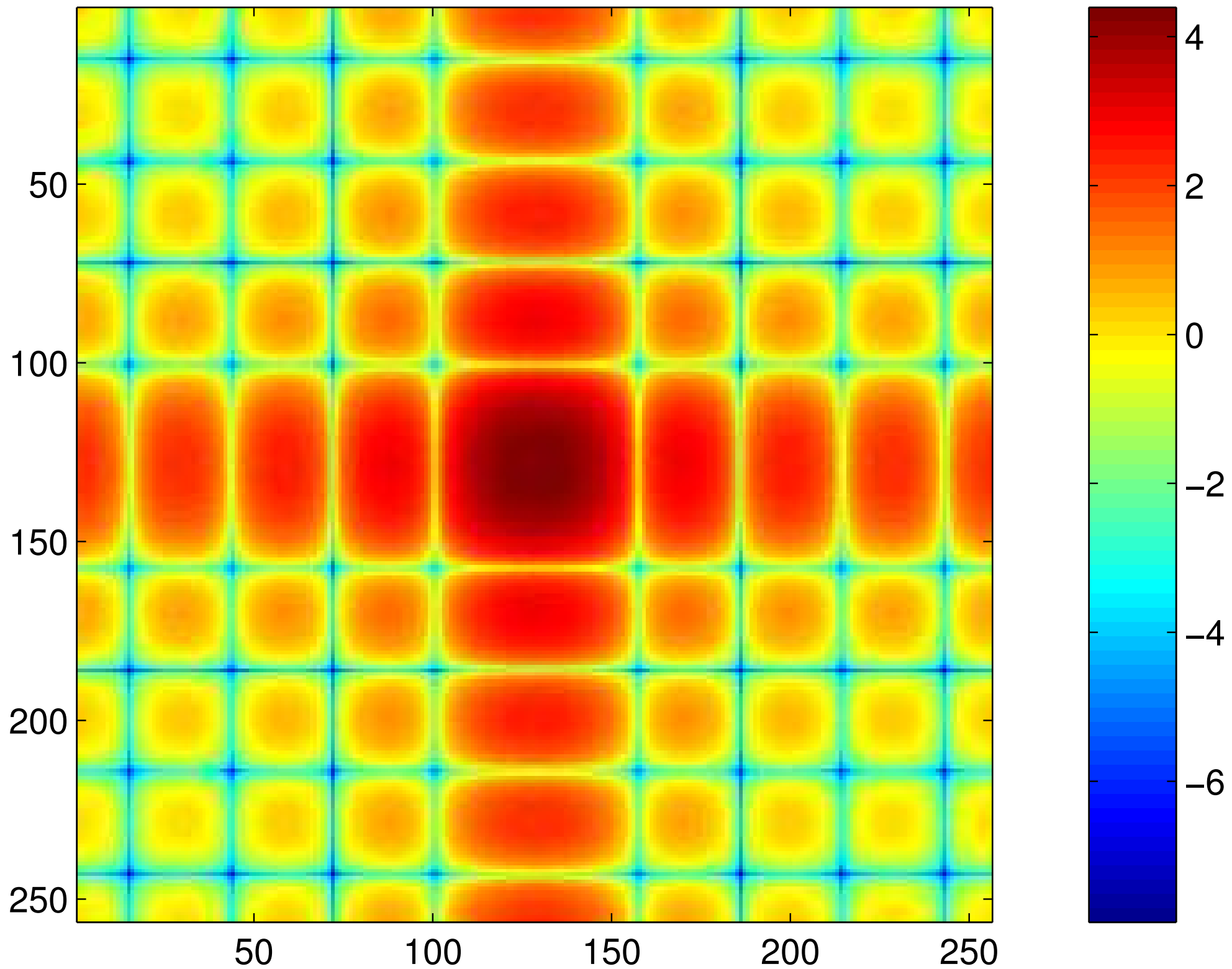
Hi frequency image: mesh print of $\log(\text{abs}(\text{FFT2}(\text{im})))$



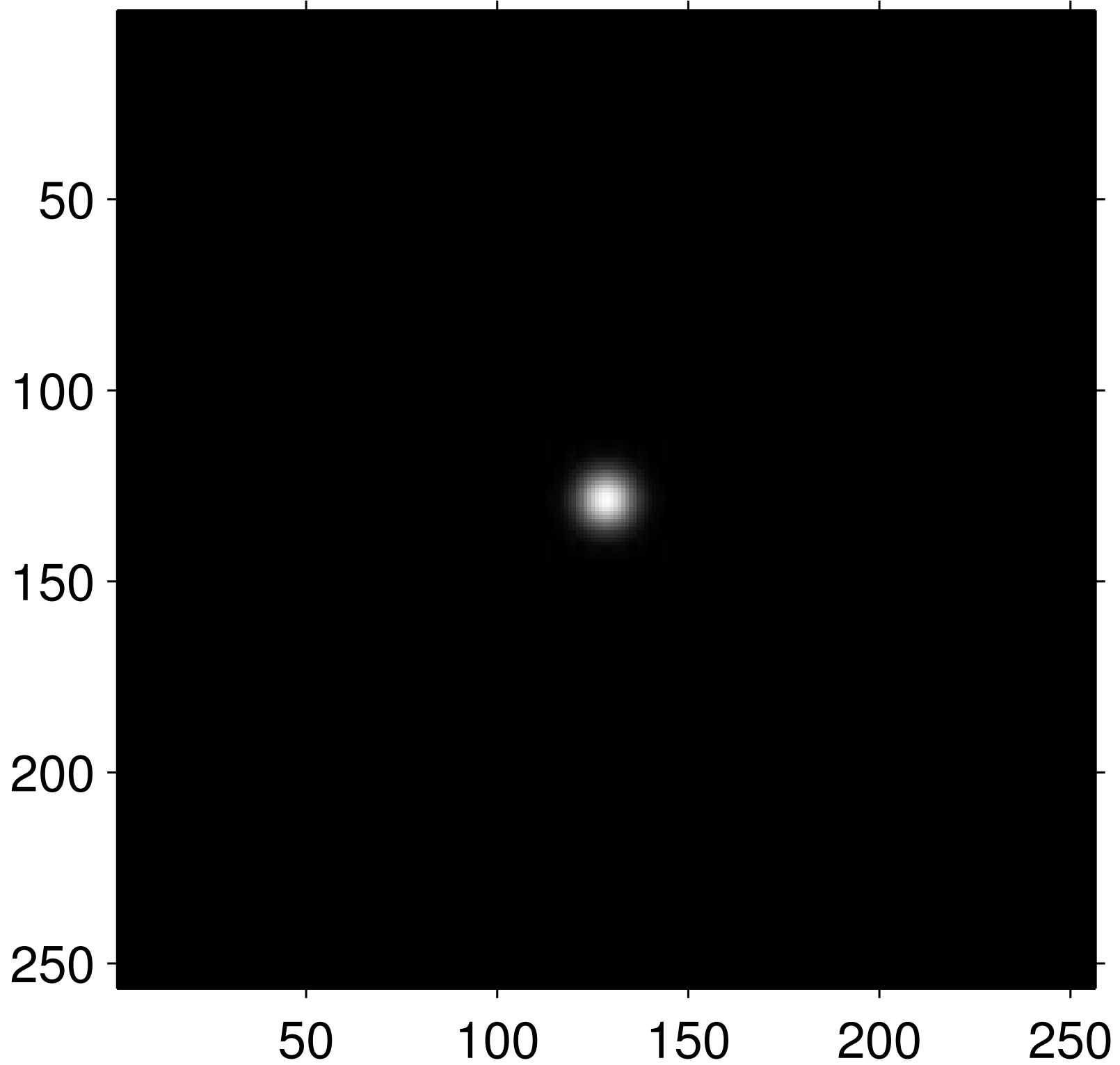
image



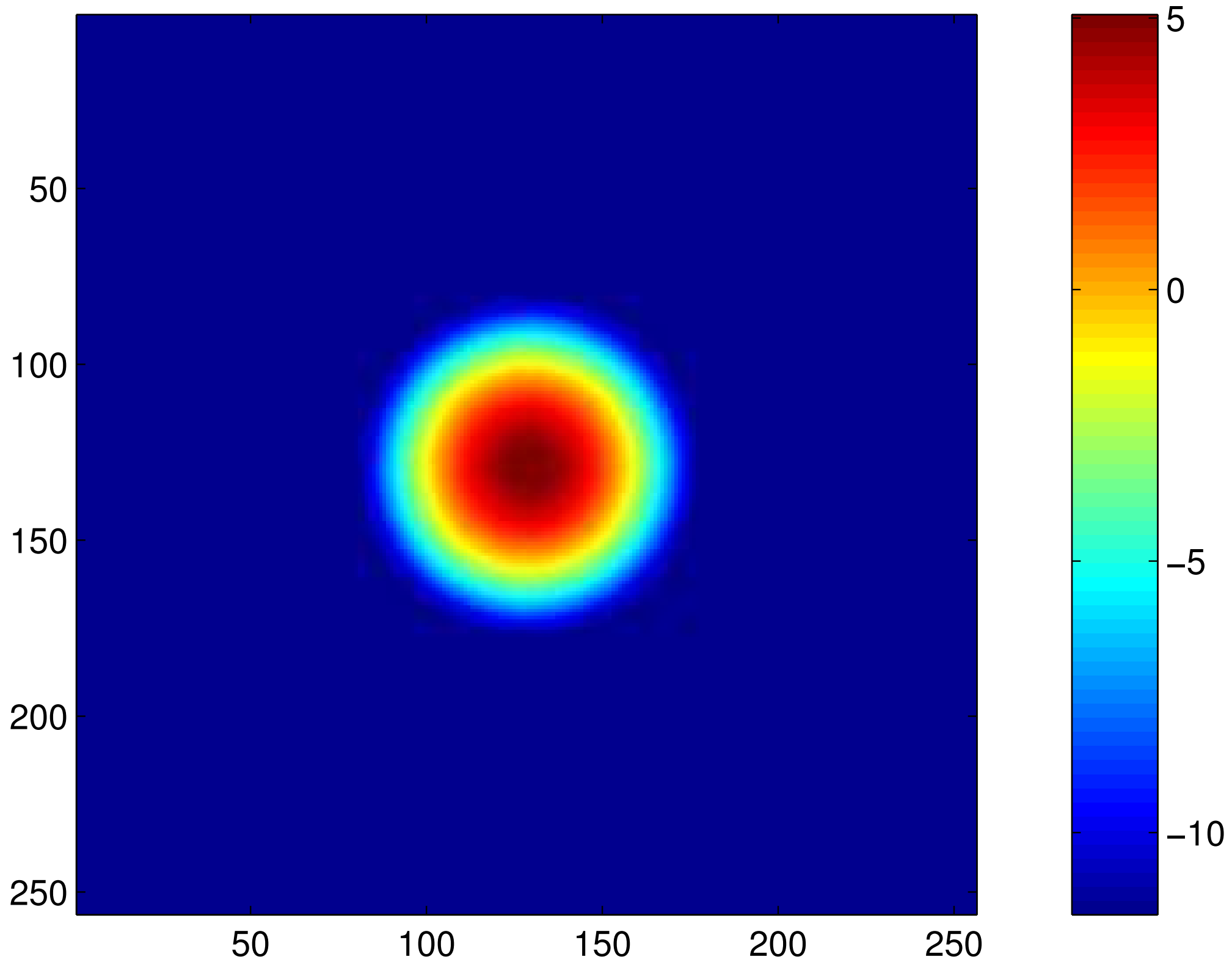
$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$



image



$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$



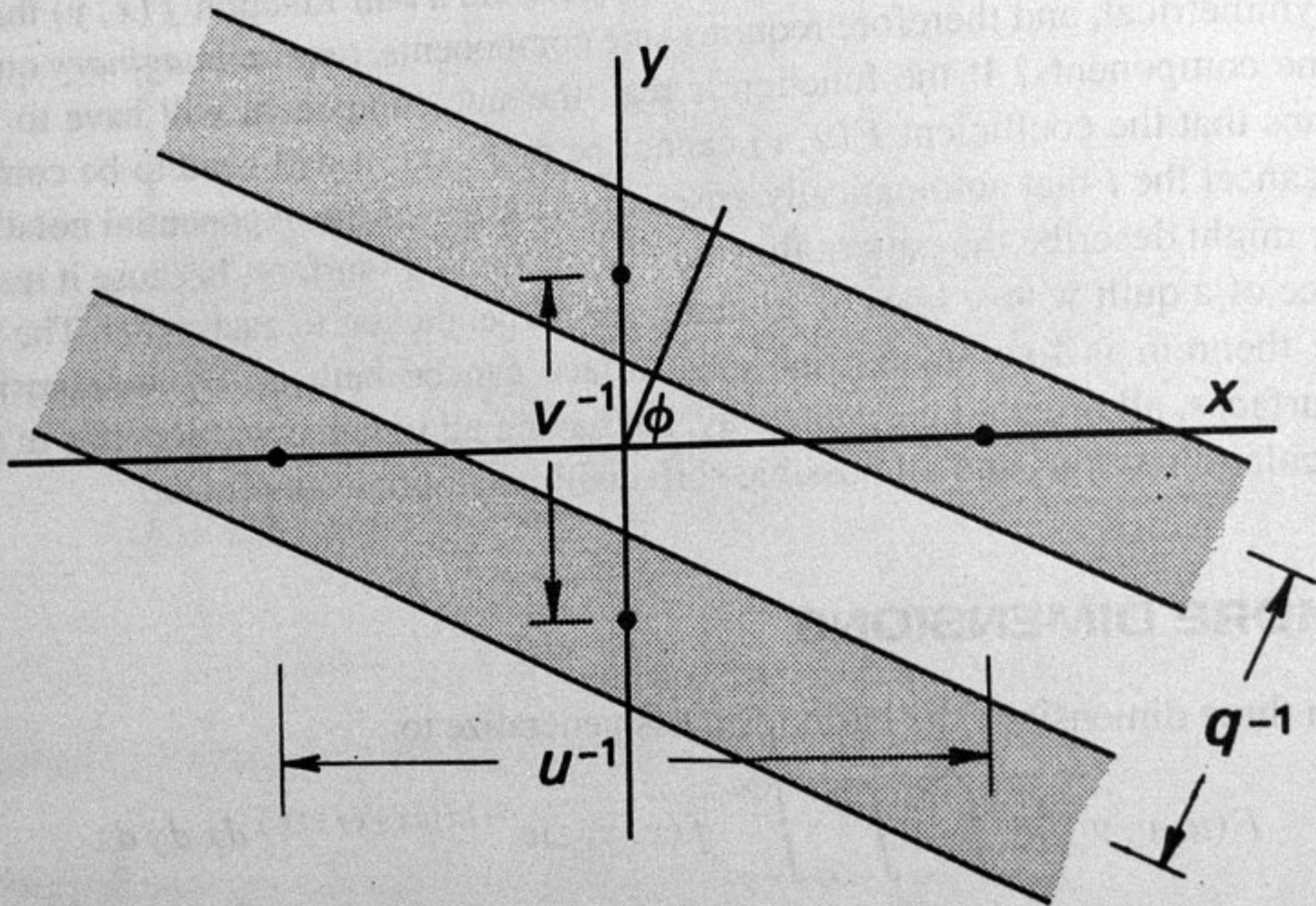


Figure 4-2 A “corrugation” $\cos[2\pi(ux + vy)]$. The shaded zones are negative. The spatial frequency in the x -direction is u (in the y -direction, v); the spatial frequency is q .

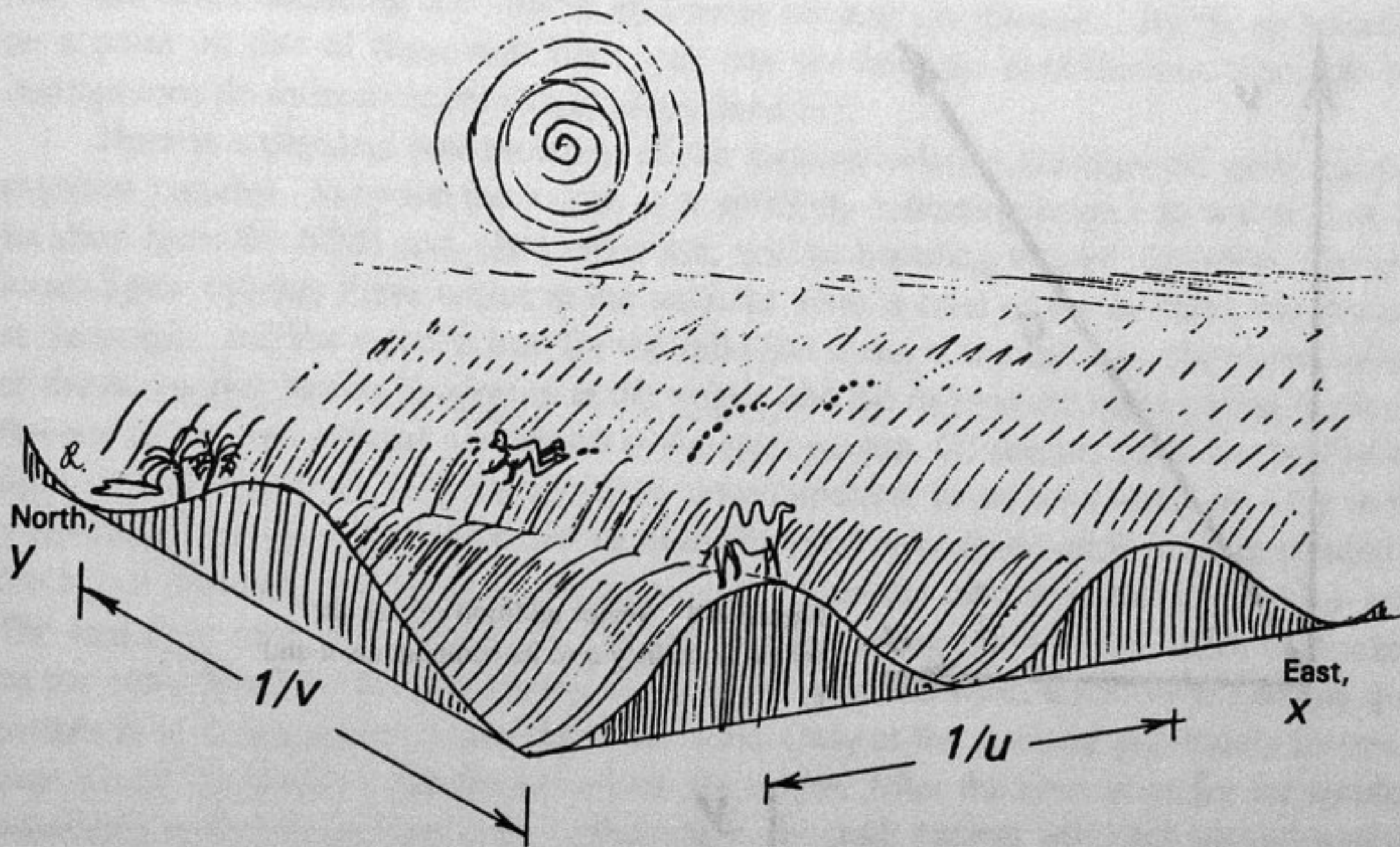
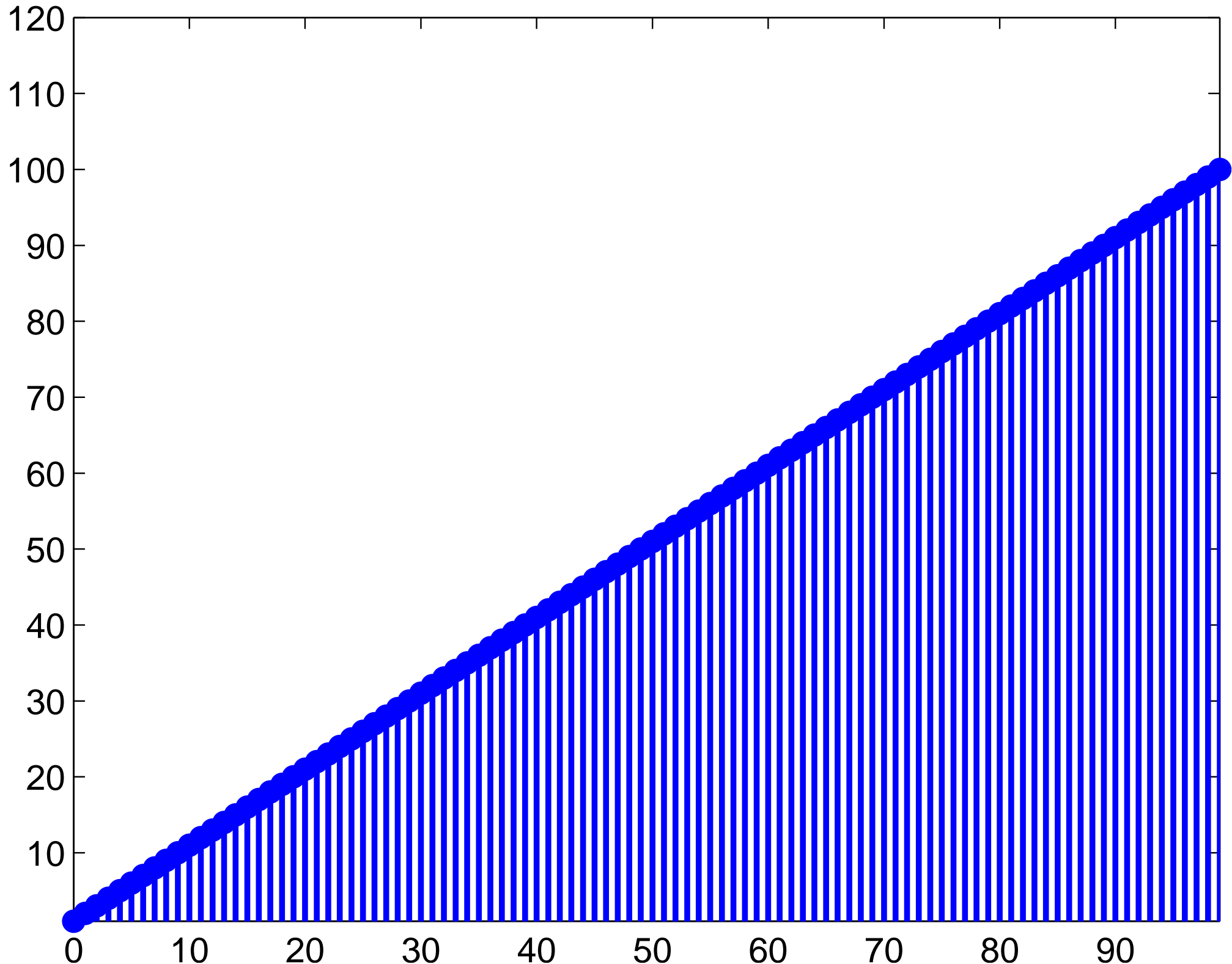
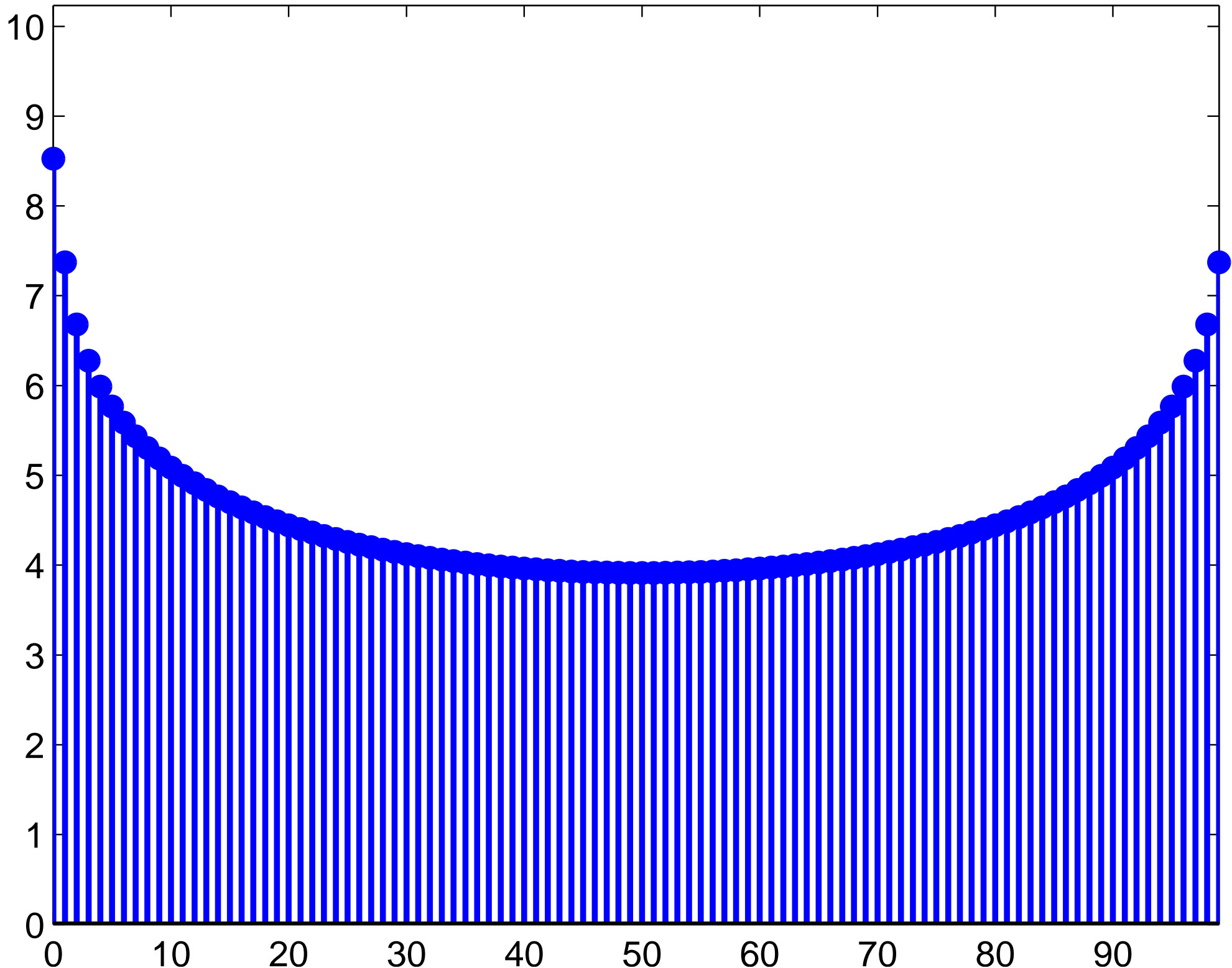


Figure 4-3 The x -component of spatial frequency is u , measured in cycles per unit of x . The spatial period, or crest-to-crest distance traveling east, is u^{-1} (see camel). The spatial frequency q is measured in the direction of hardest going (see man).

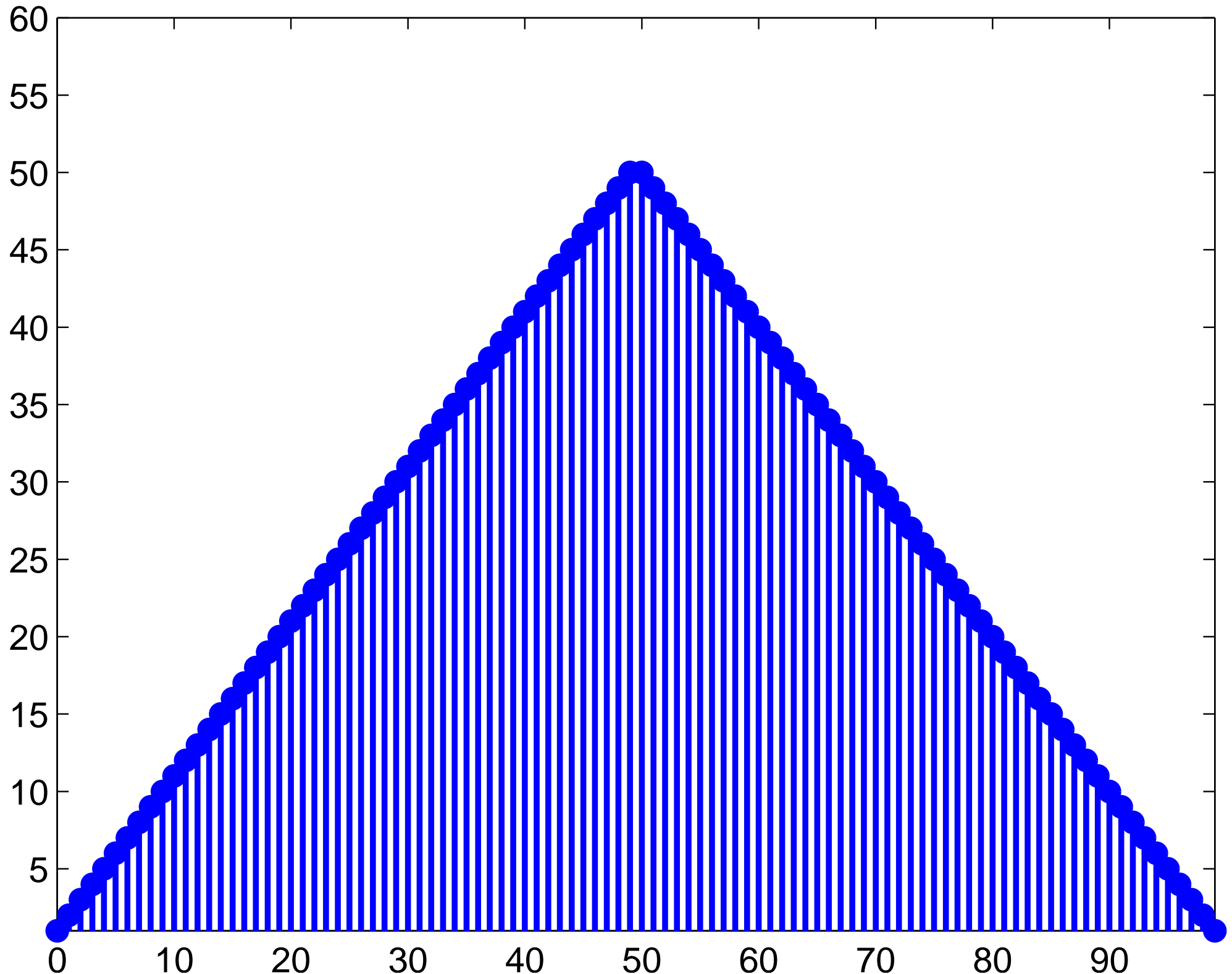
$f(x)$



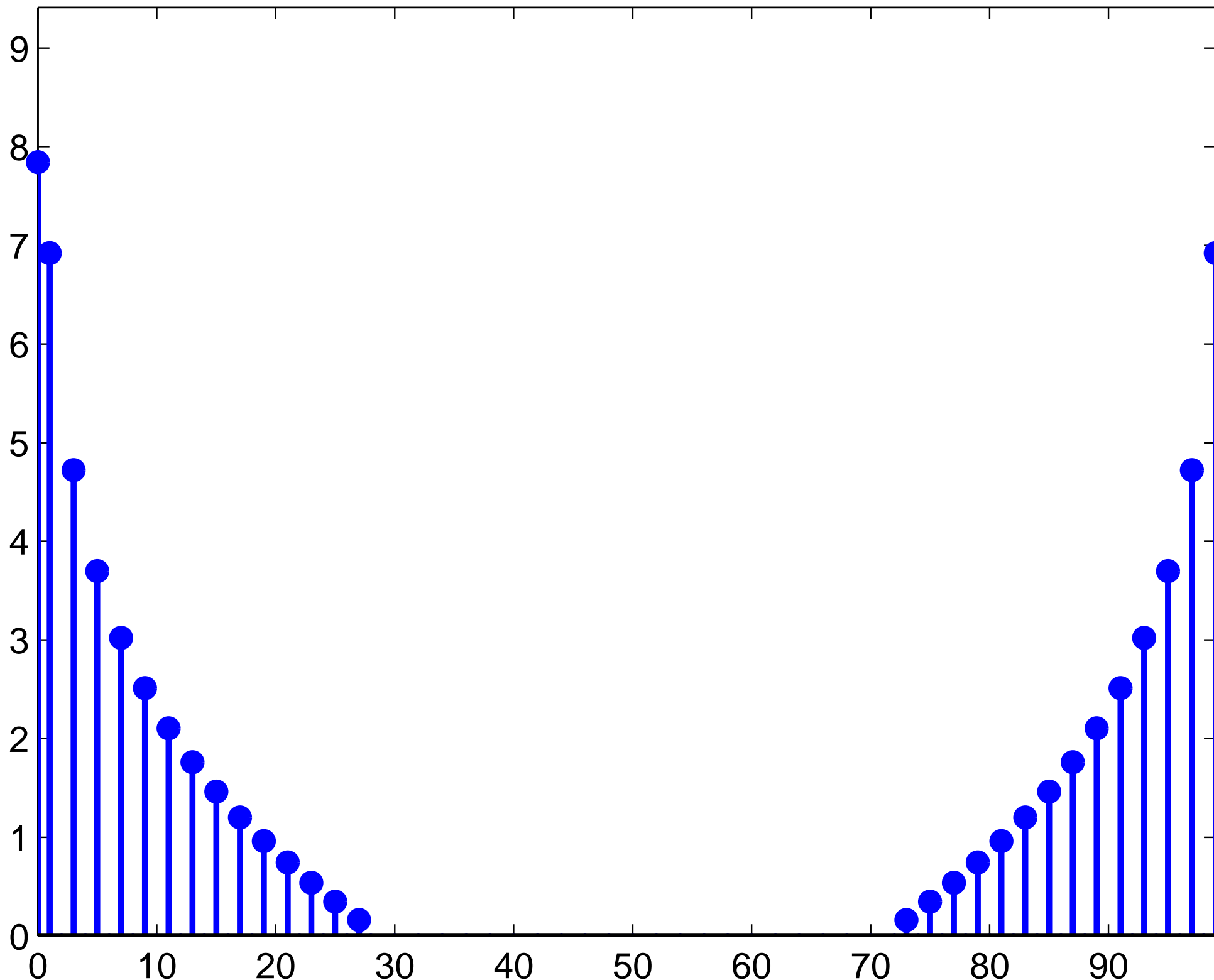
log of power spectrum

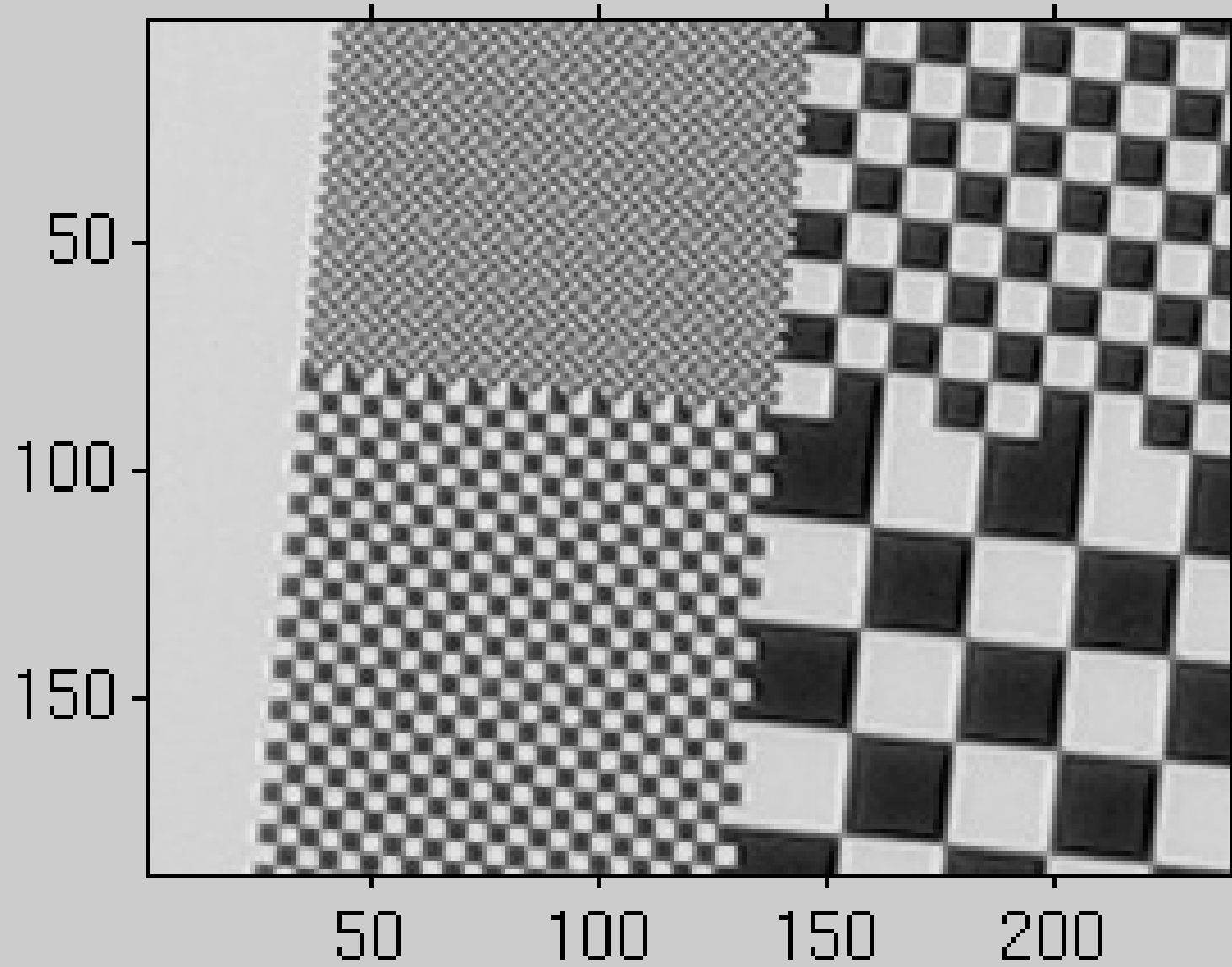


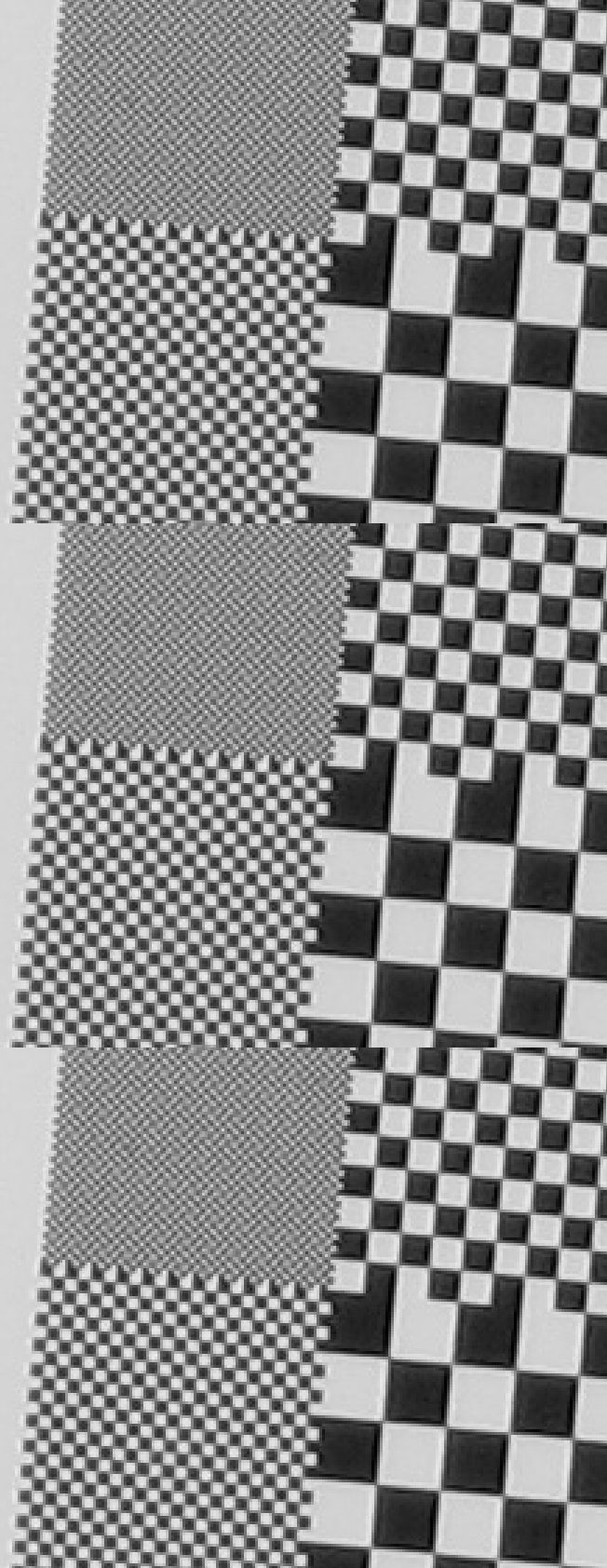
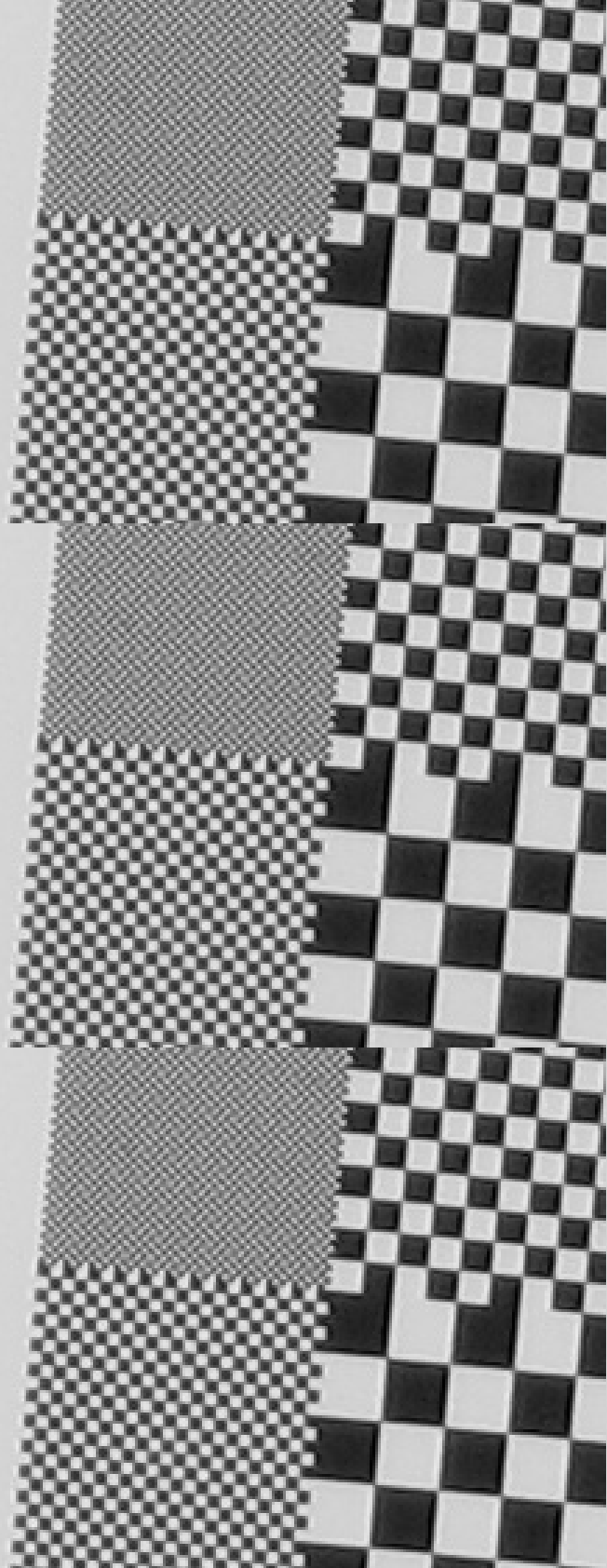
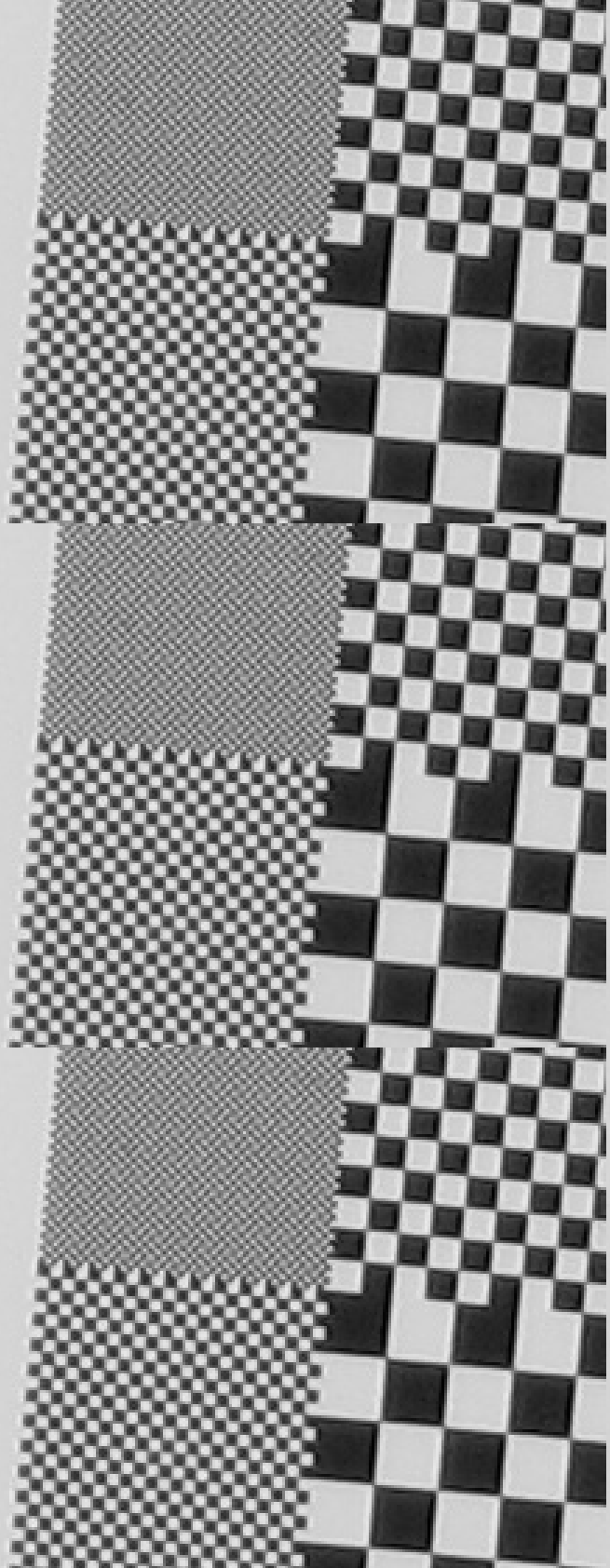
$f(x)$



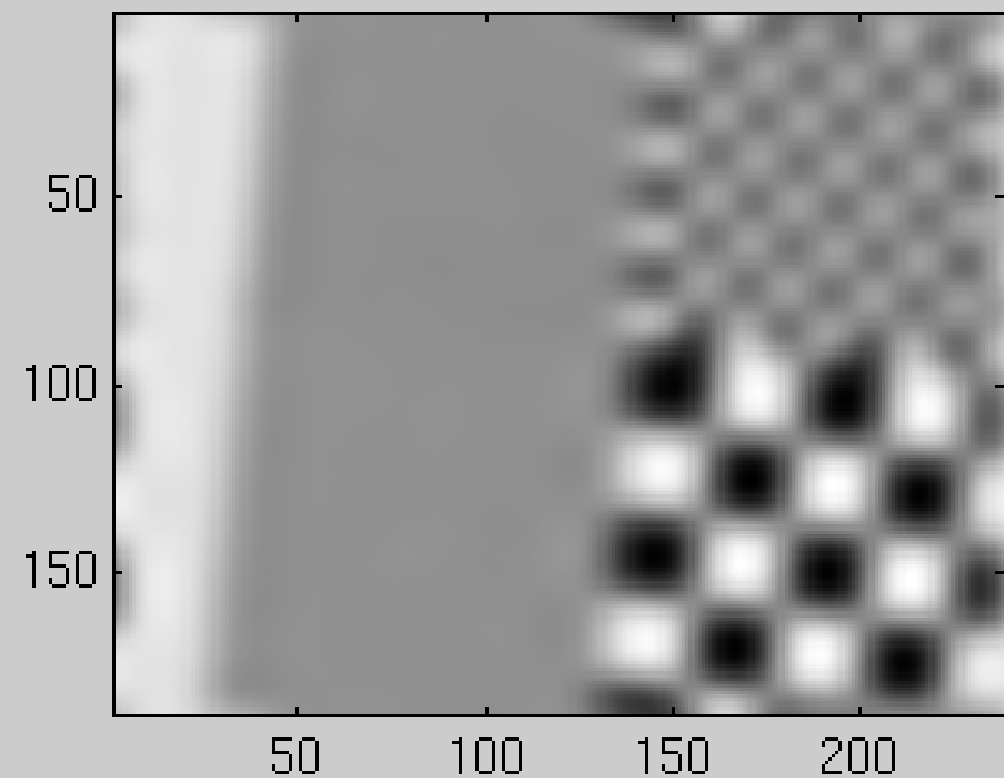
log of power spectrum



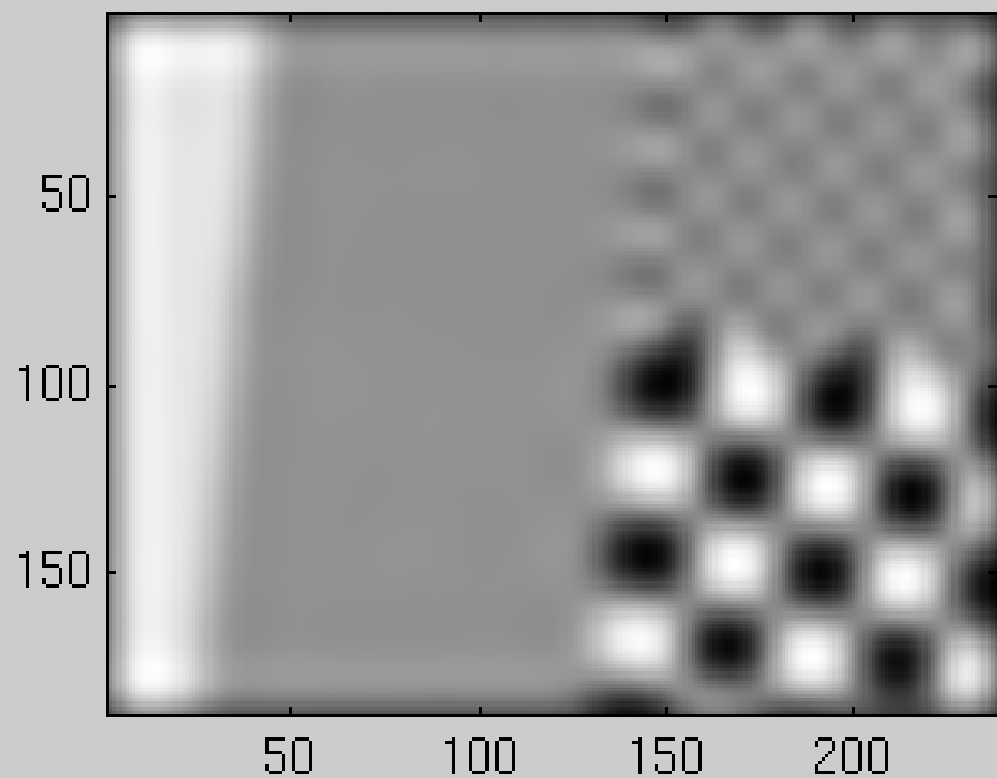




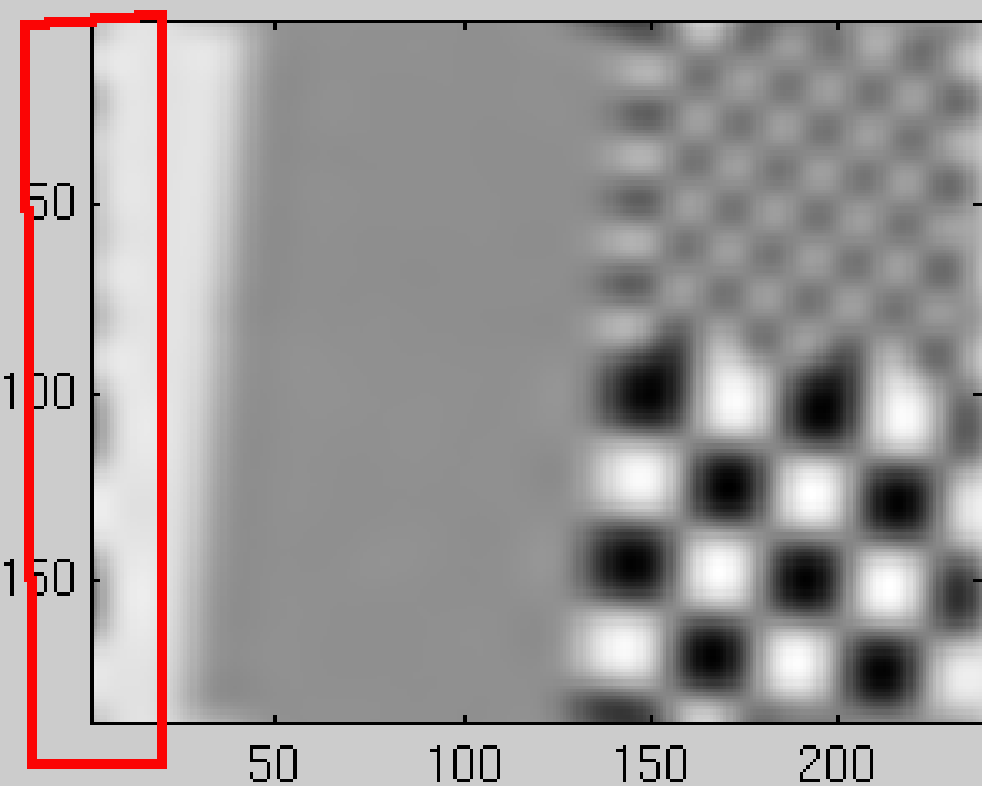
Low-pass filtered image (without zero padding)



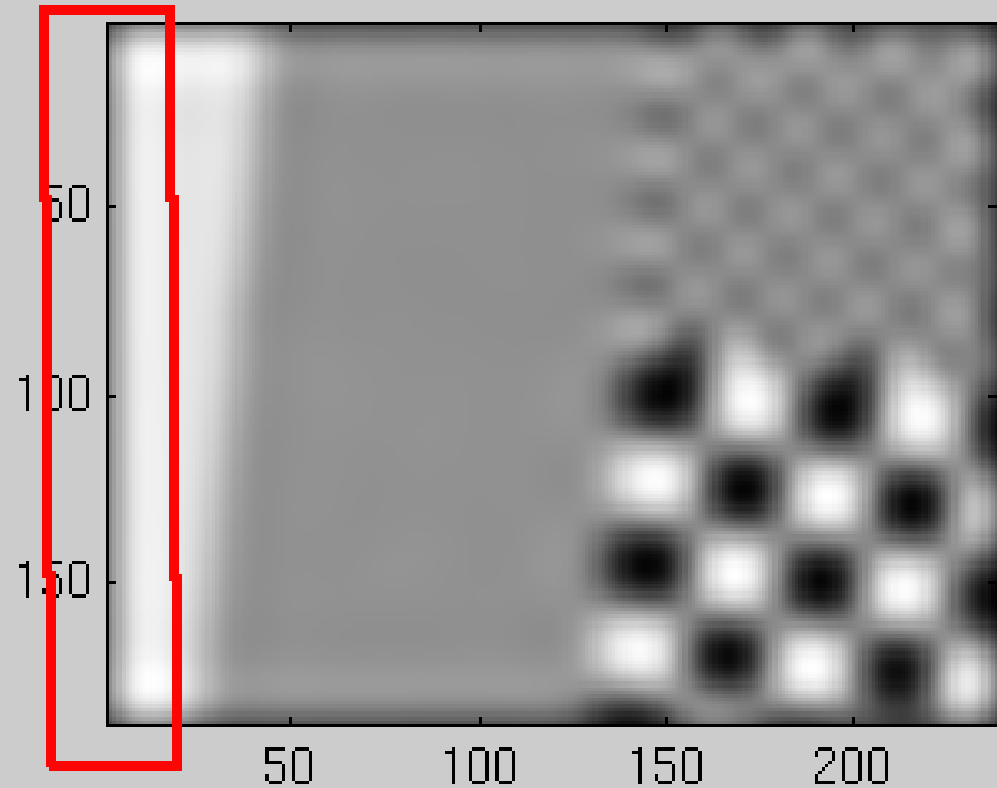
Low-pass filtered image (zero padding applied)



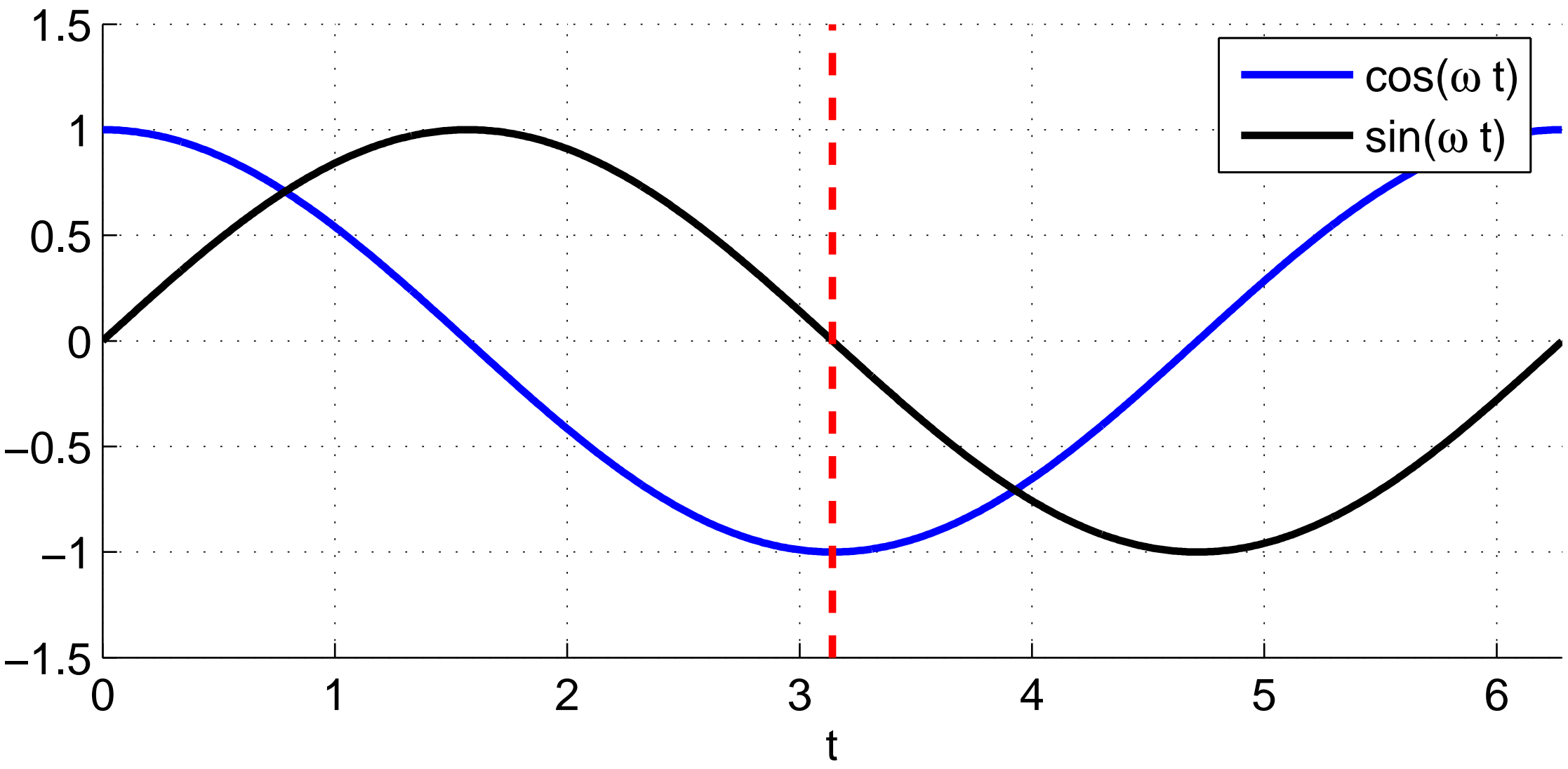
Low-pass filtered image (without zero padding)



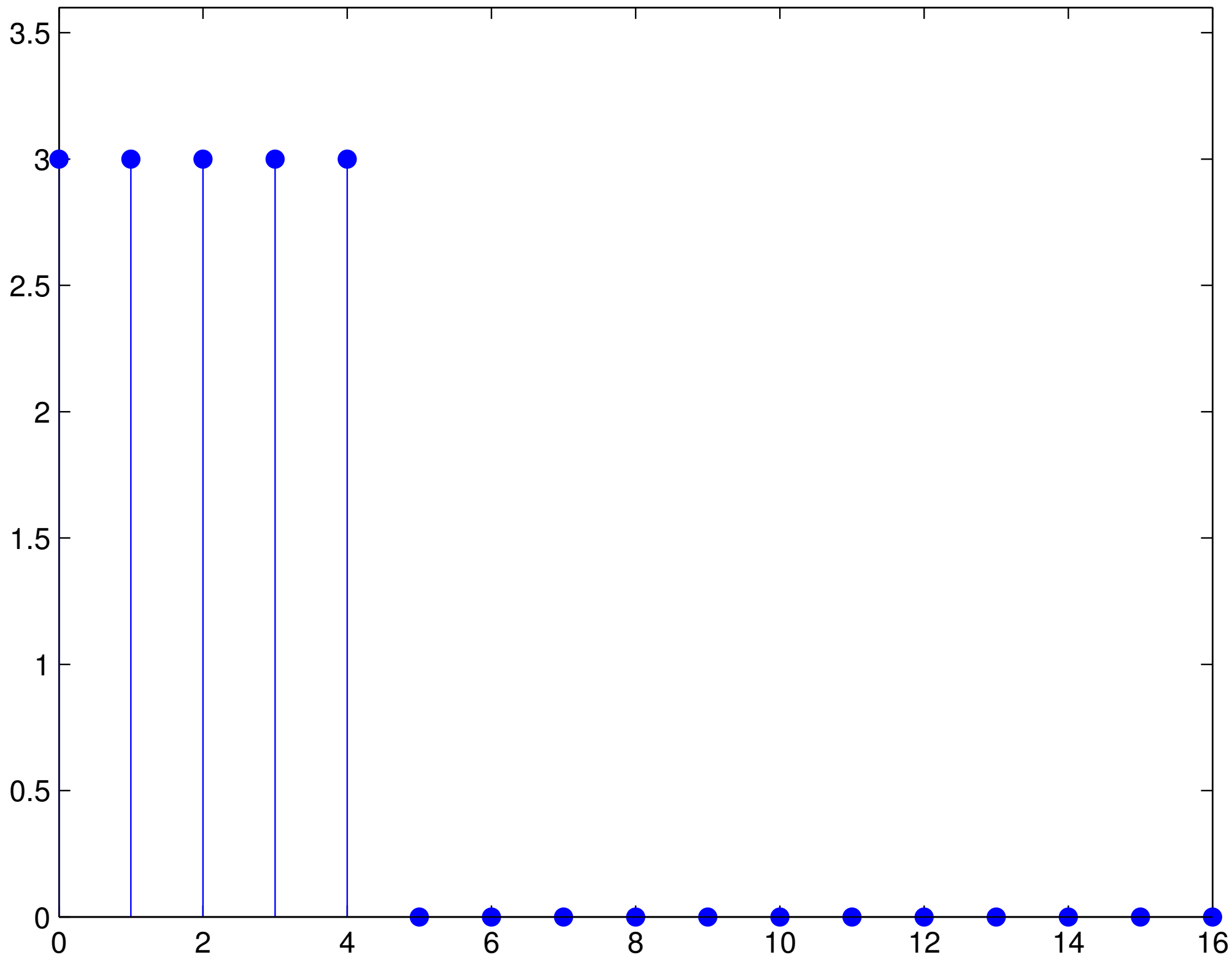
Low-pass filtered image (zero padding applied)



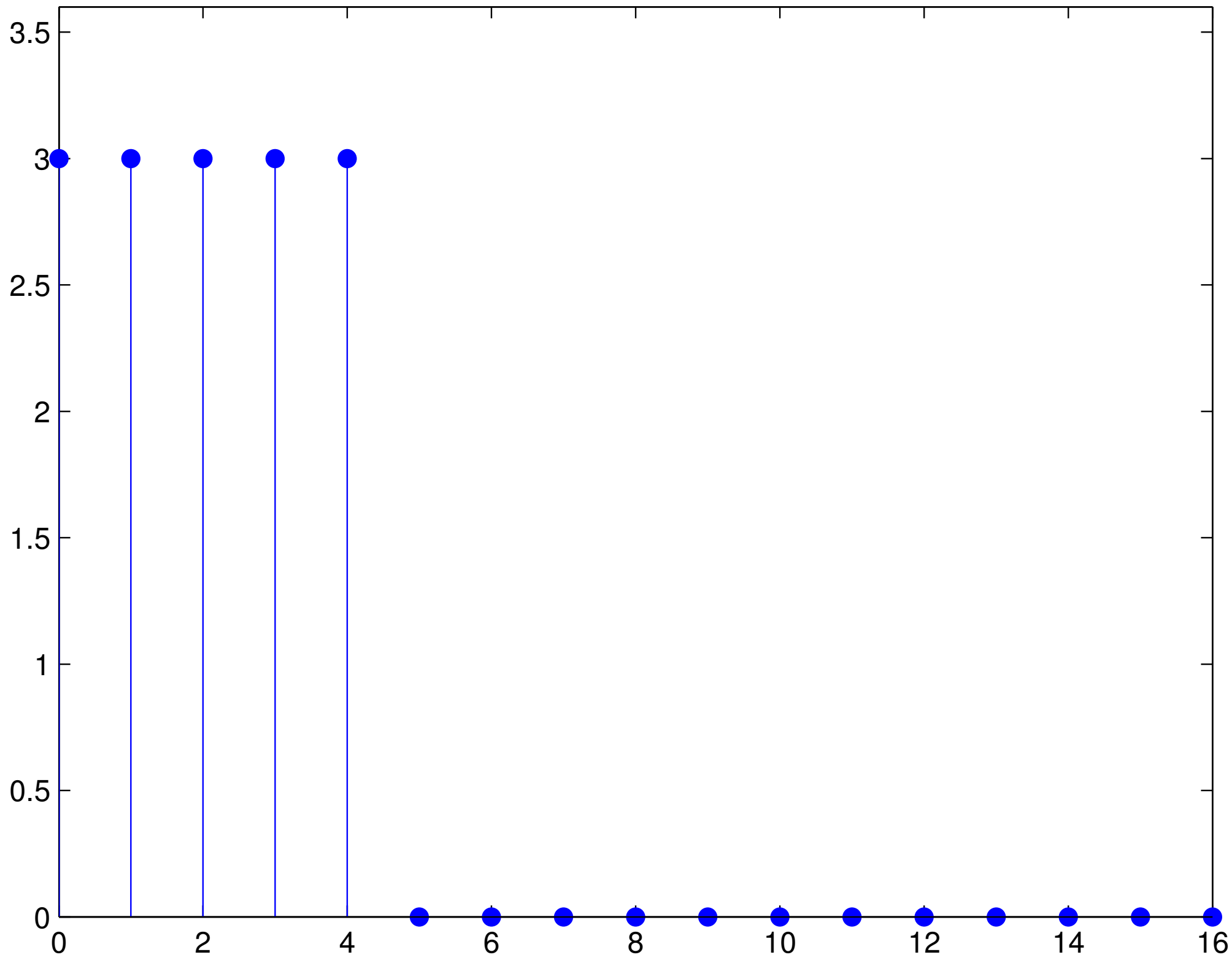
$\omega = 1.00$



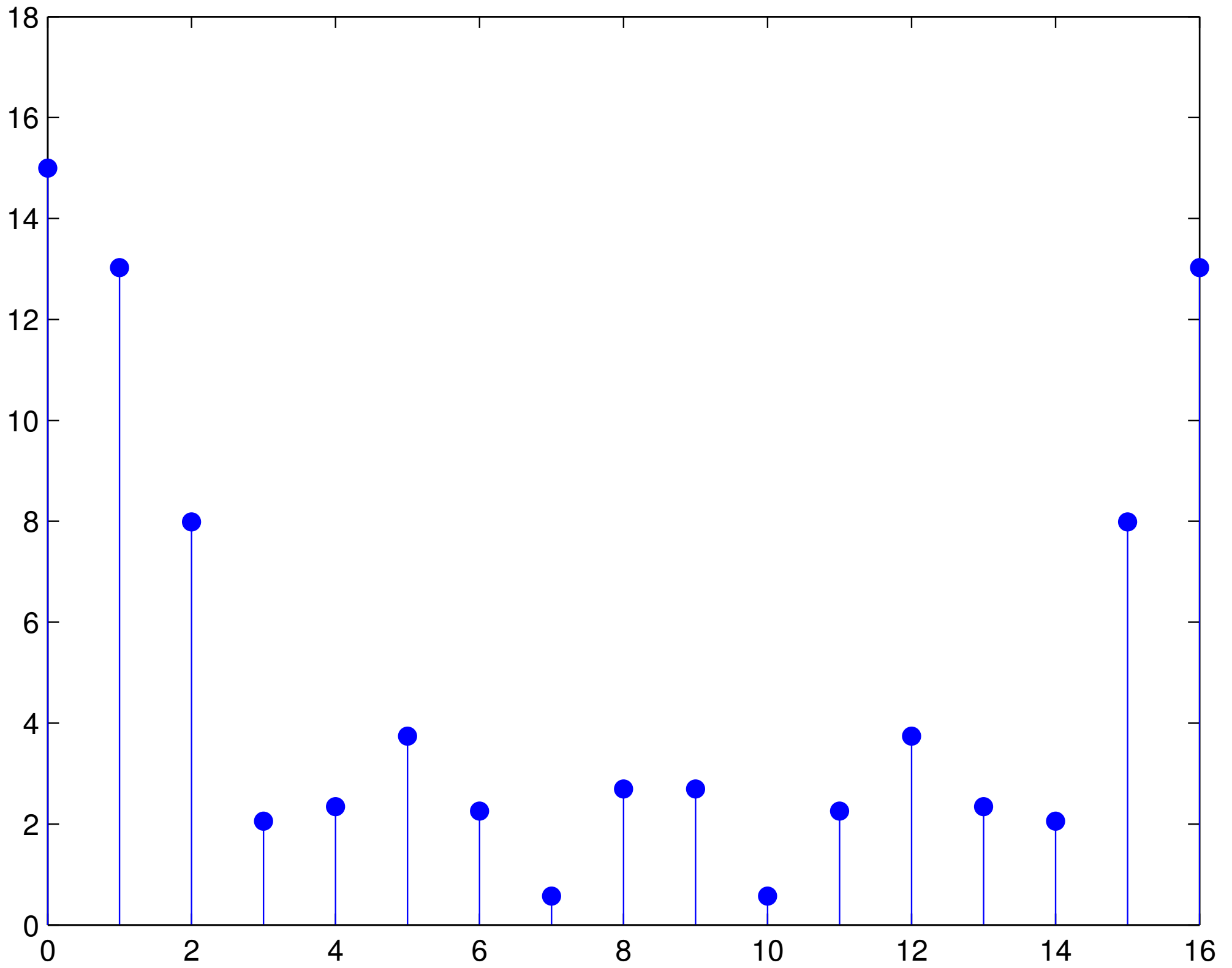
$f(x)$



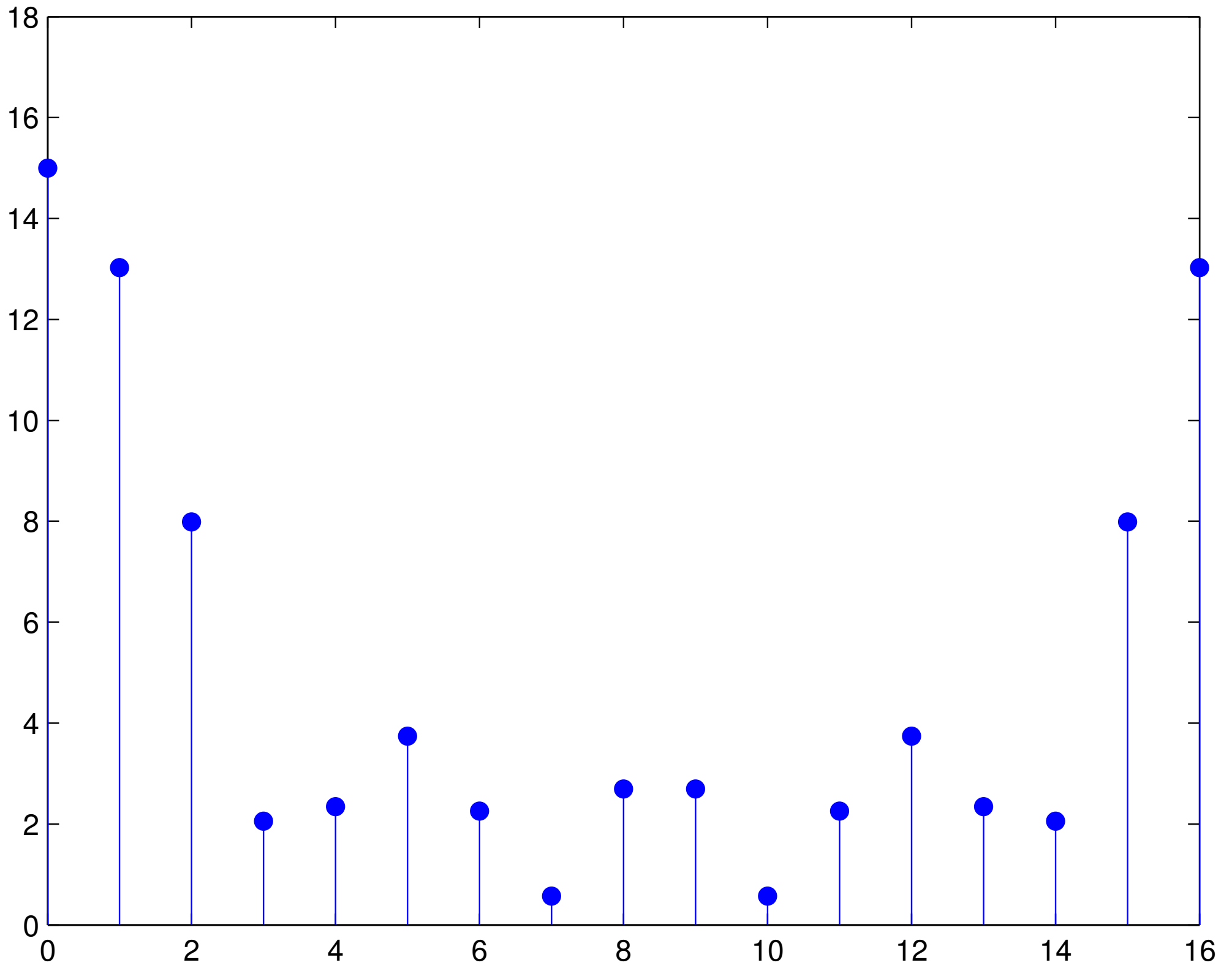
$f(x)$



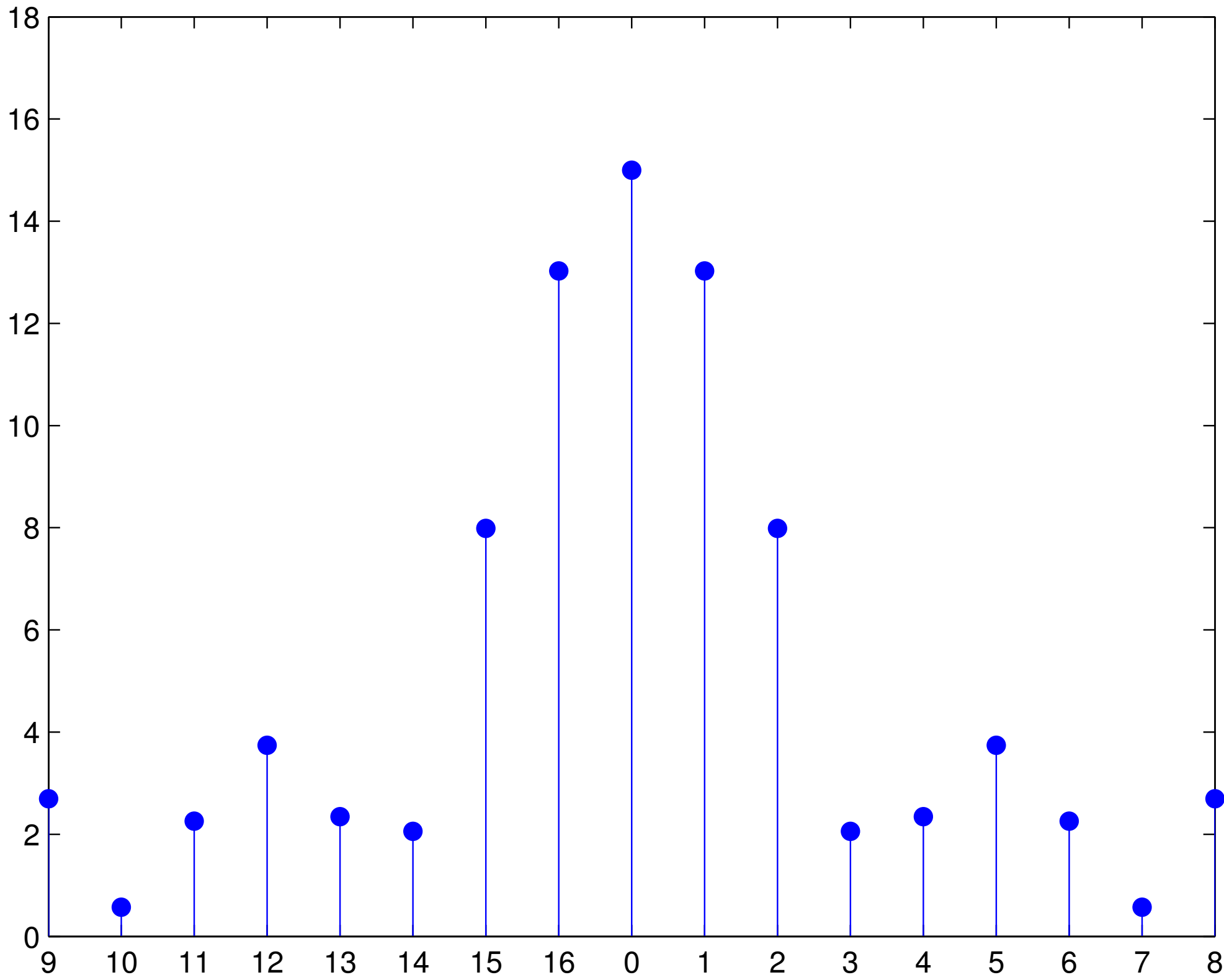
abs(fft(f))



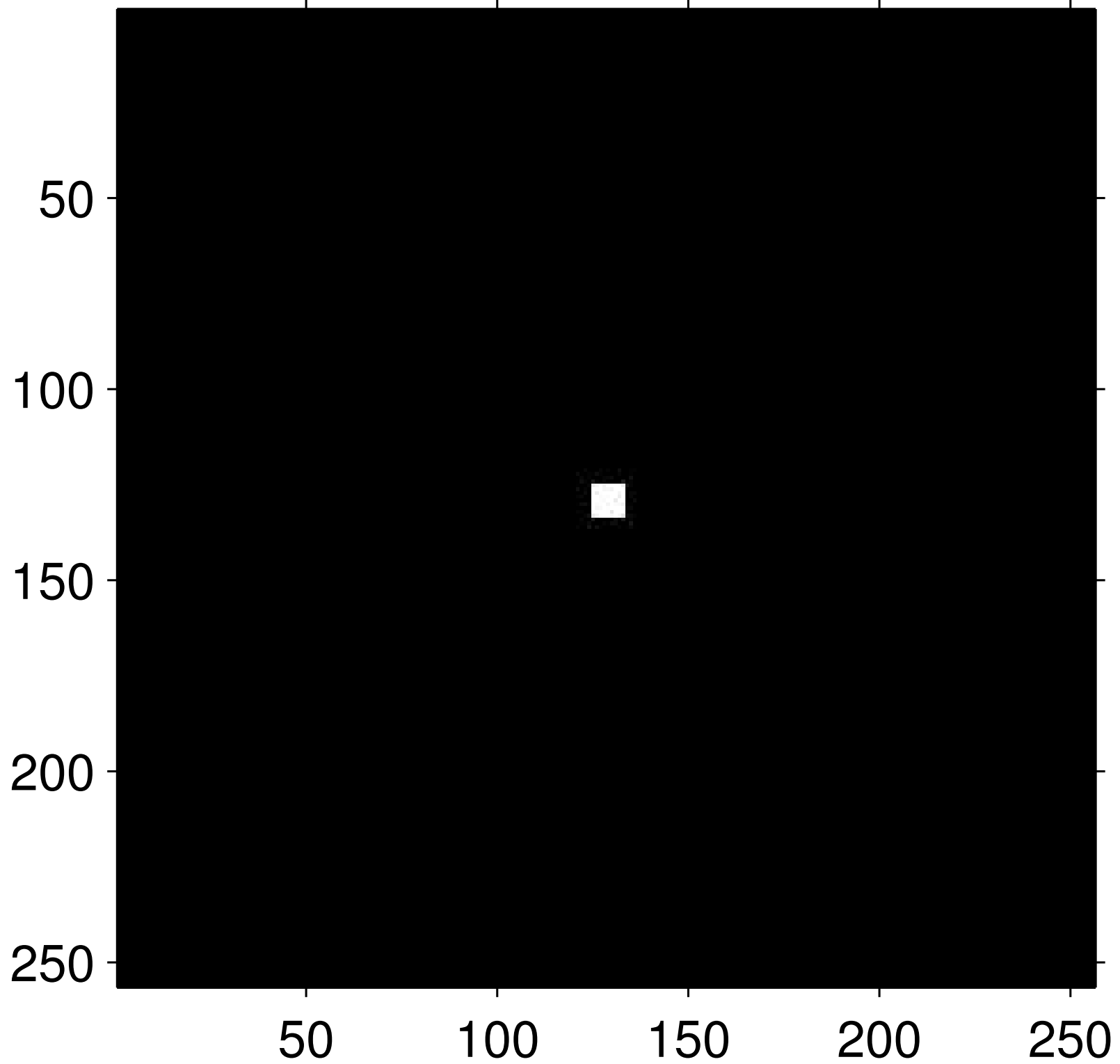
abs(fft(f))



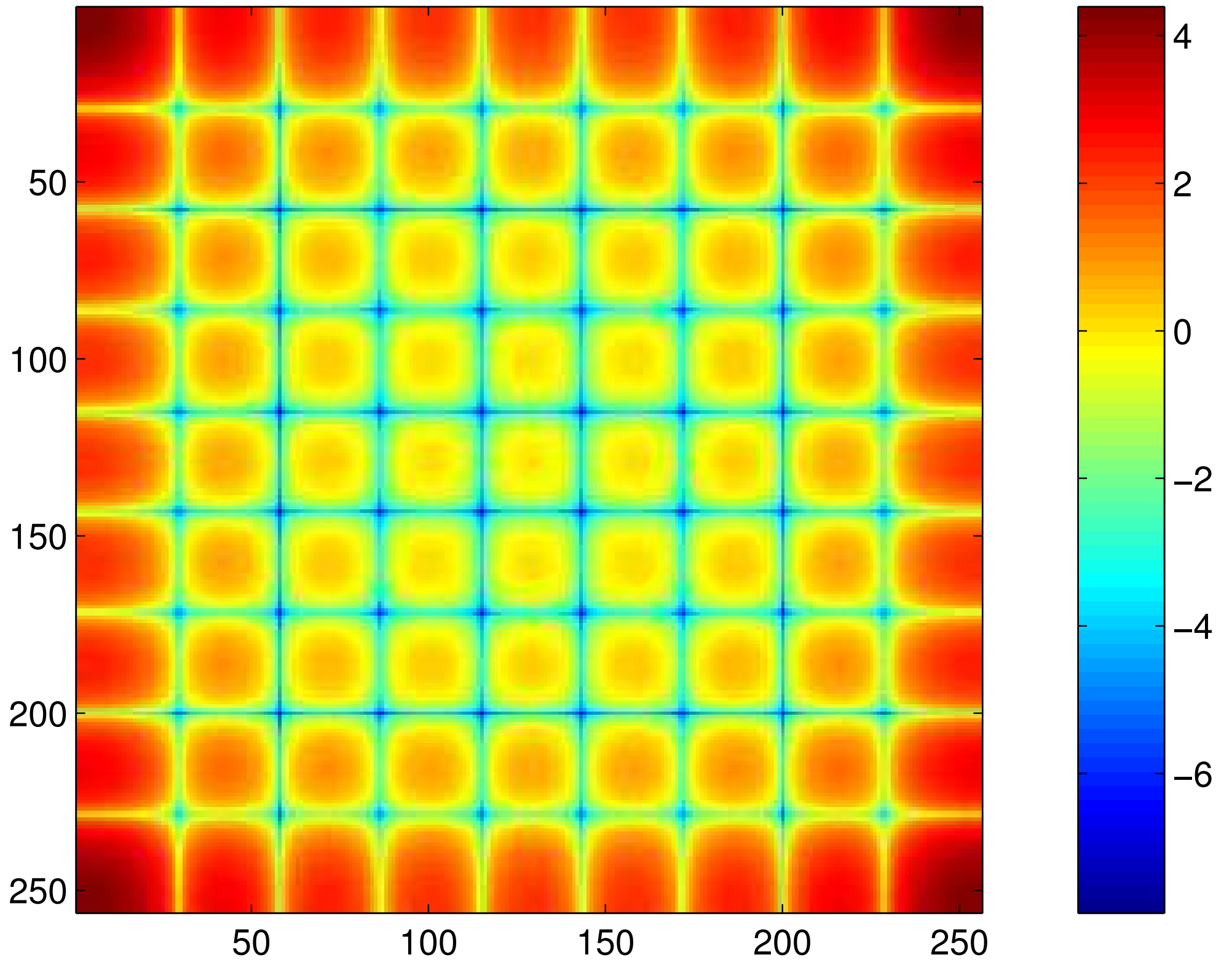
$\text{abs}(\text{fftshift}(\text{fft}(f)))$



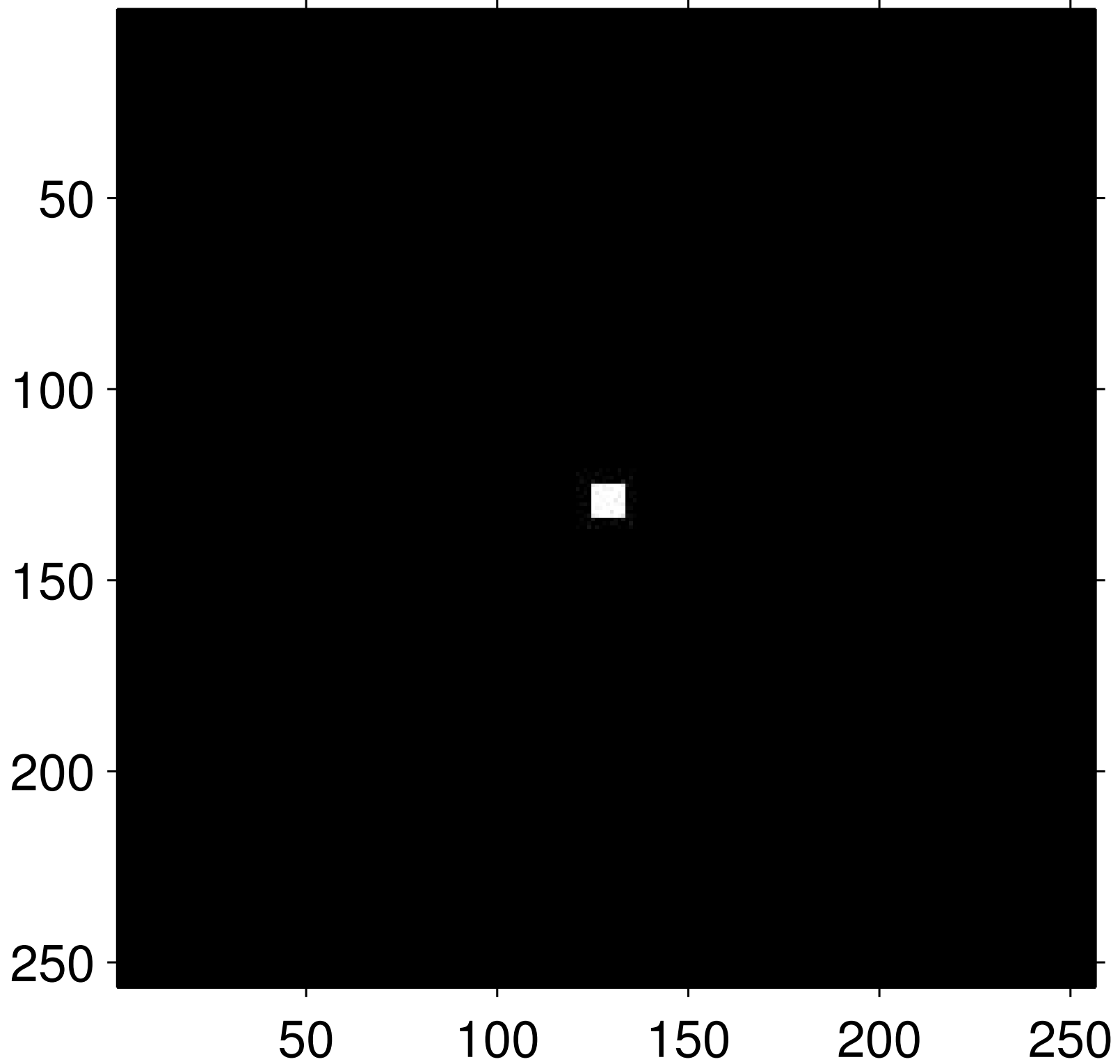
image



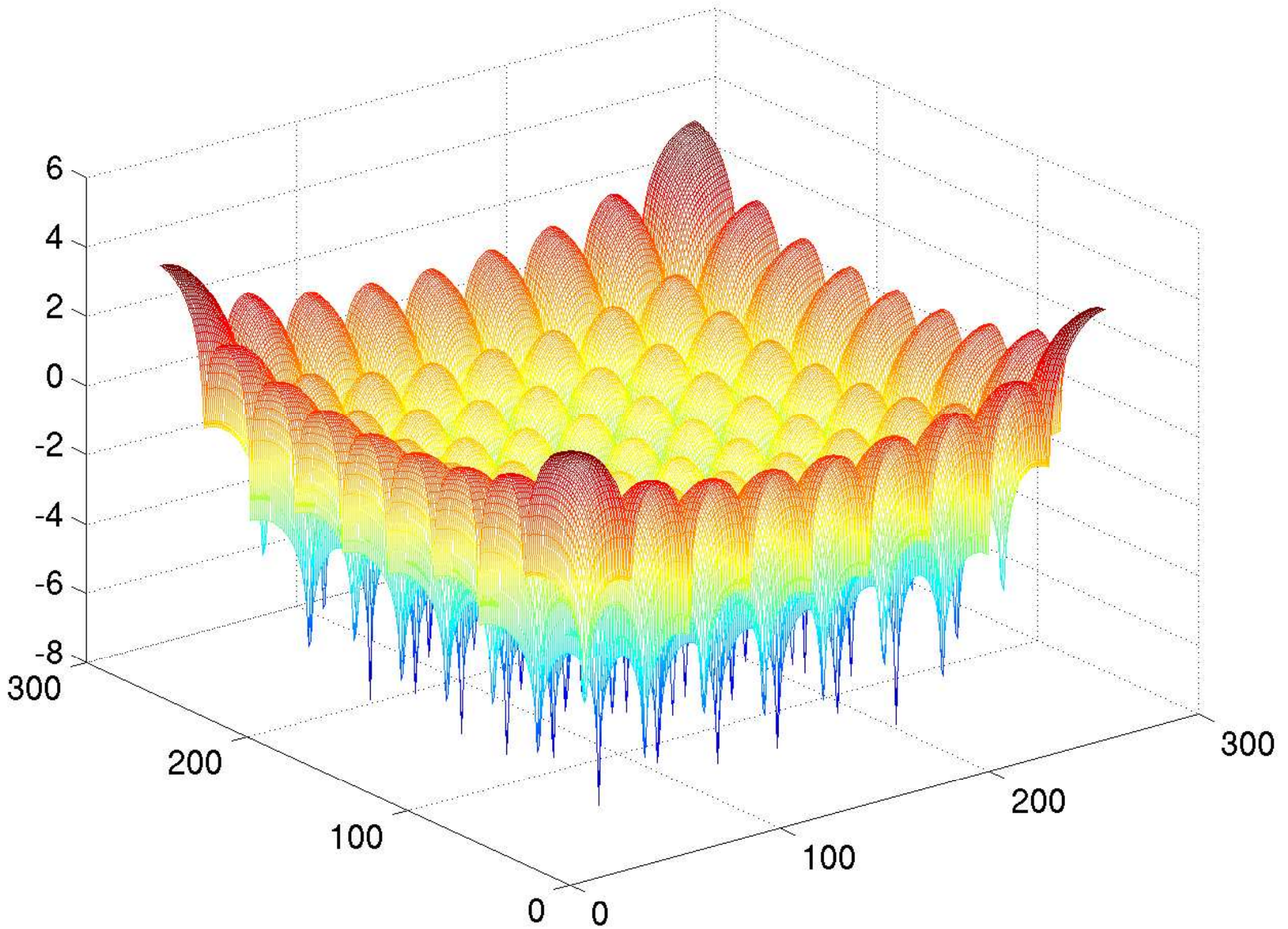
$\log(\text{abs}(\text{fft2}(\text{im})))$



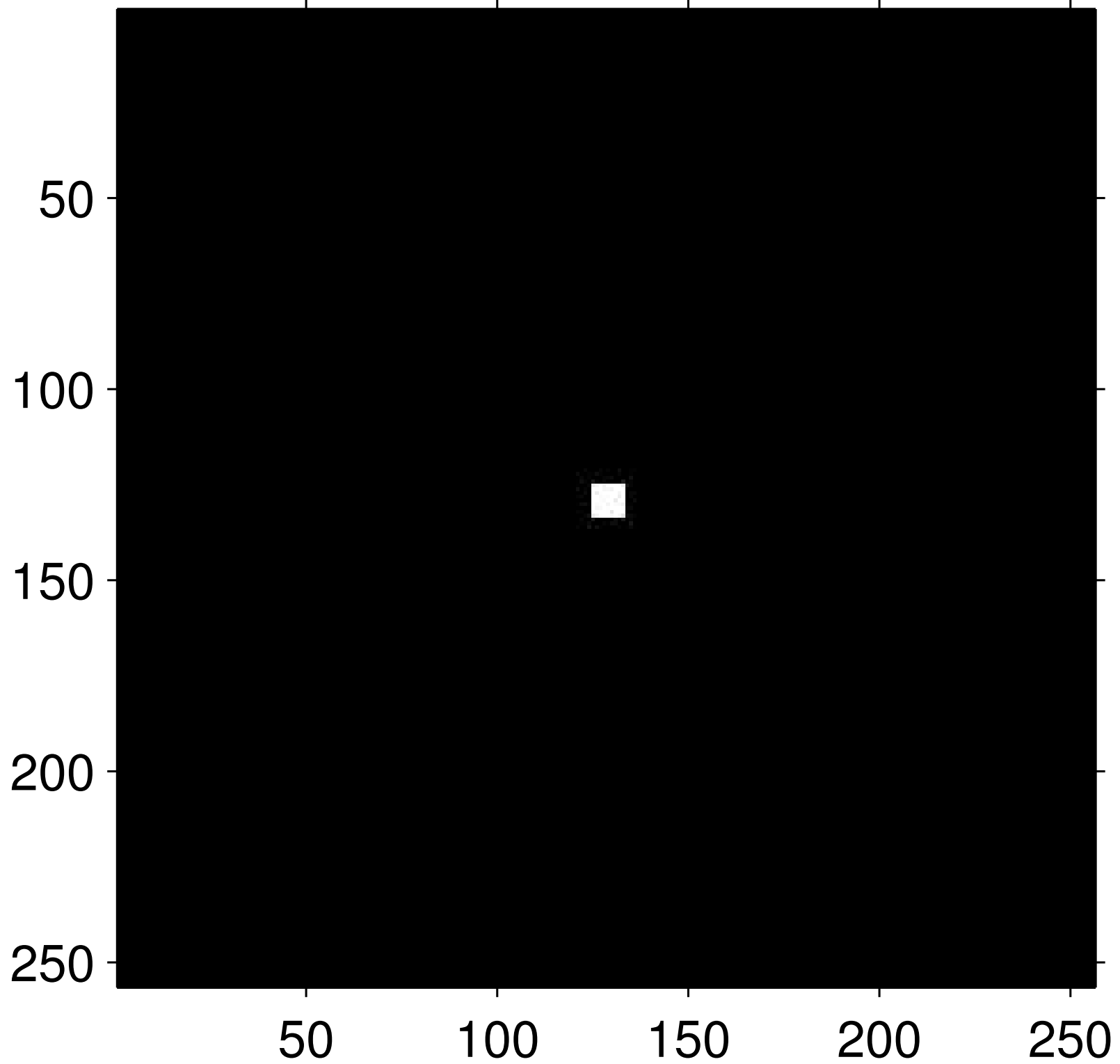
image



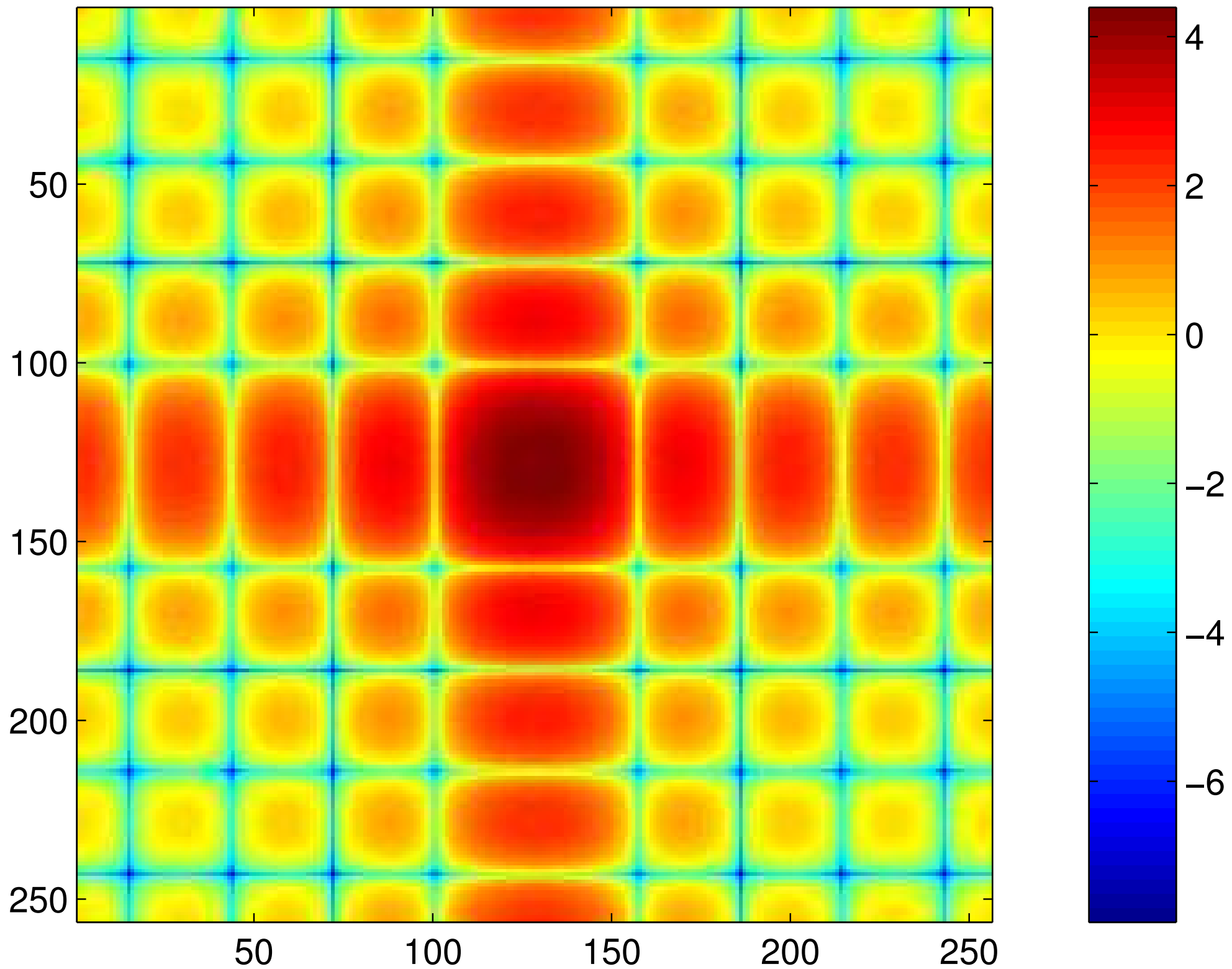
$\log(\text{abs}(\text{fft2}(\text{im})))$



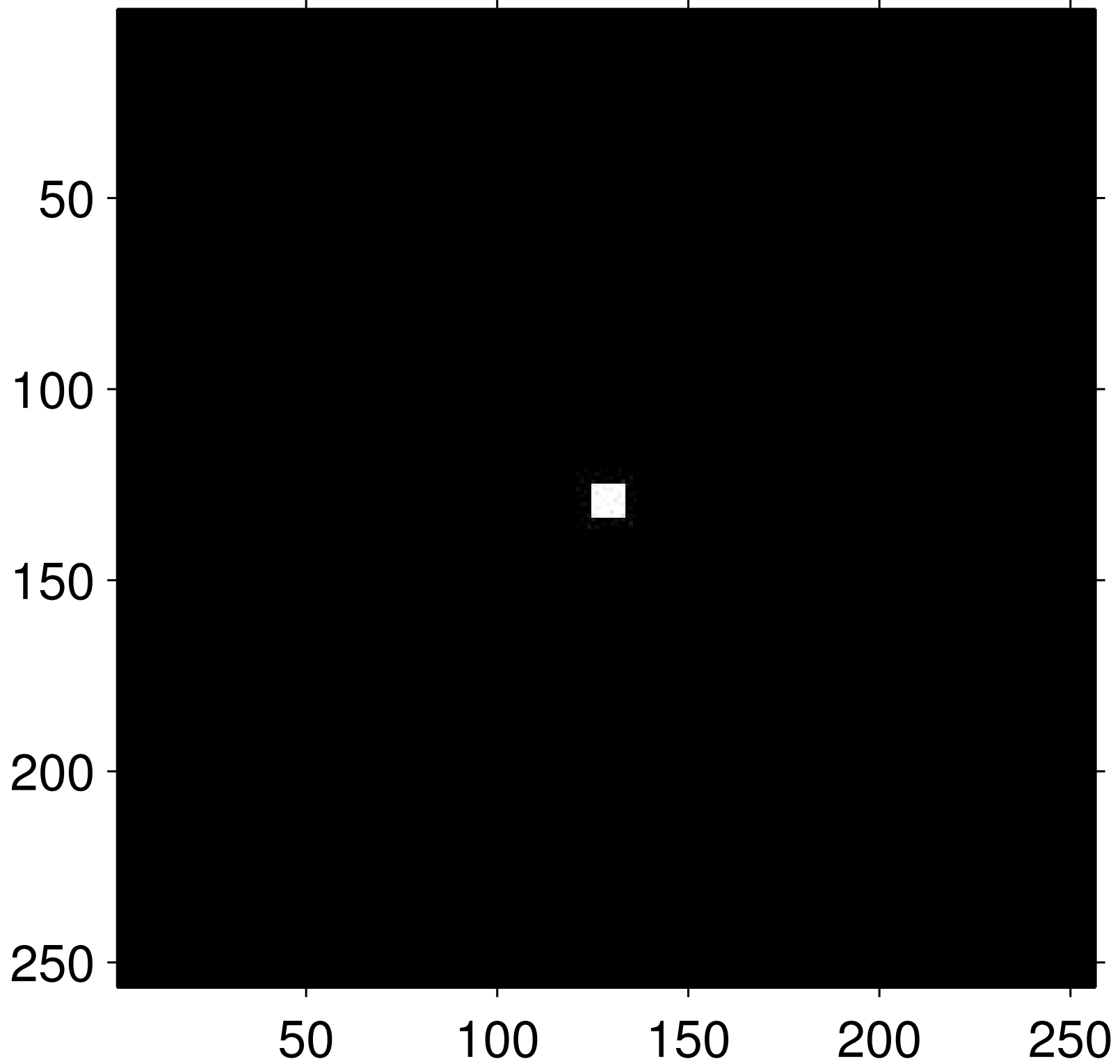
image



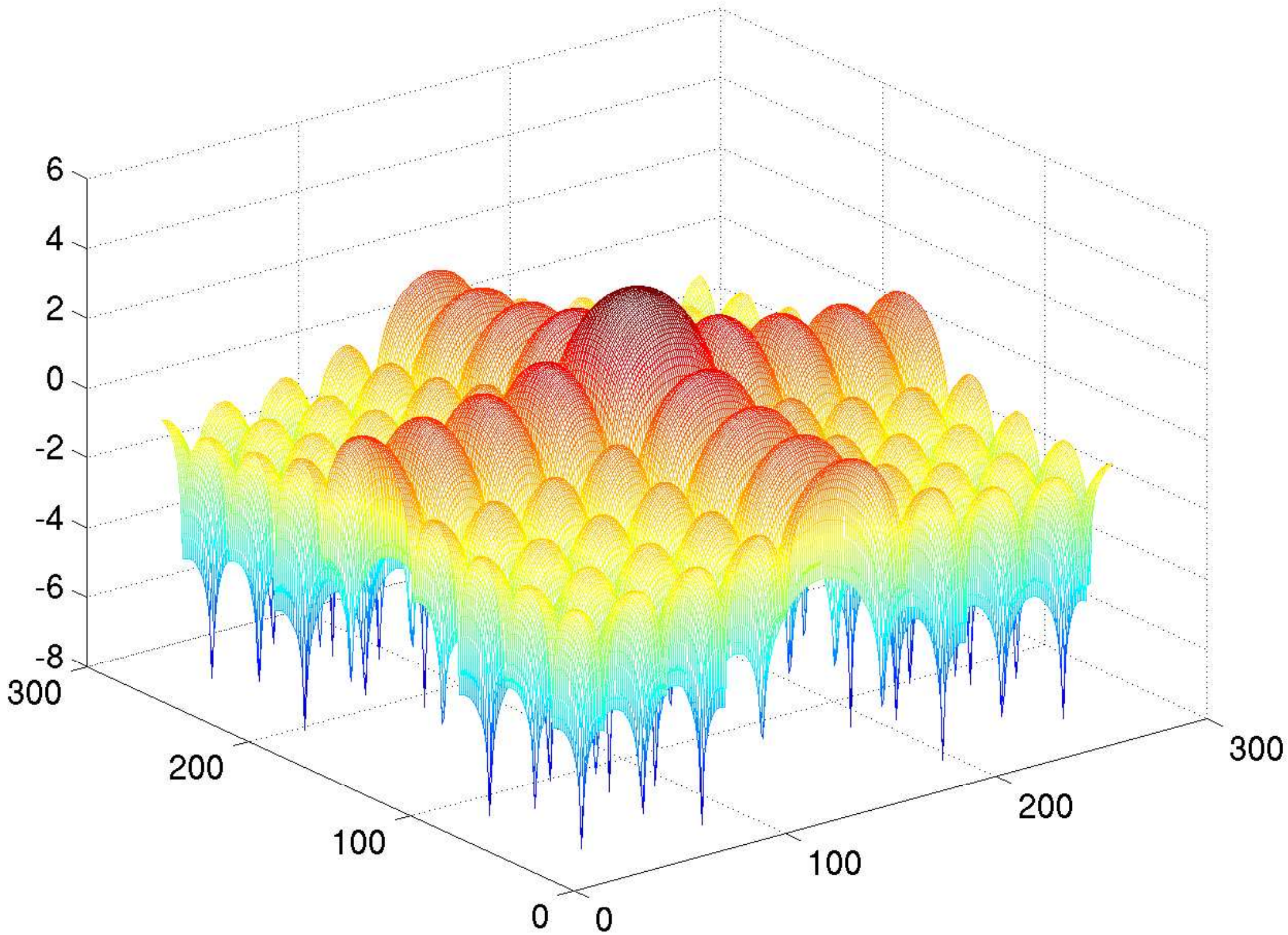
$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$

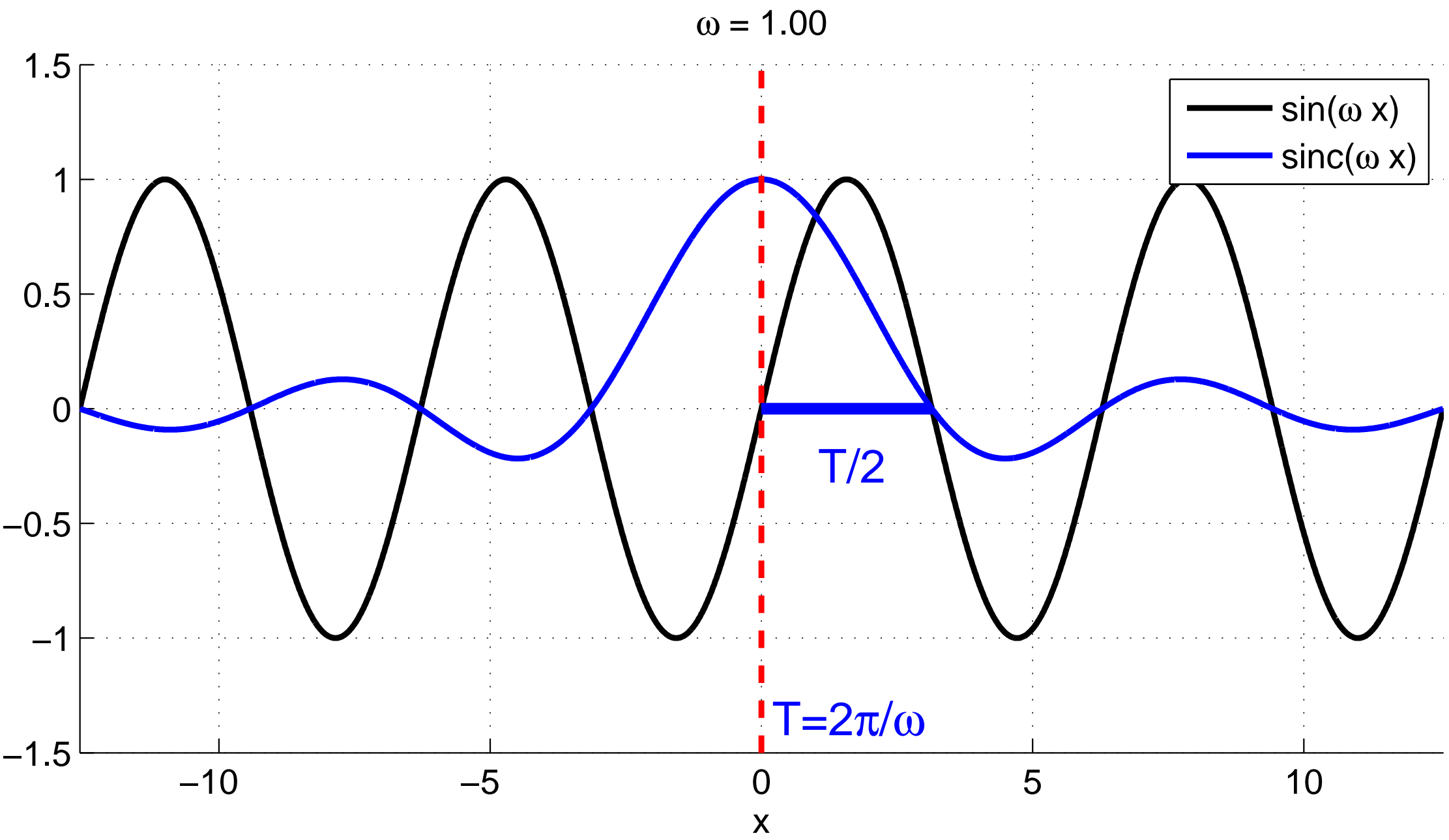


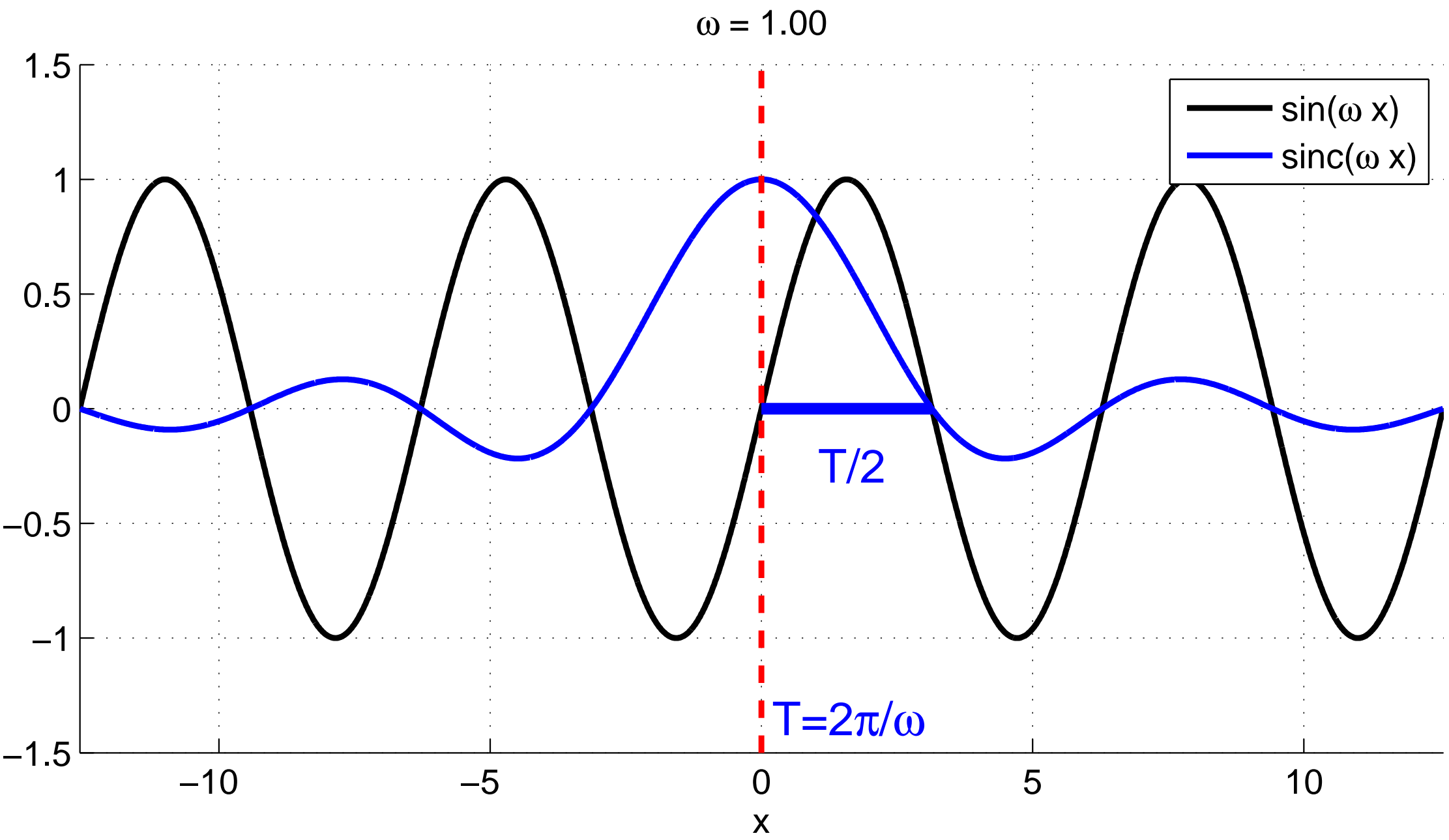
image

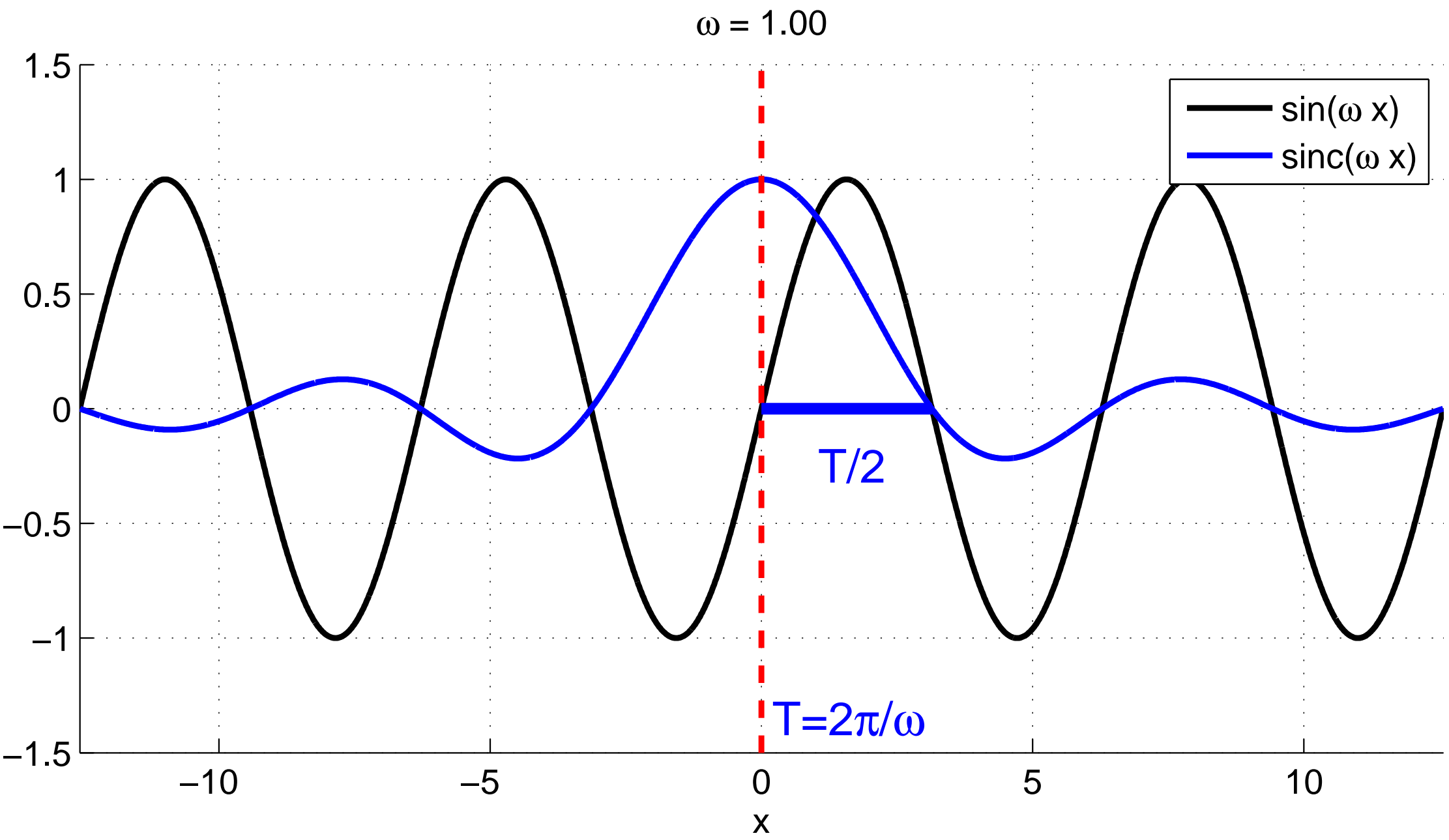


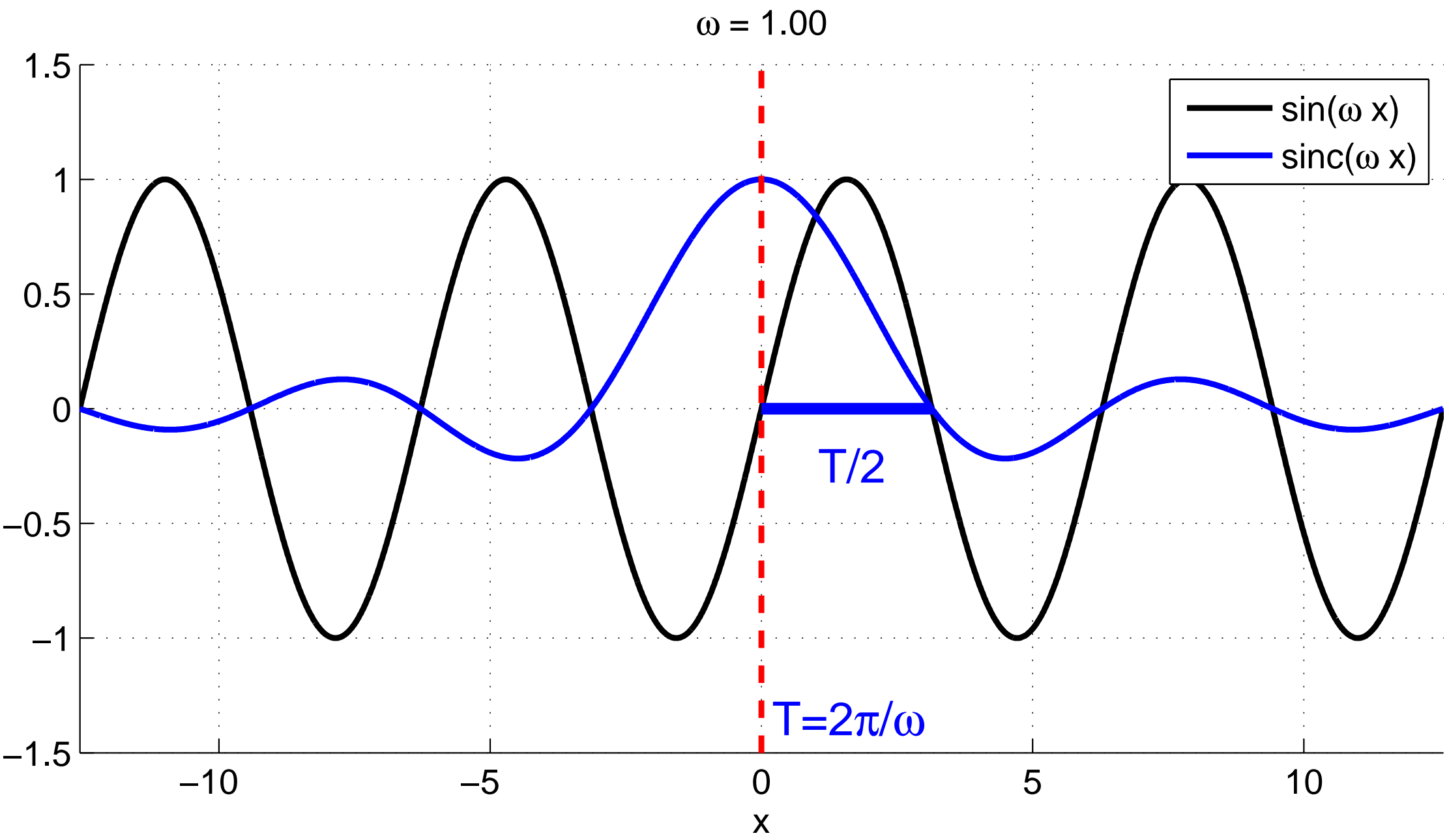
$\log(\text{abs}(\text{fftshift}(\text{fft2}(\text{im}))))$

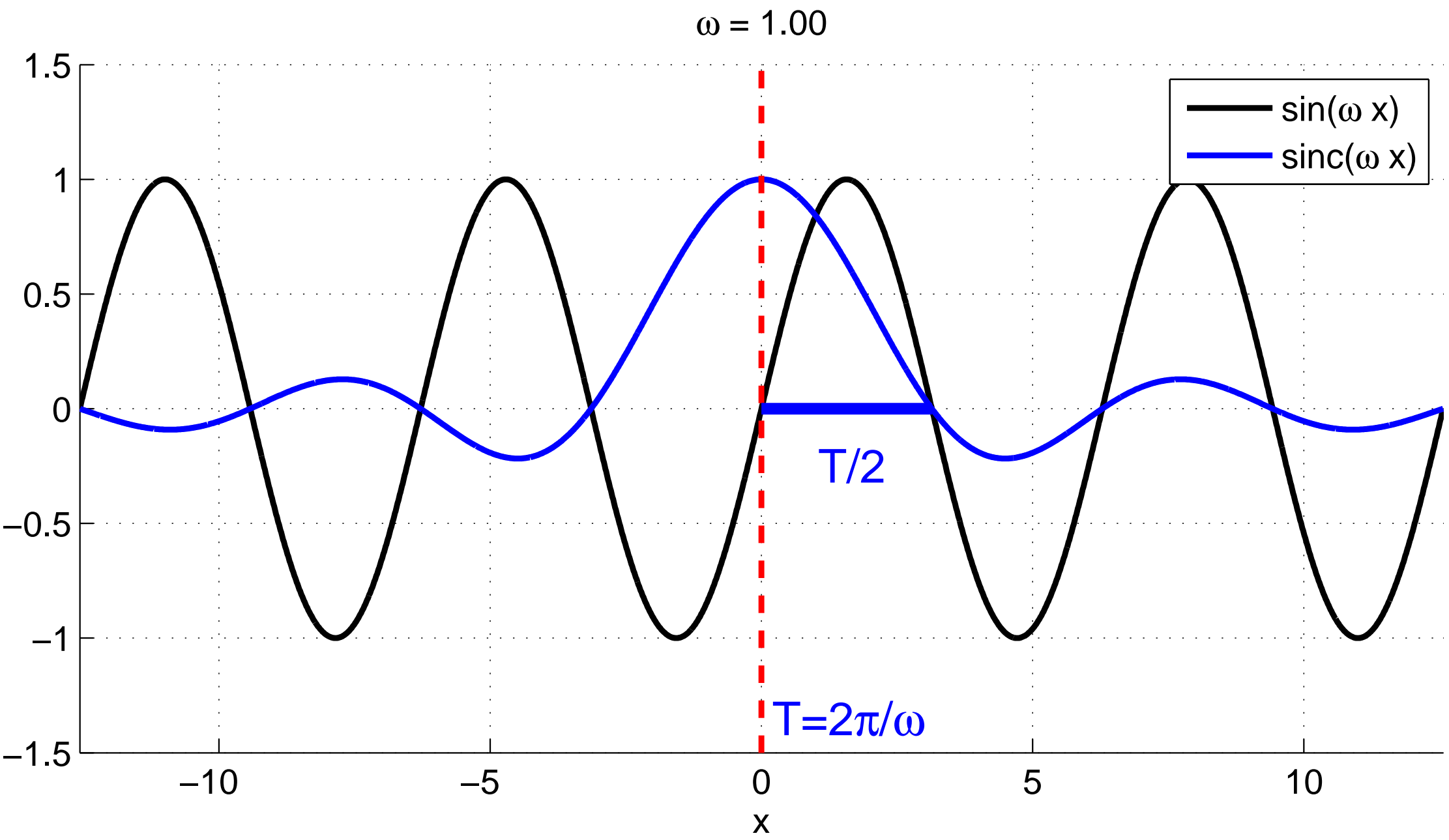




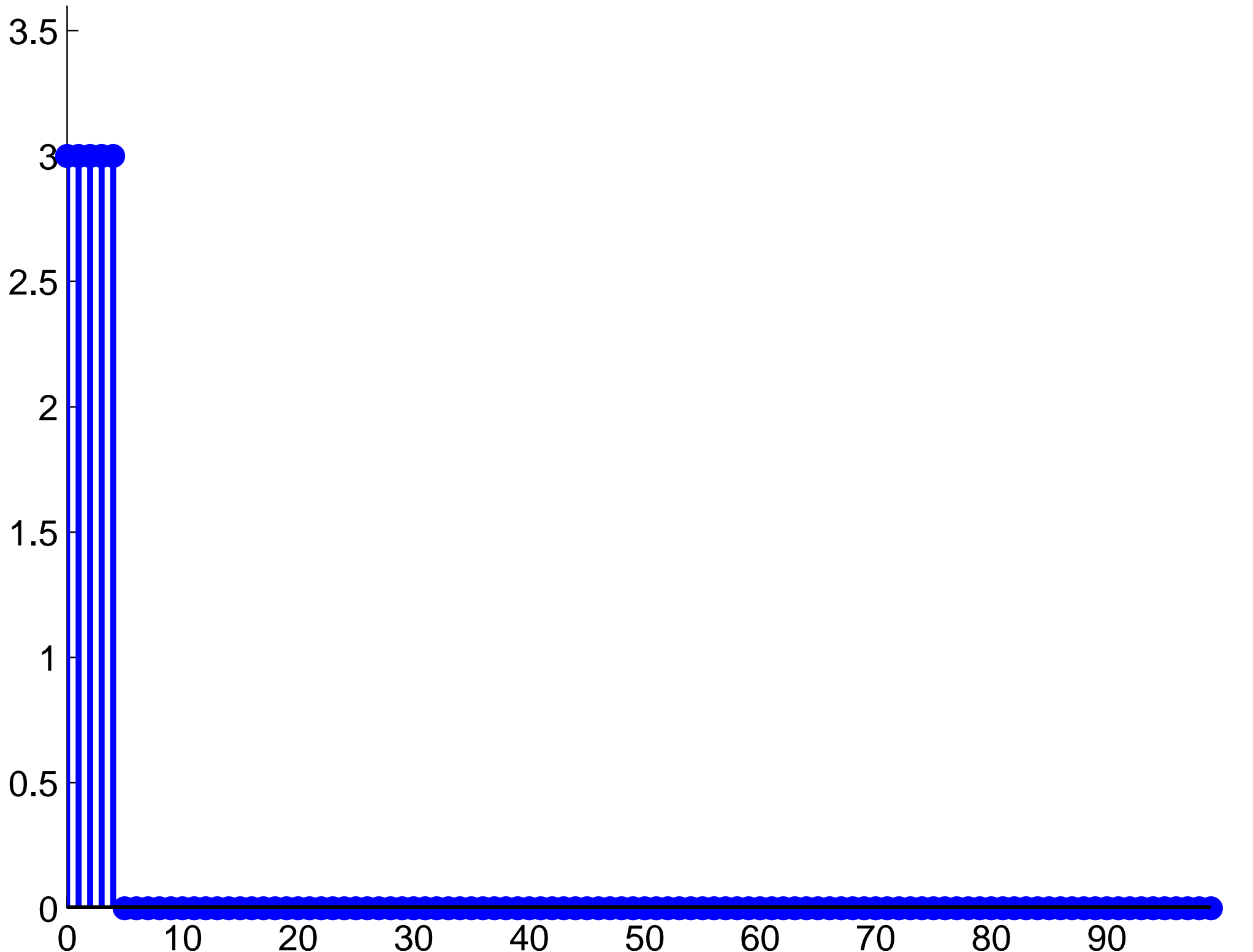




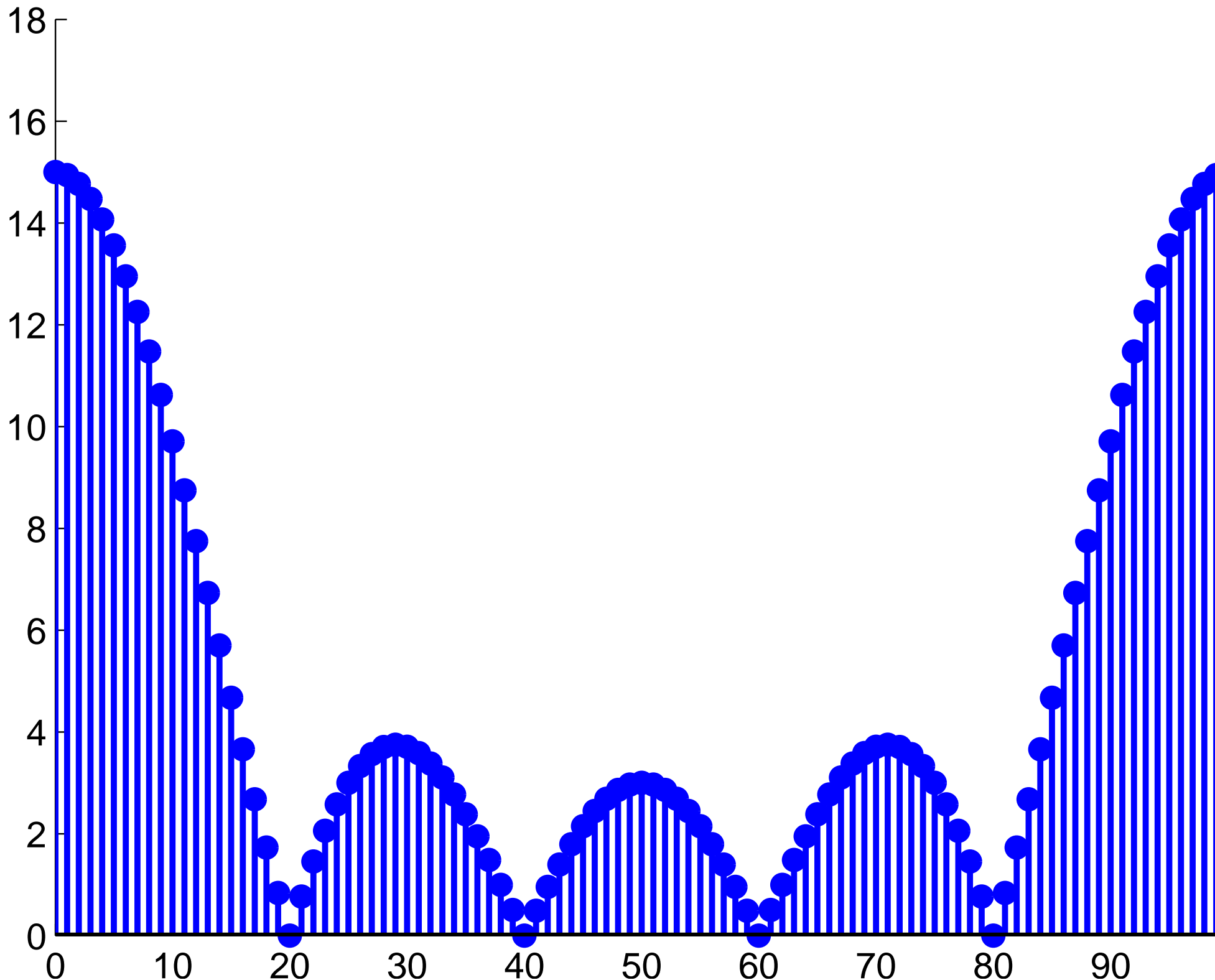




$f(x)$

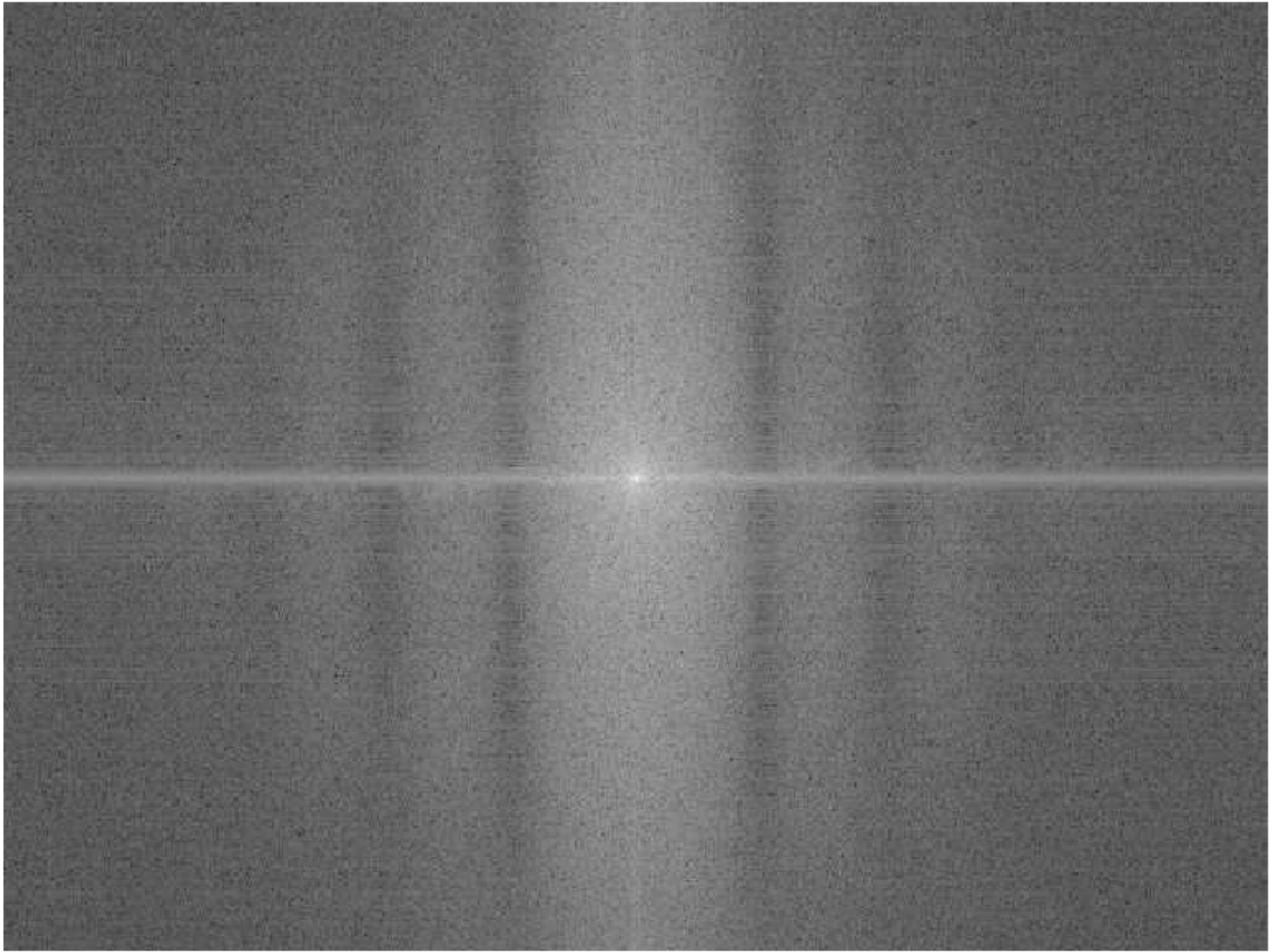


abs(fft(f))

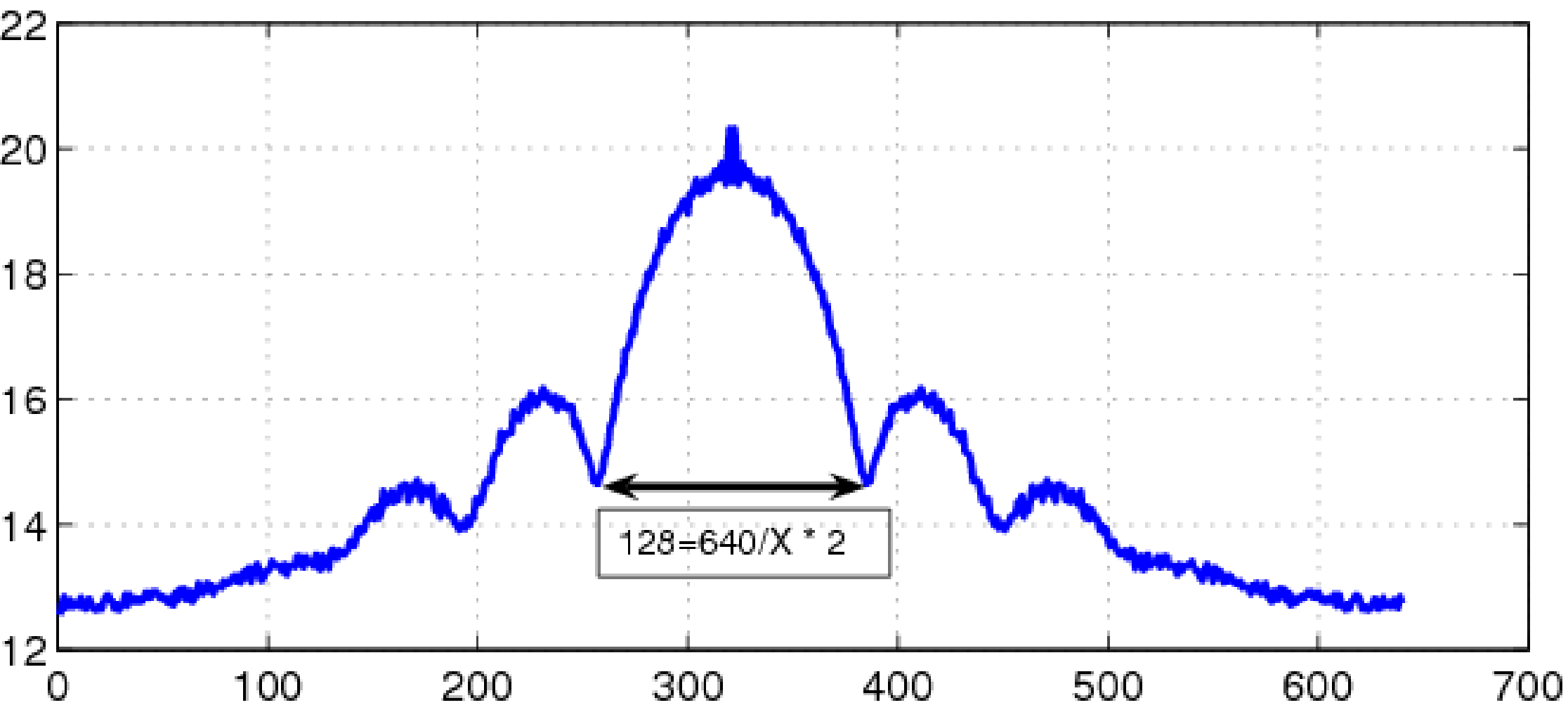




spectrum of blurred image

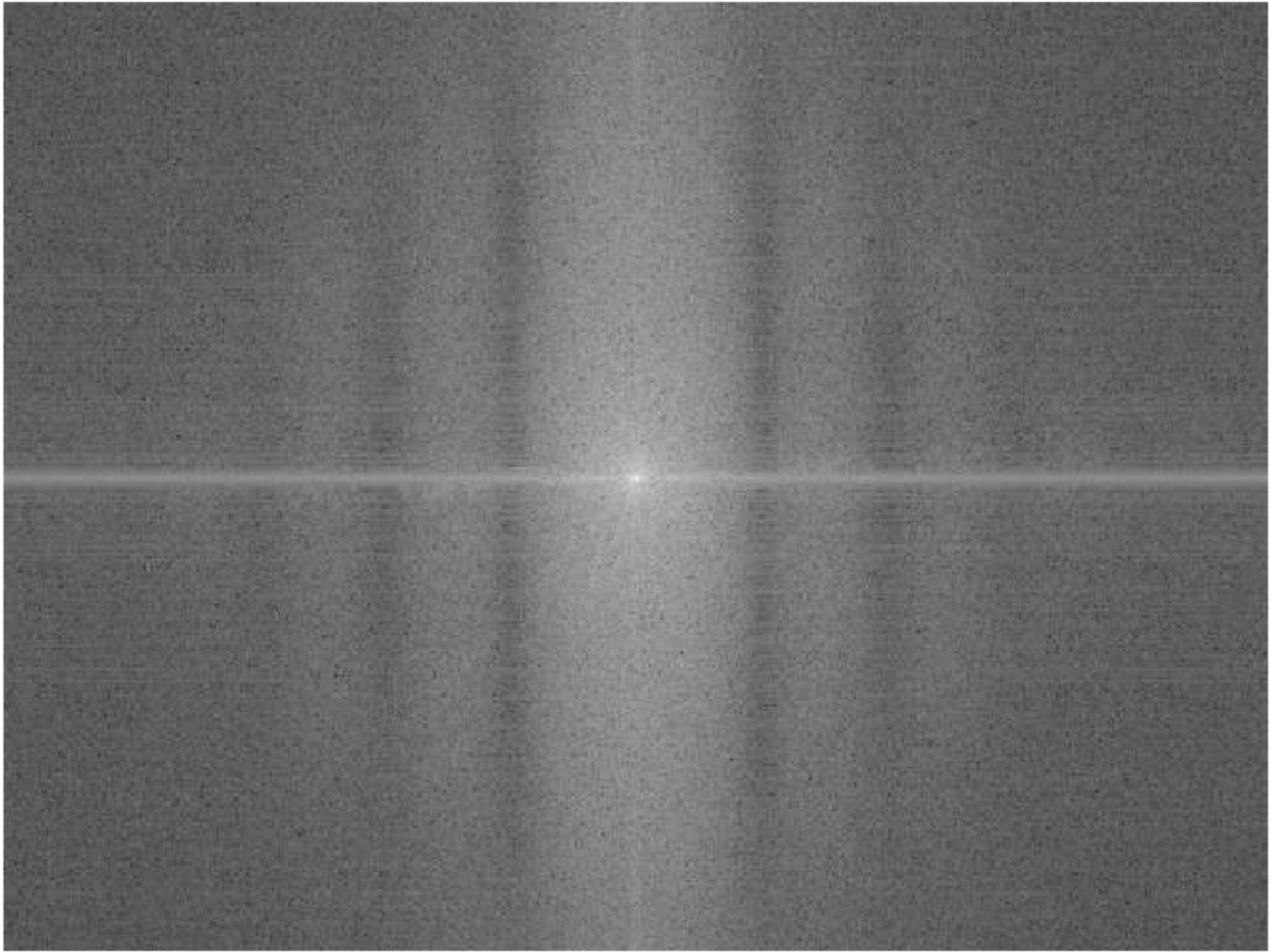


median(log(1+abs(fft)))

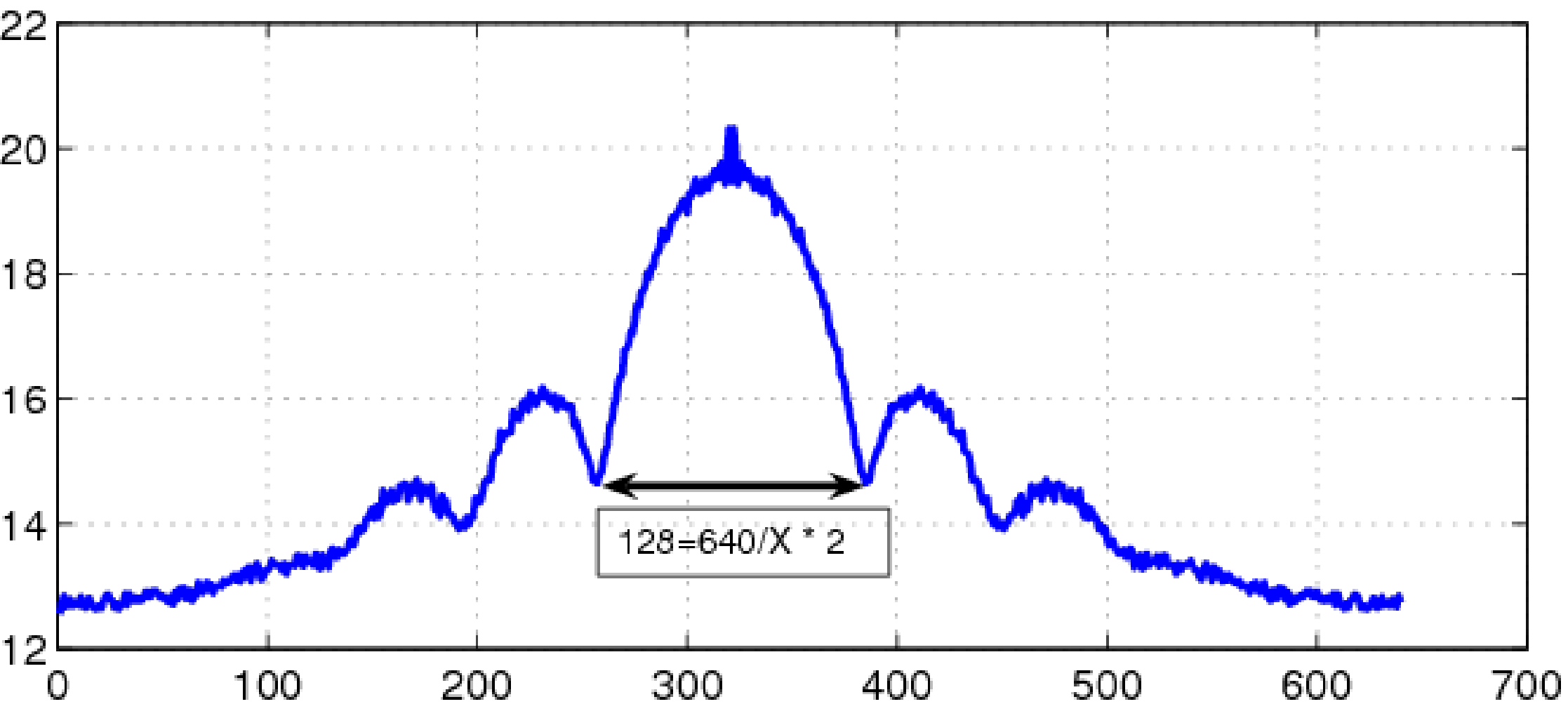




spectrum of blurred image

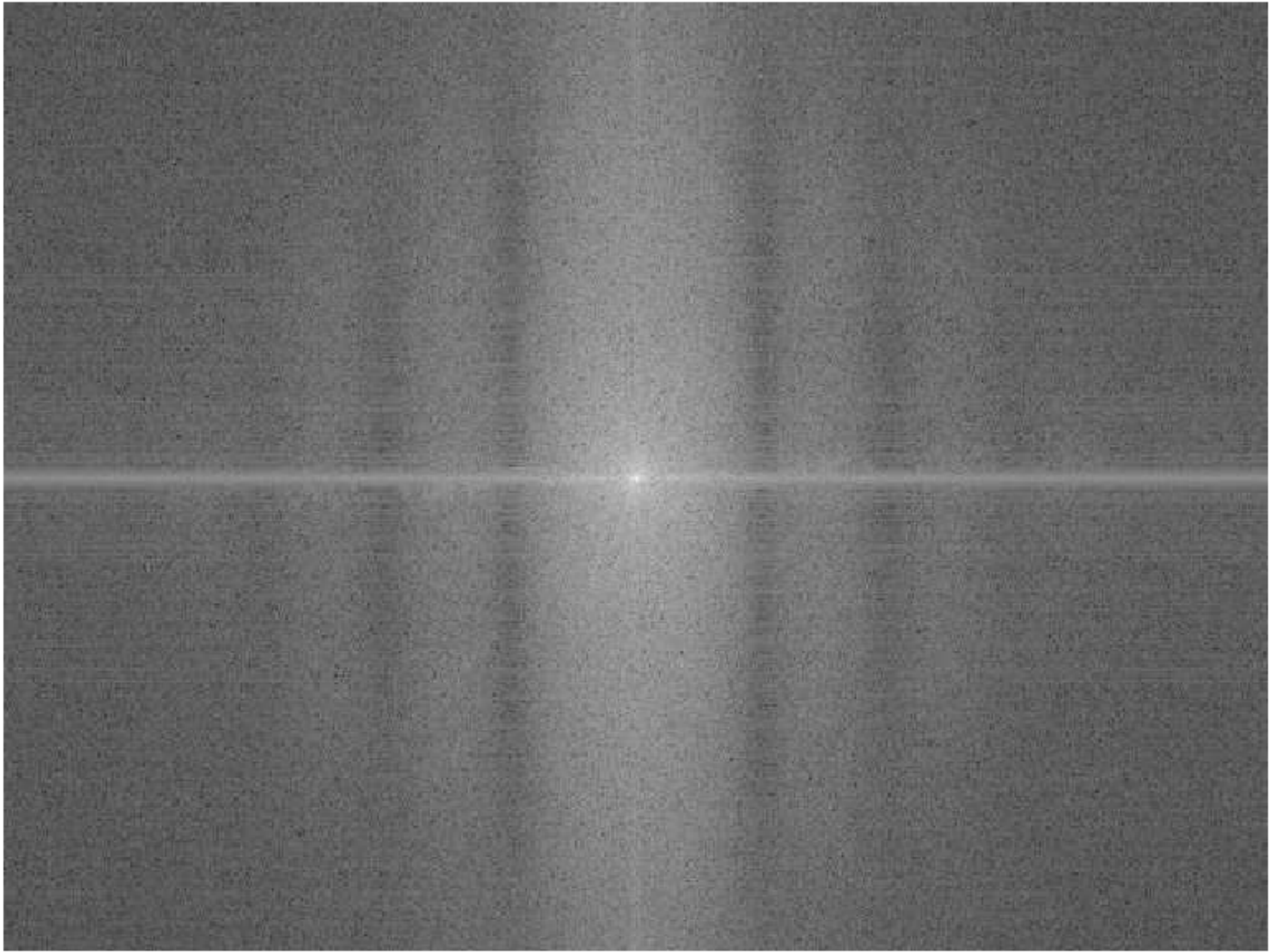


median(log(1+abs(fft)))

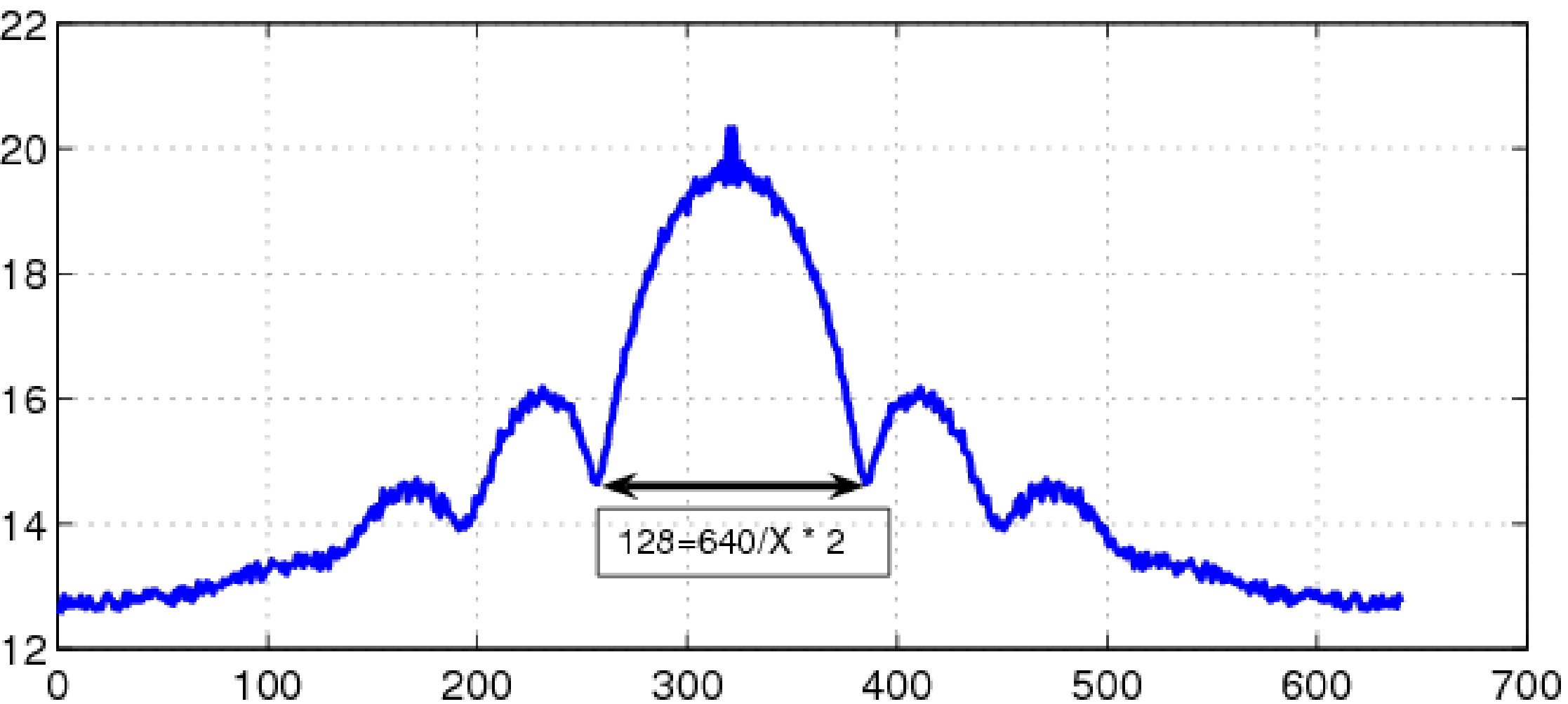




spectrum of blurred image

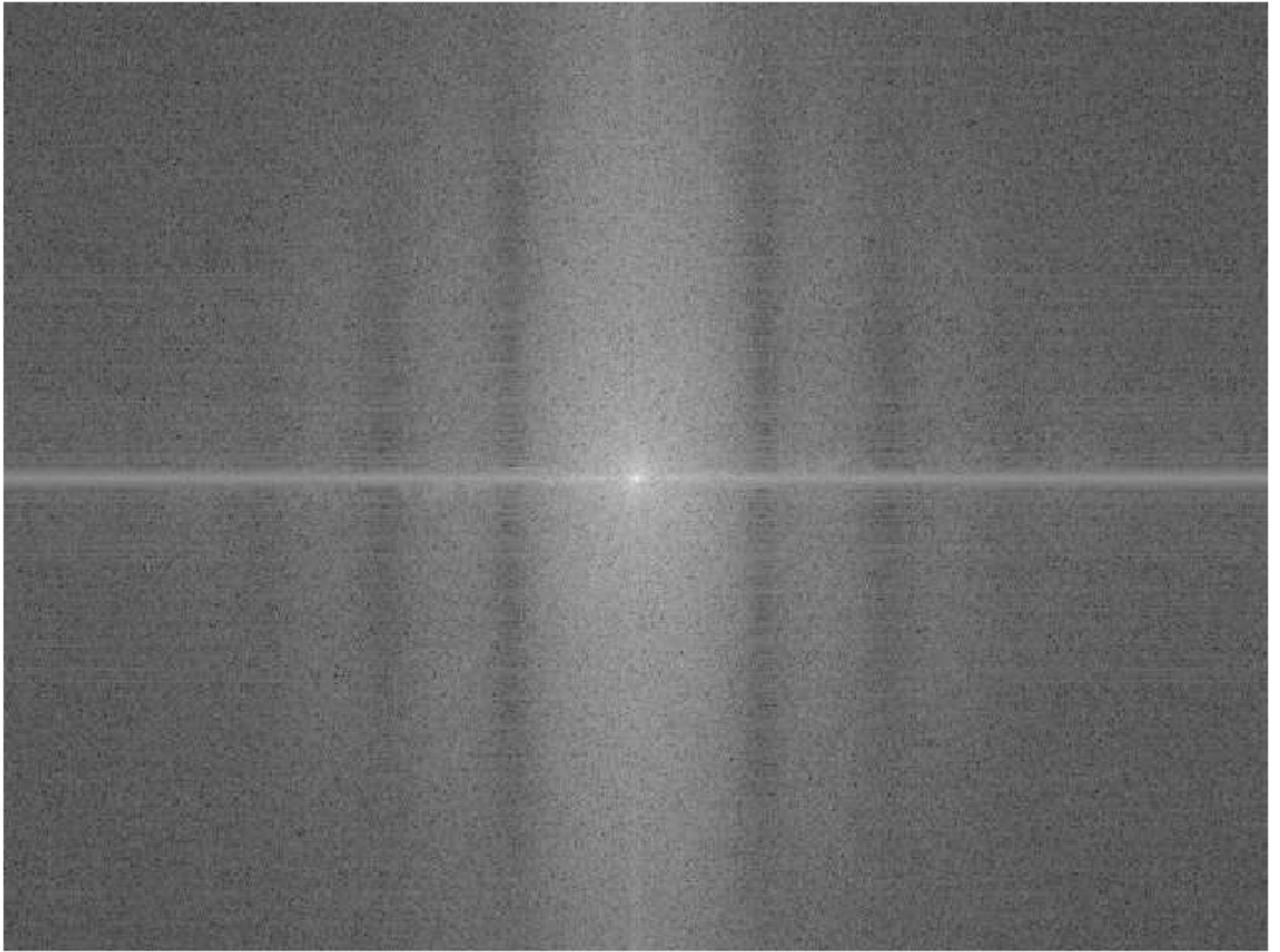


median(log(1+abs(fft)))

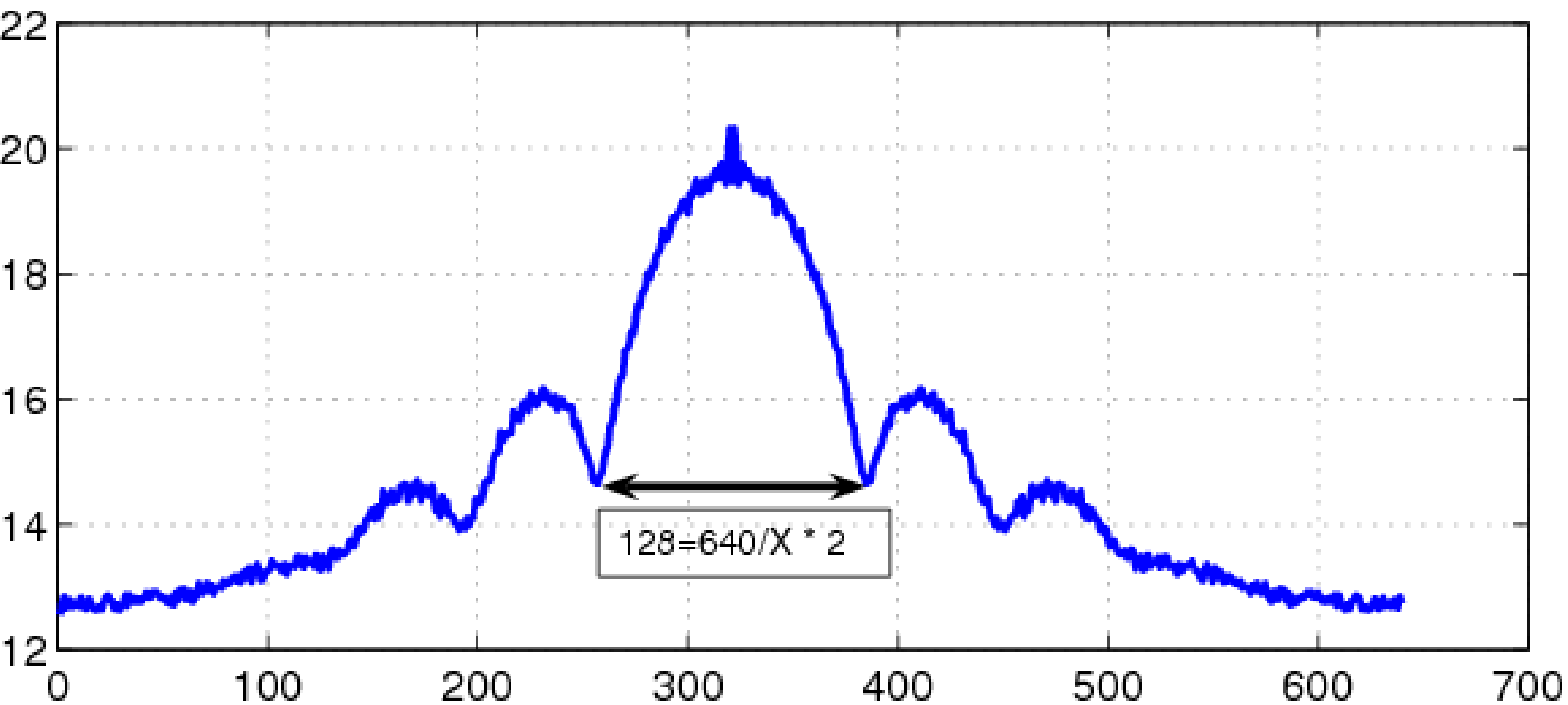




spectrum of blurred image



median(log(1+abs(fft)))









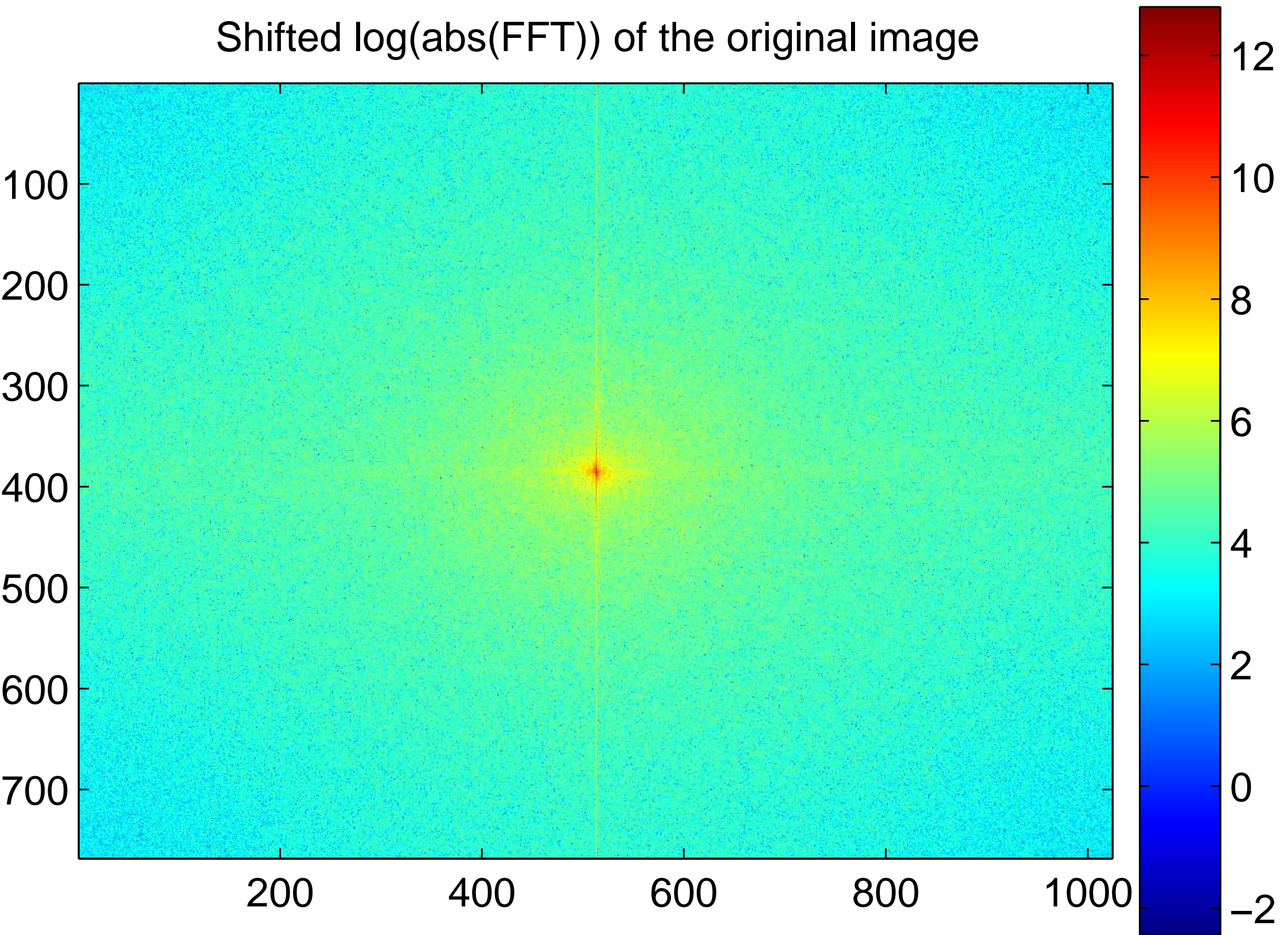




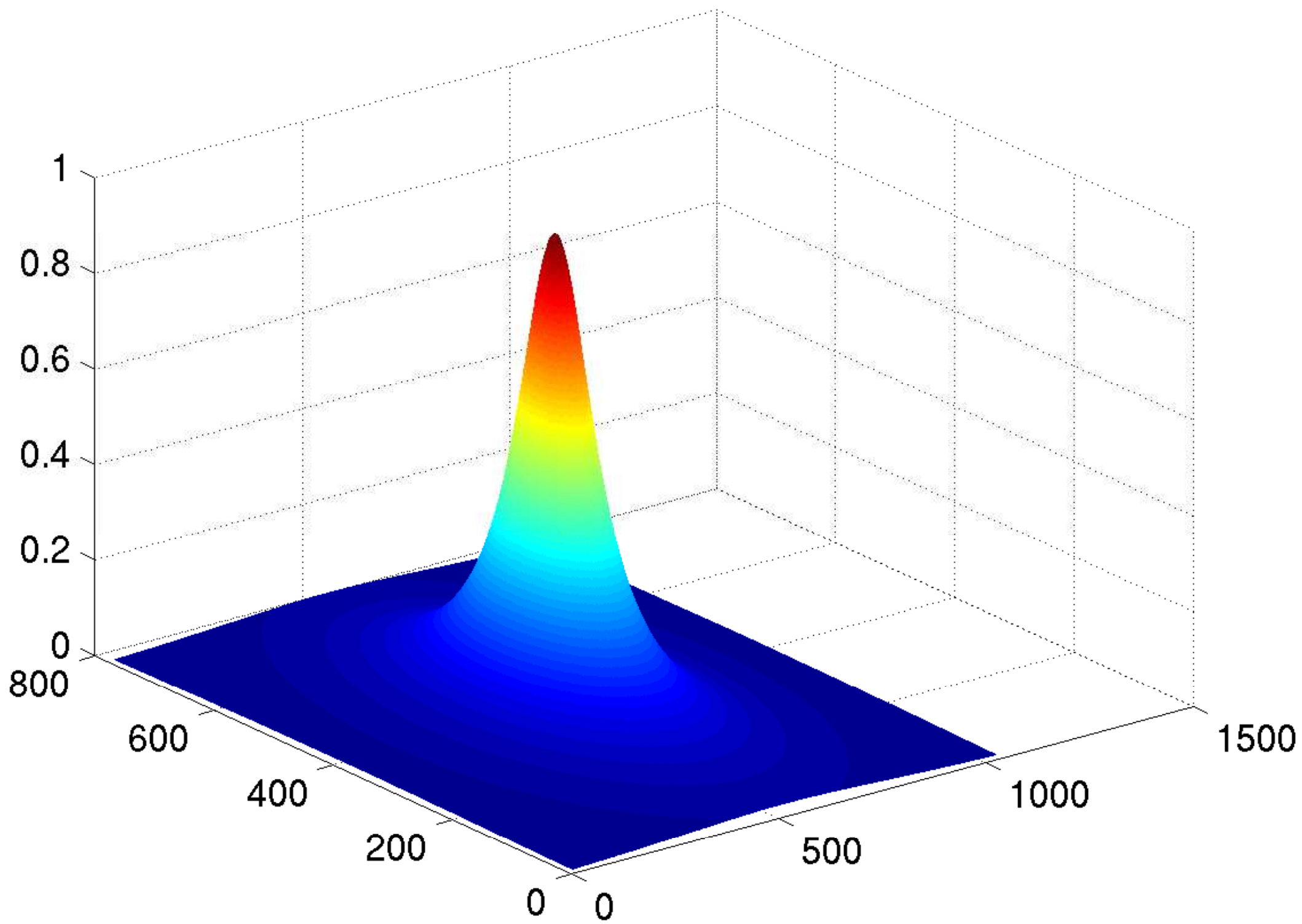




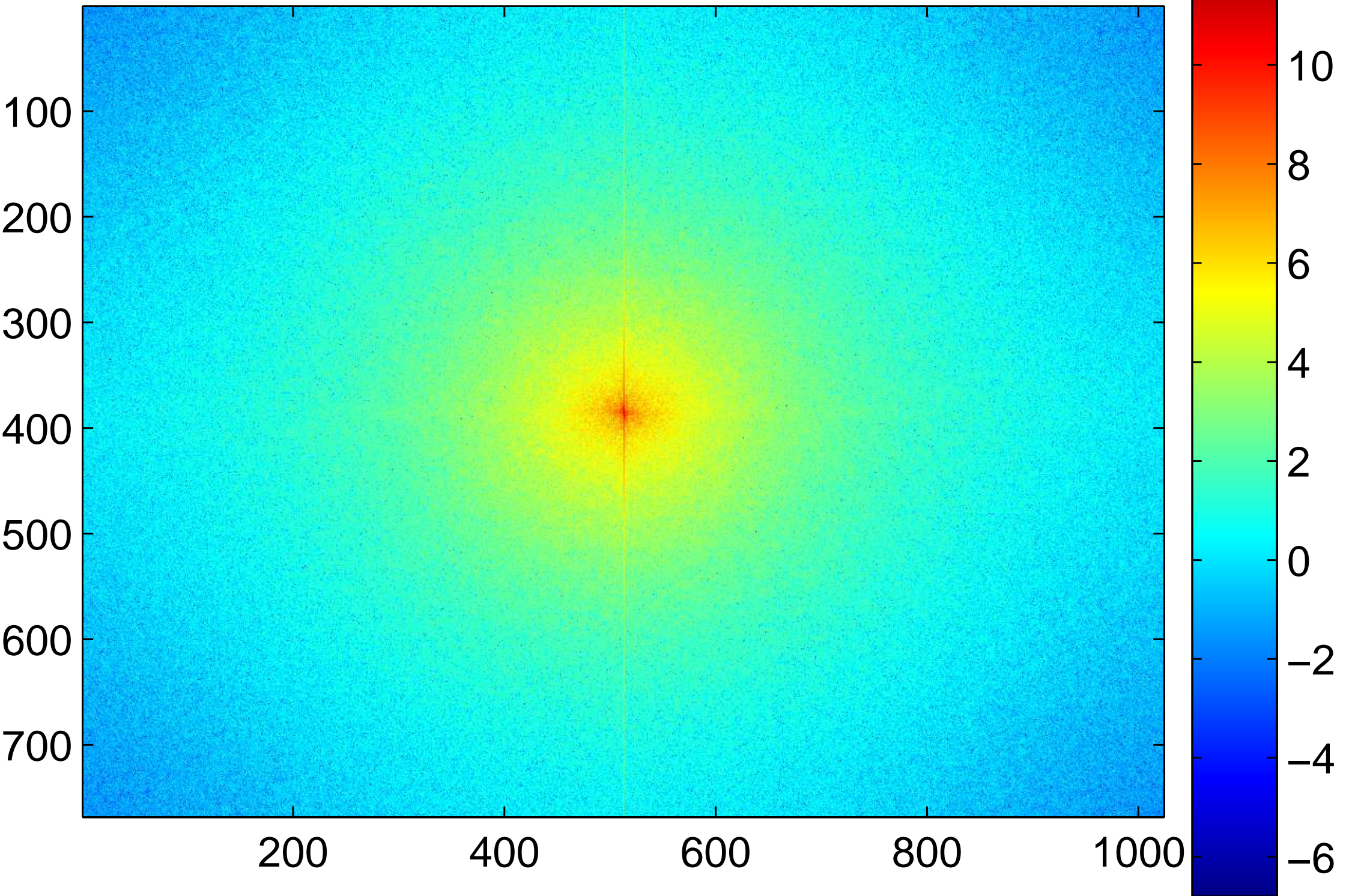
Shifted $\log(\text{abs}(\text{FFT}))$ of the original image



lp Butt filter n=1, cutoff=66



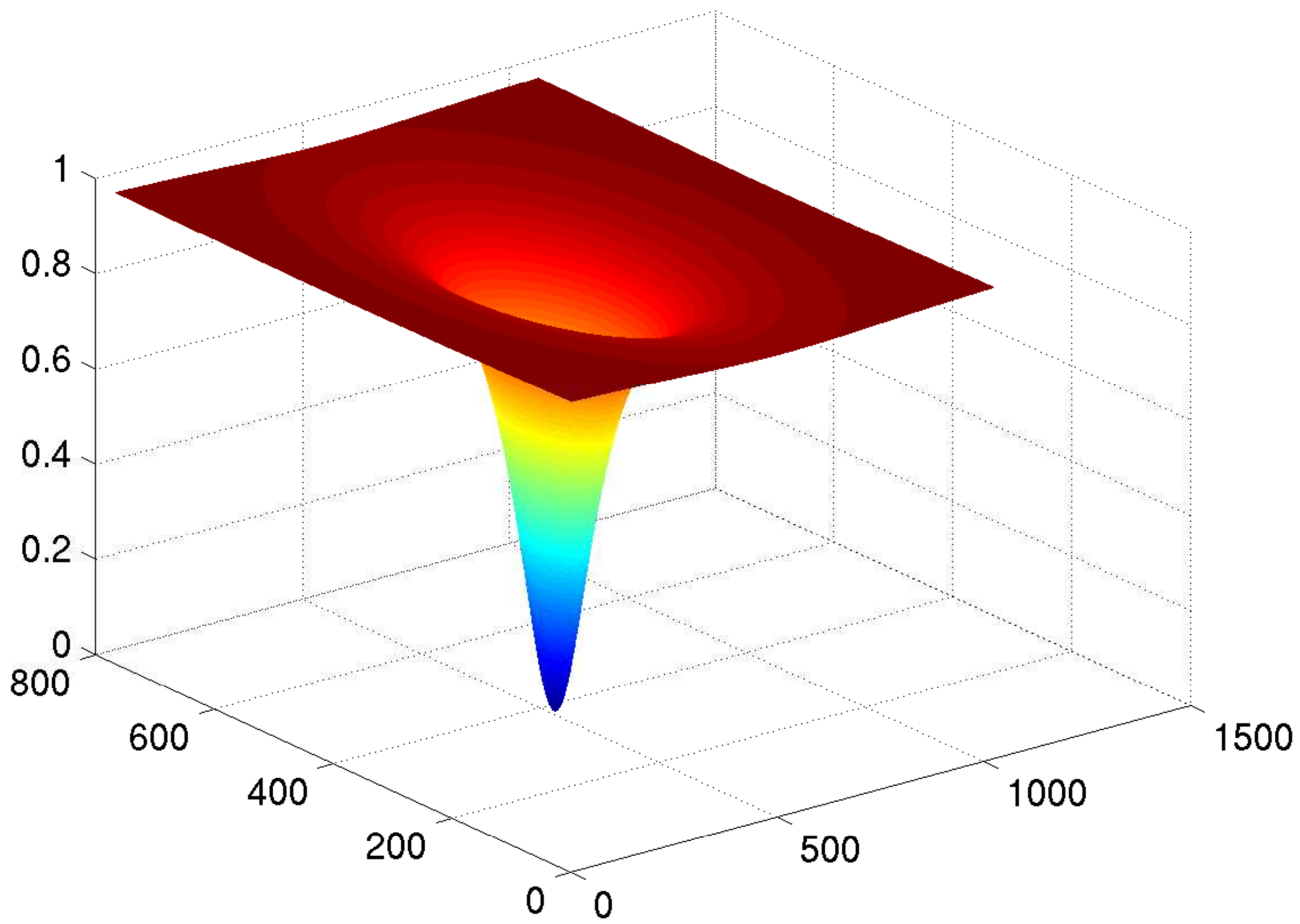
Shifted $\log(\text{abs}(\text{FFT}))$ of the filtered image



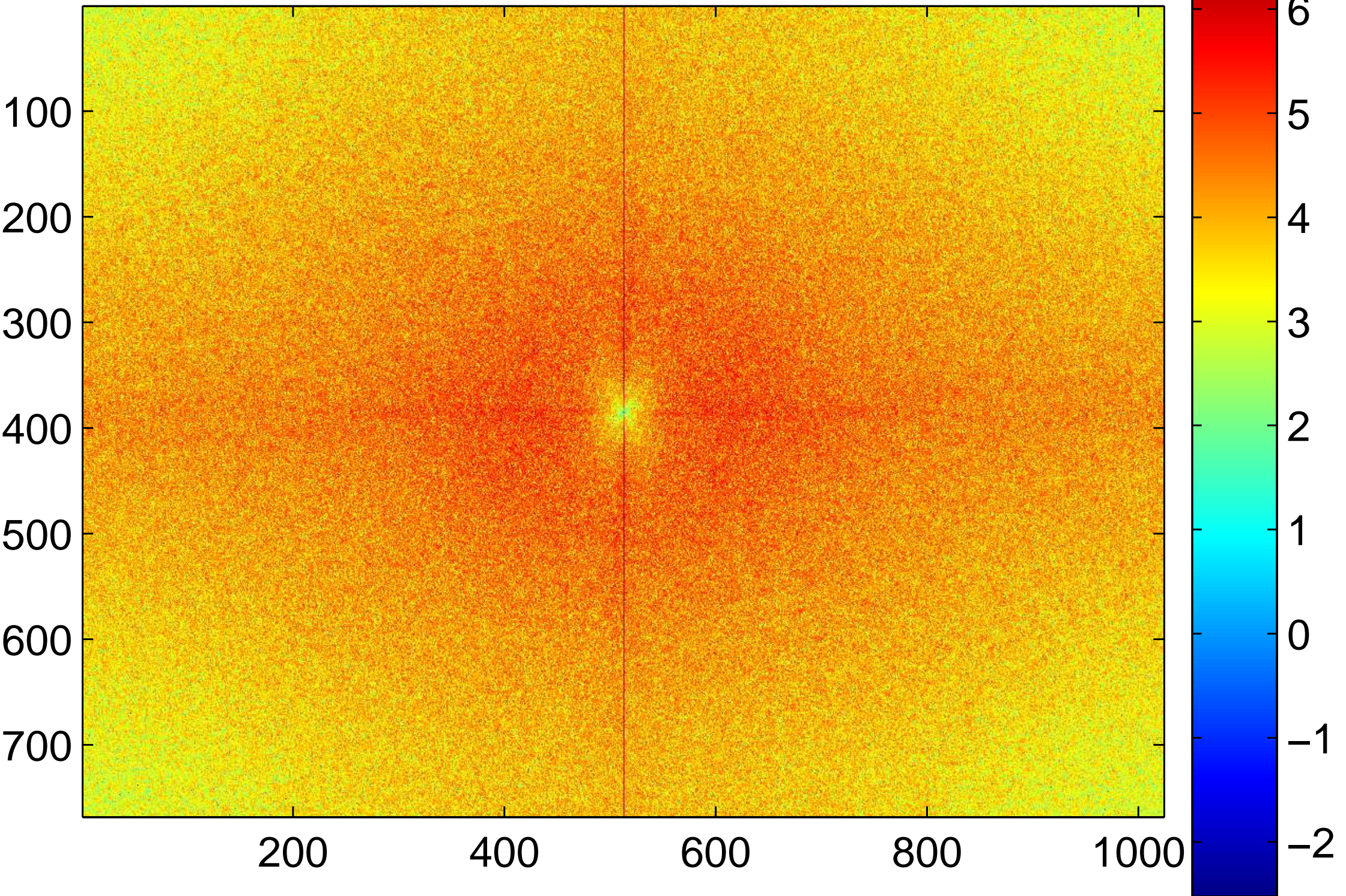




hp Butt filter n=1, cutoff=66

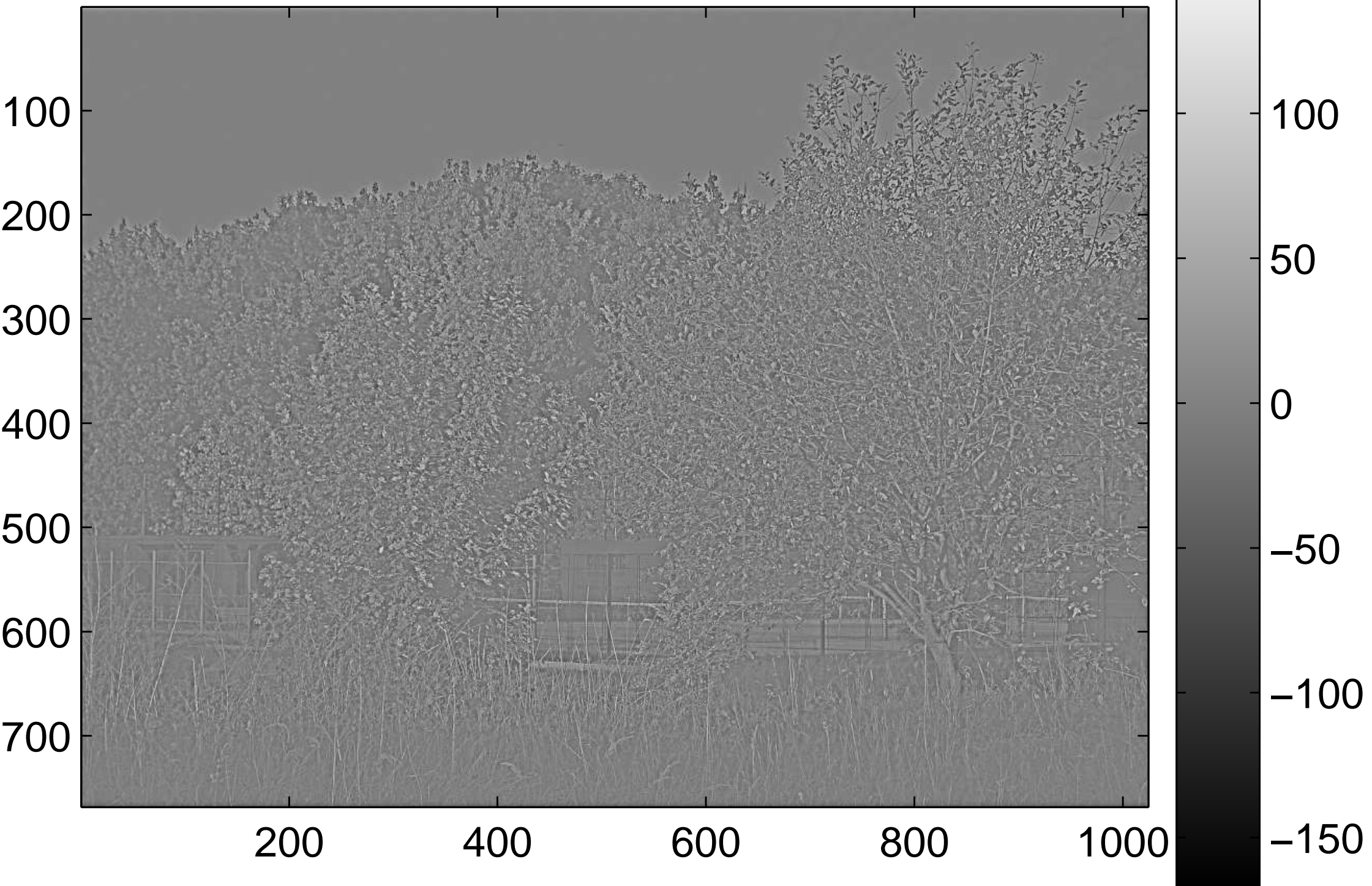


Shifted $\log(\text{abs}(\text{FFT}))$ of the filtered image

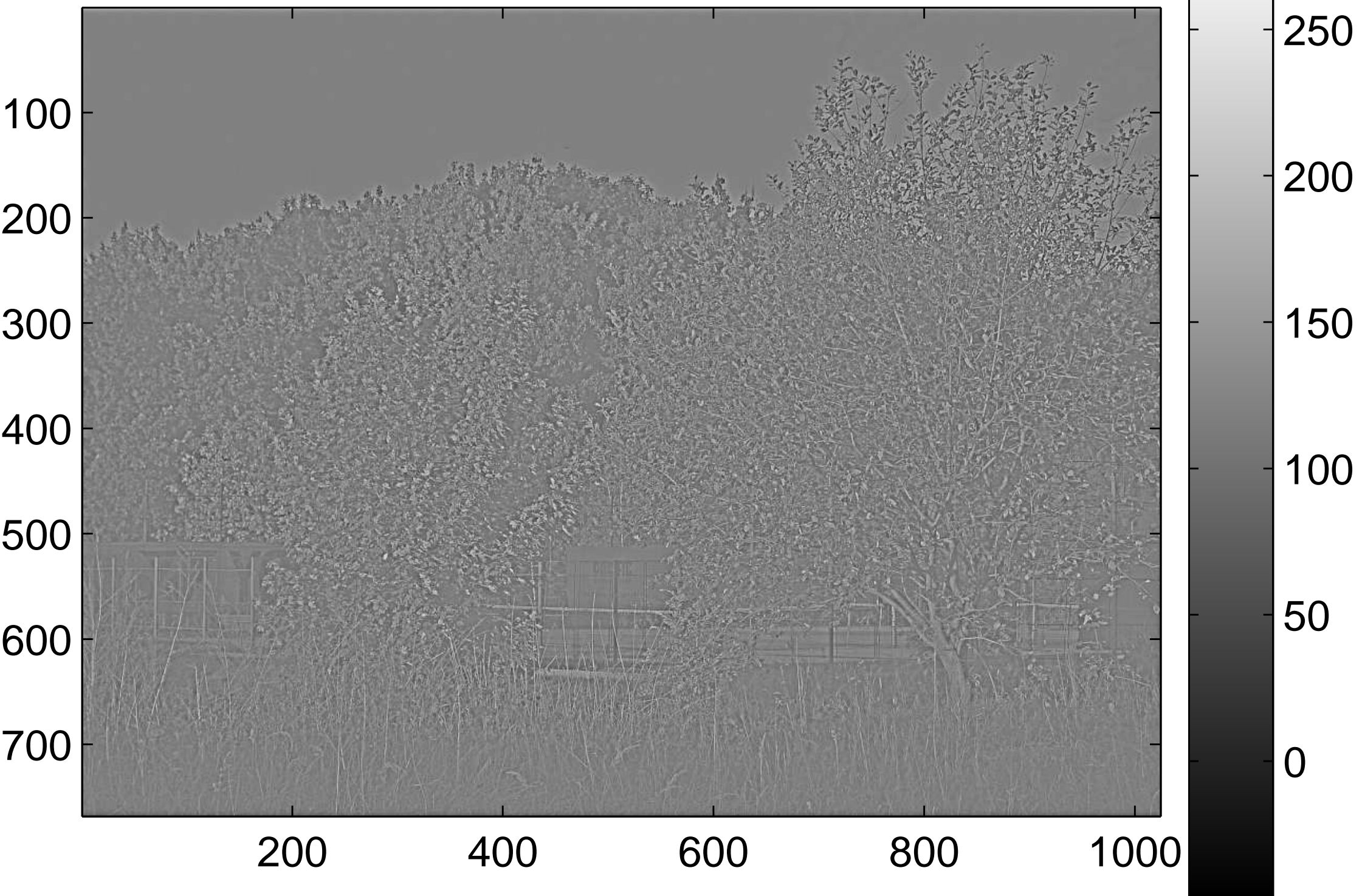




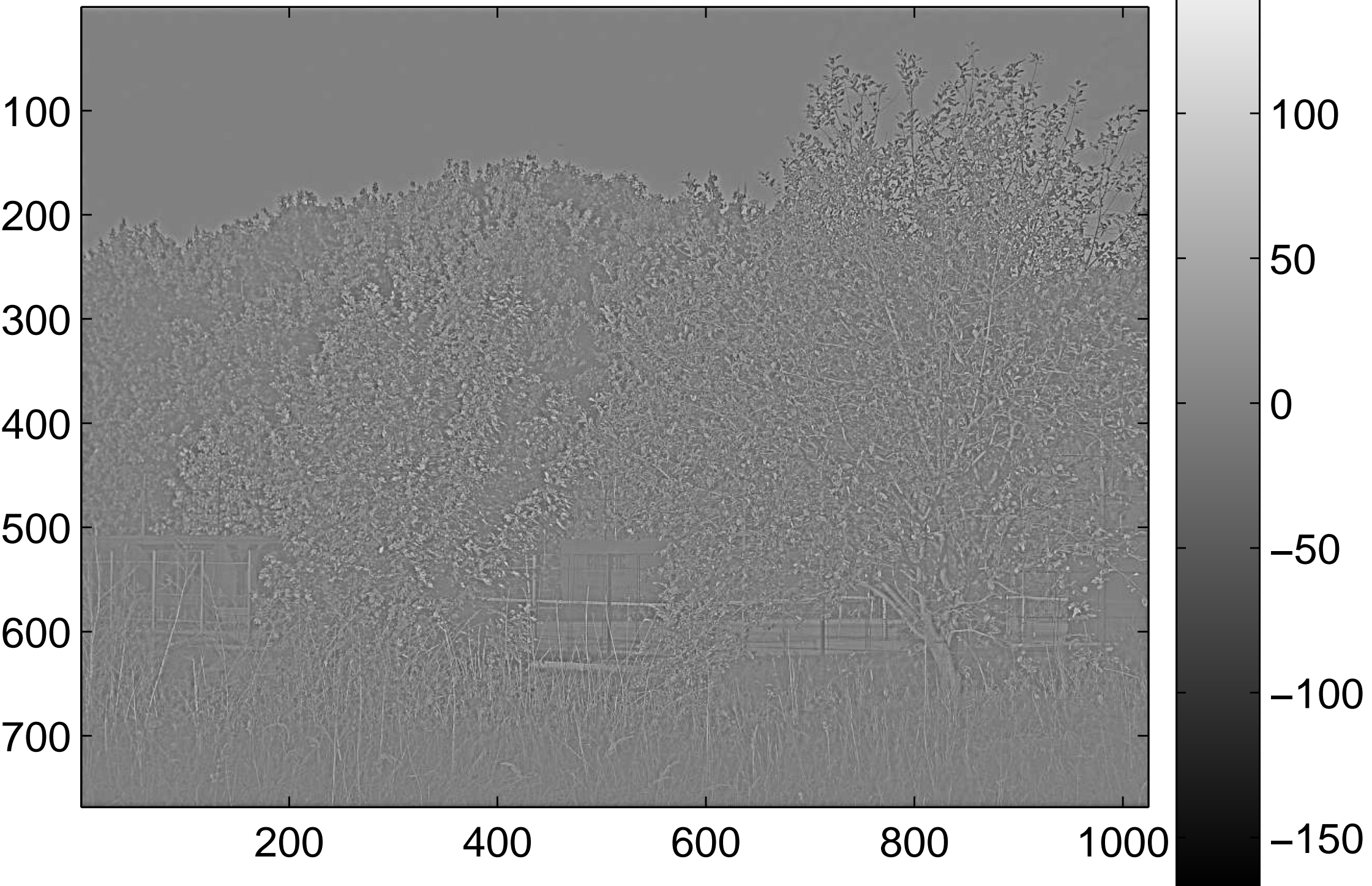
filtered image



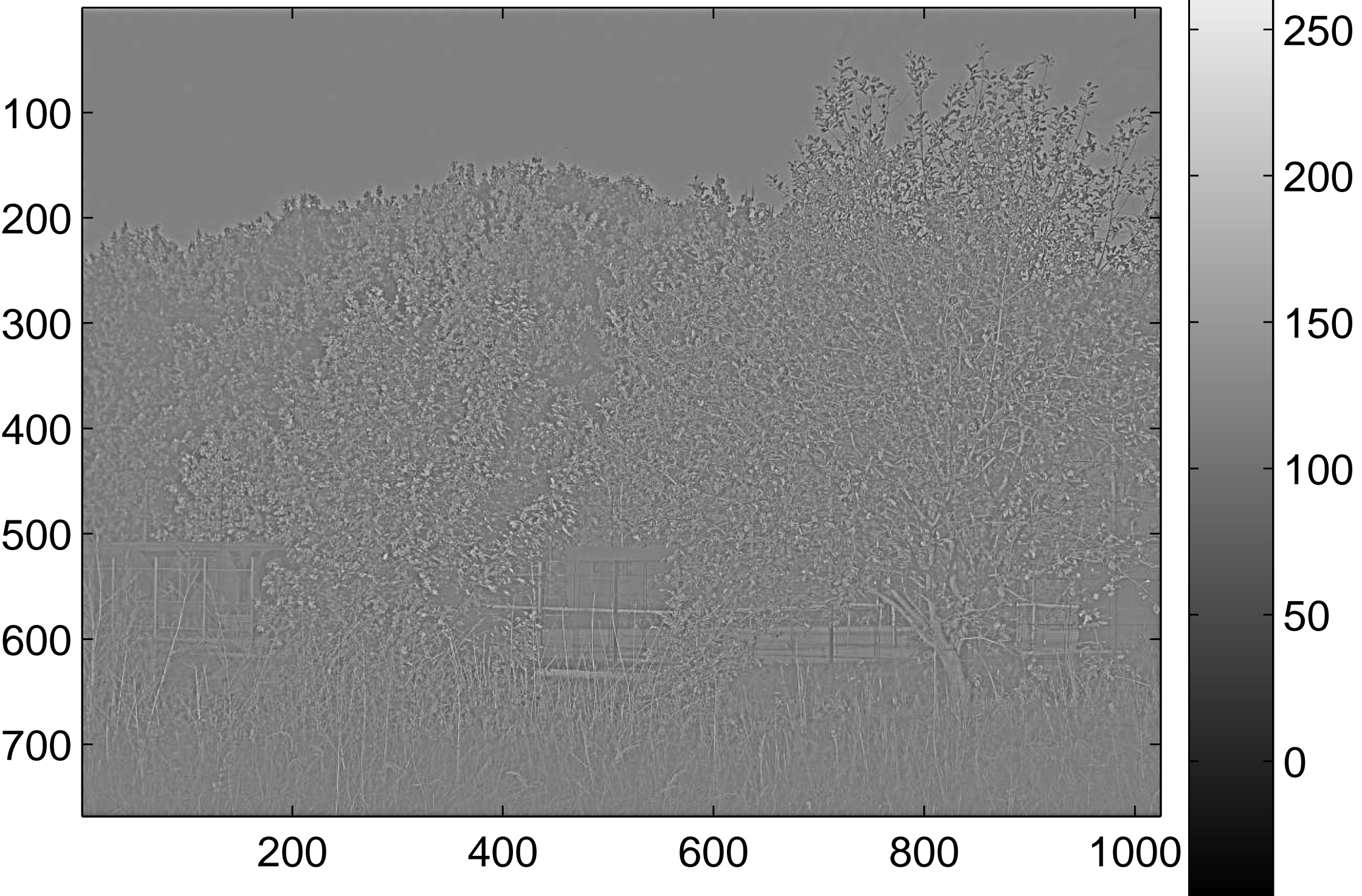
filtered image with added mean (DC)



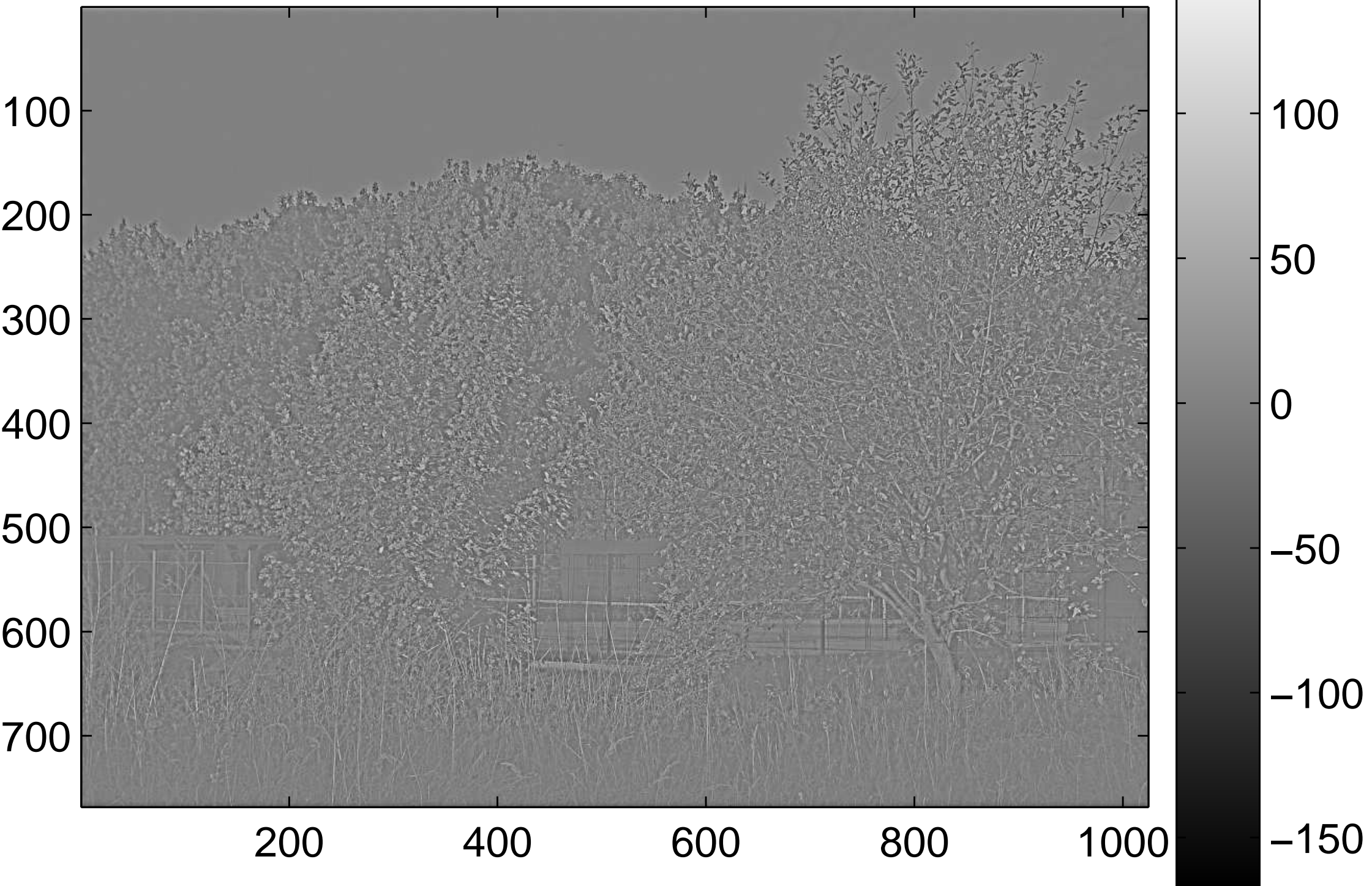
filtered image



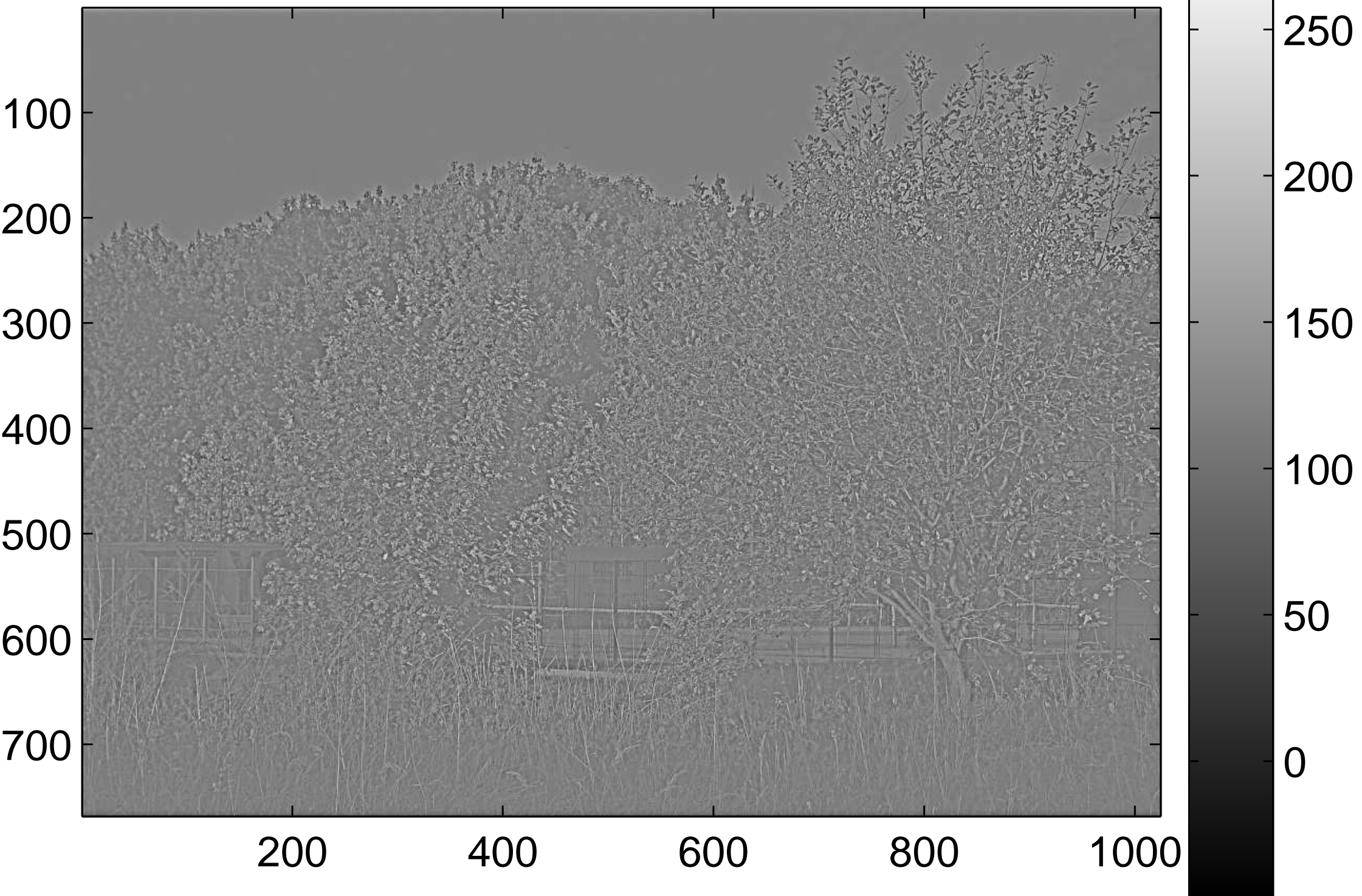
filtered image with added mean (DC)



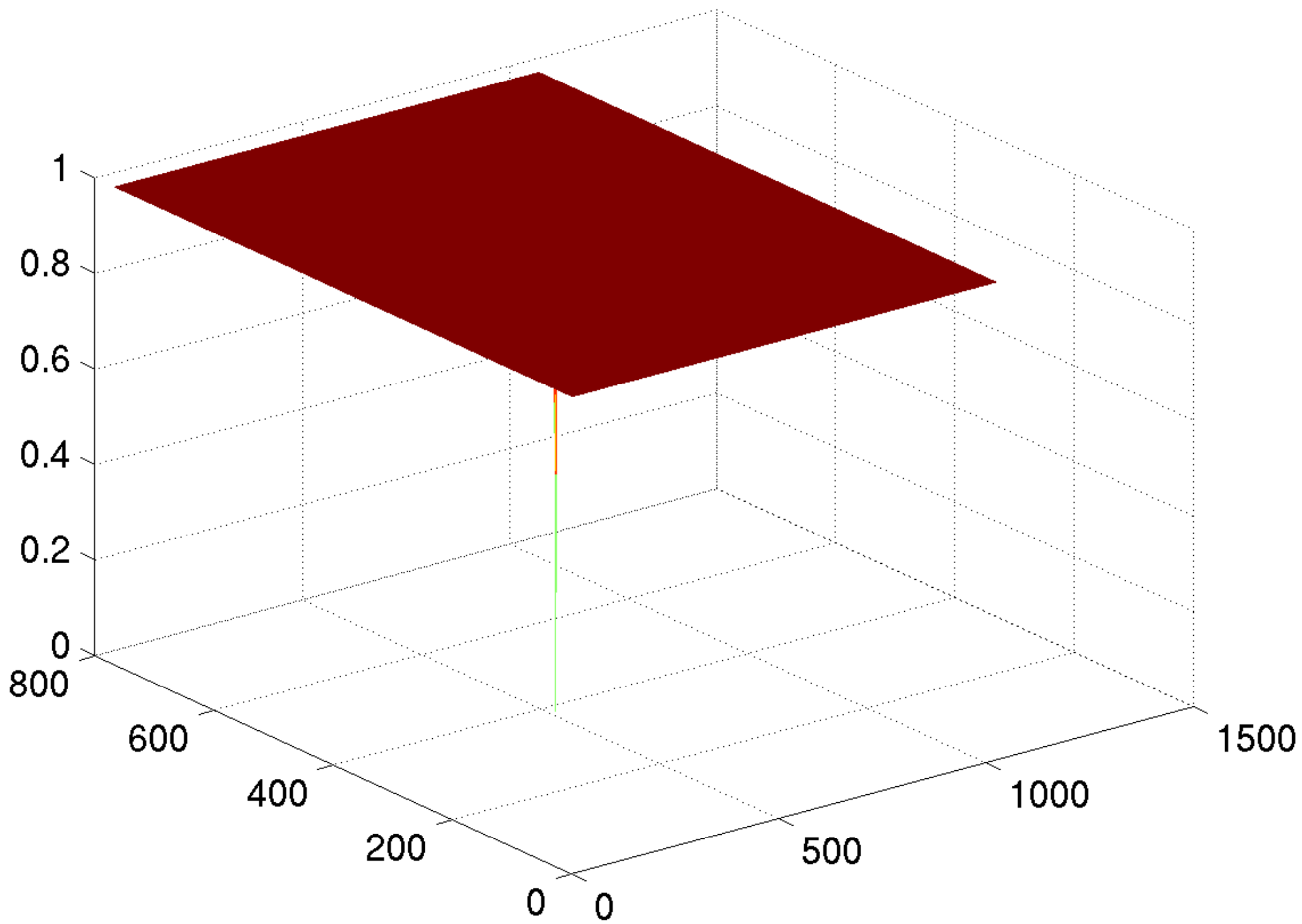
filtered image



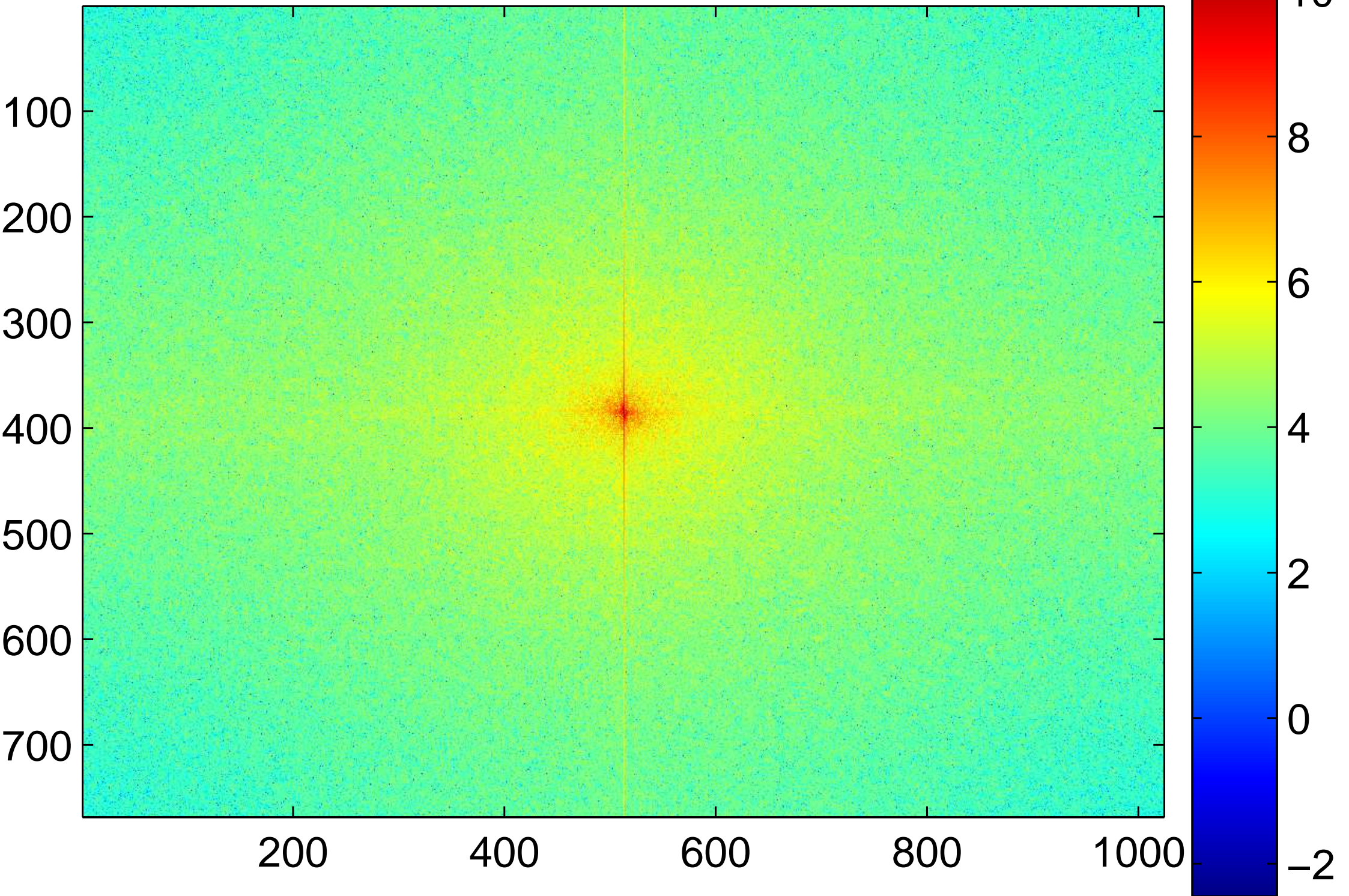
filtered image with added mean (DC)



hp Butt filter n=1, cutoff=1



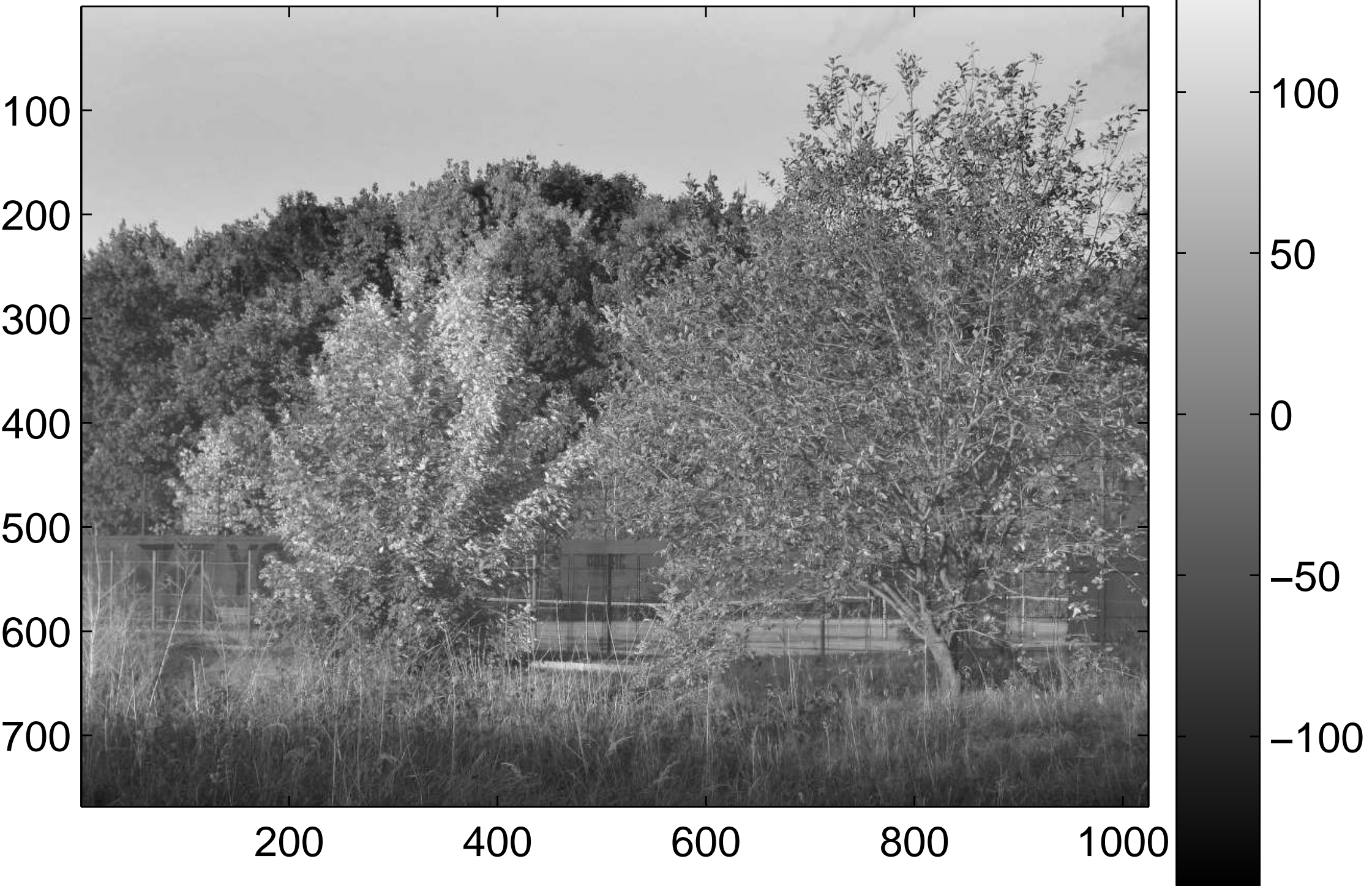
Shifted $\log(\text{abs}(\text{FFT}))$ of the filtered image



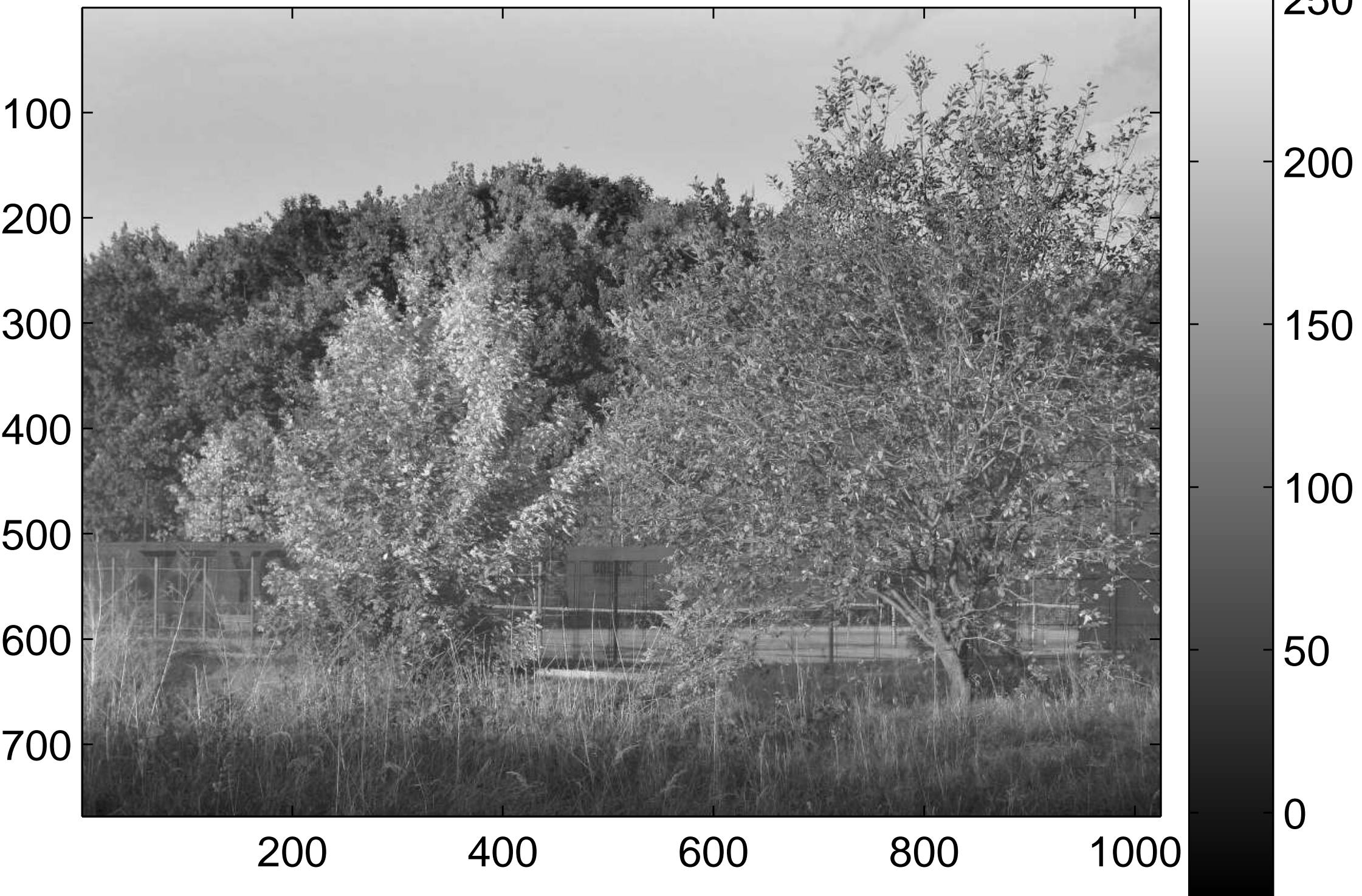




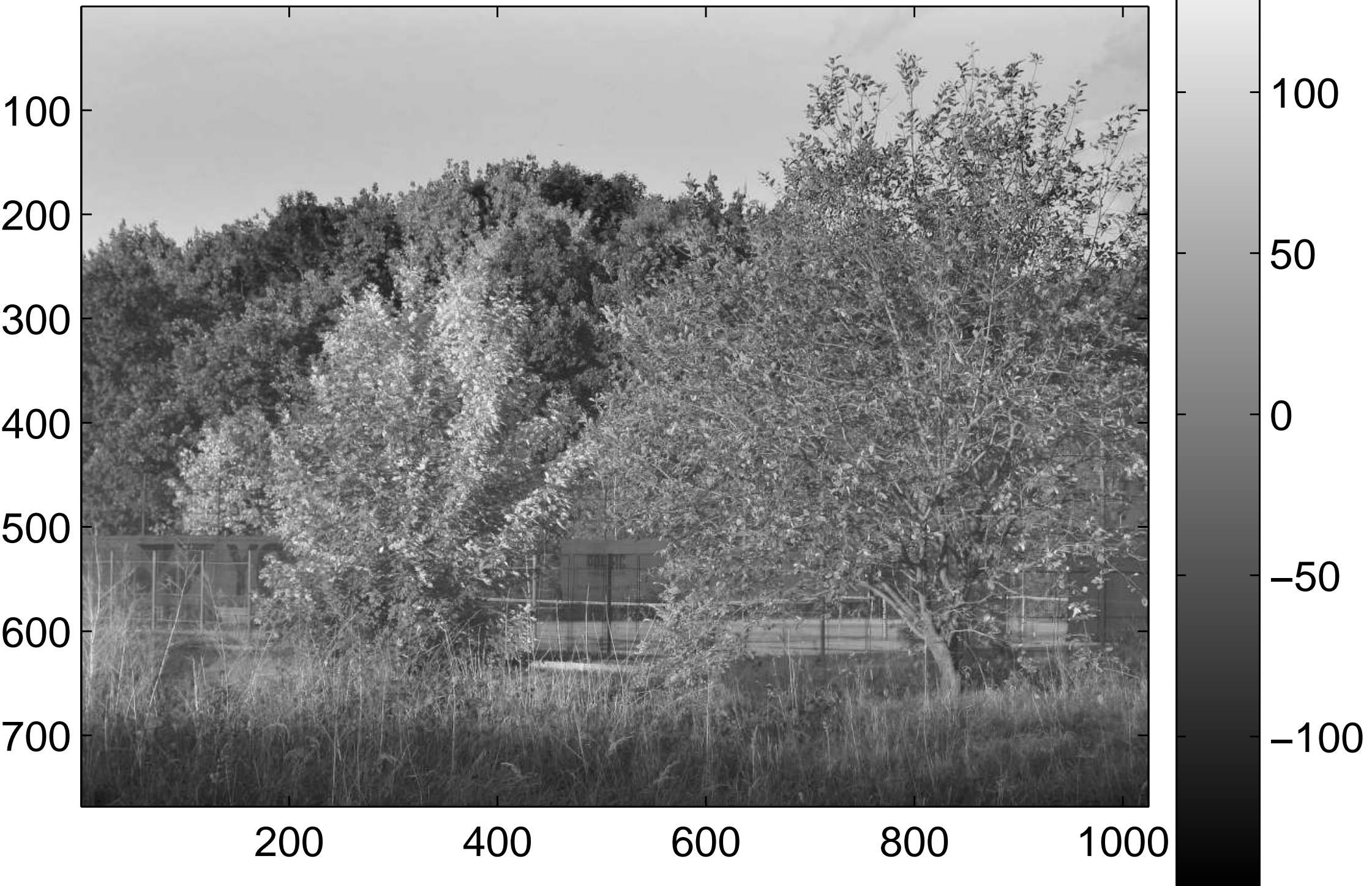
filtered image



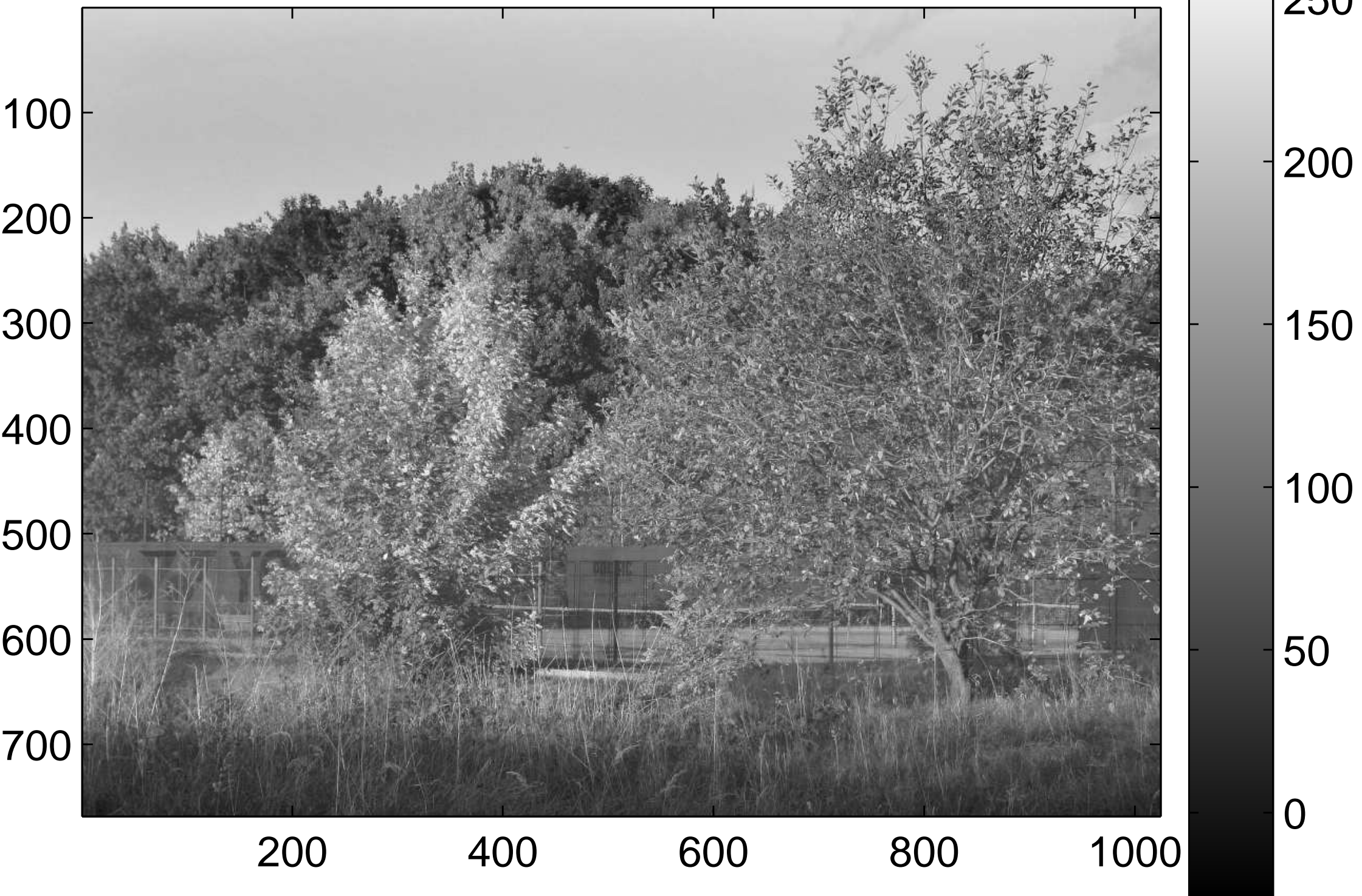
filtered image with added mean (DC)



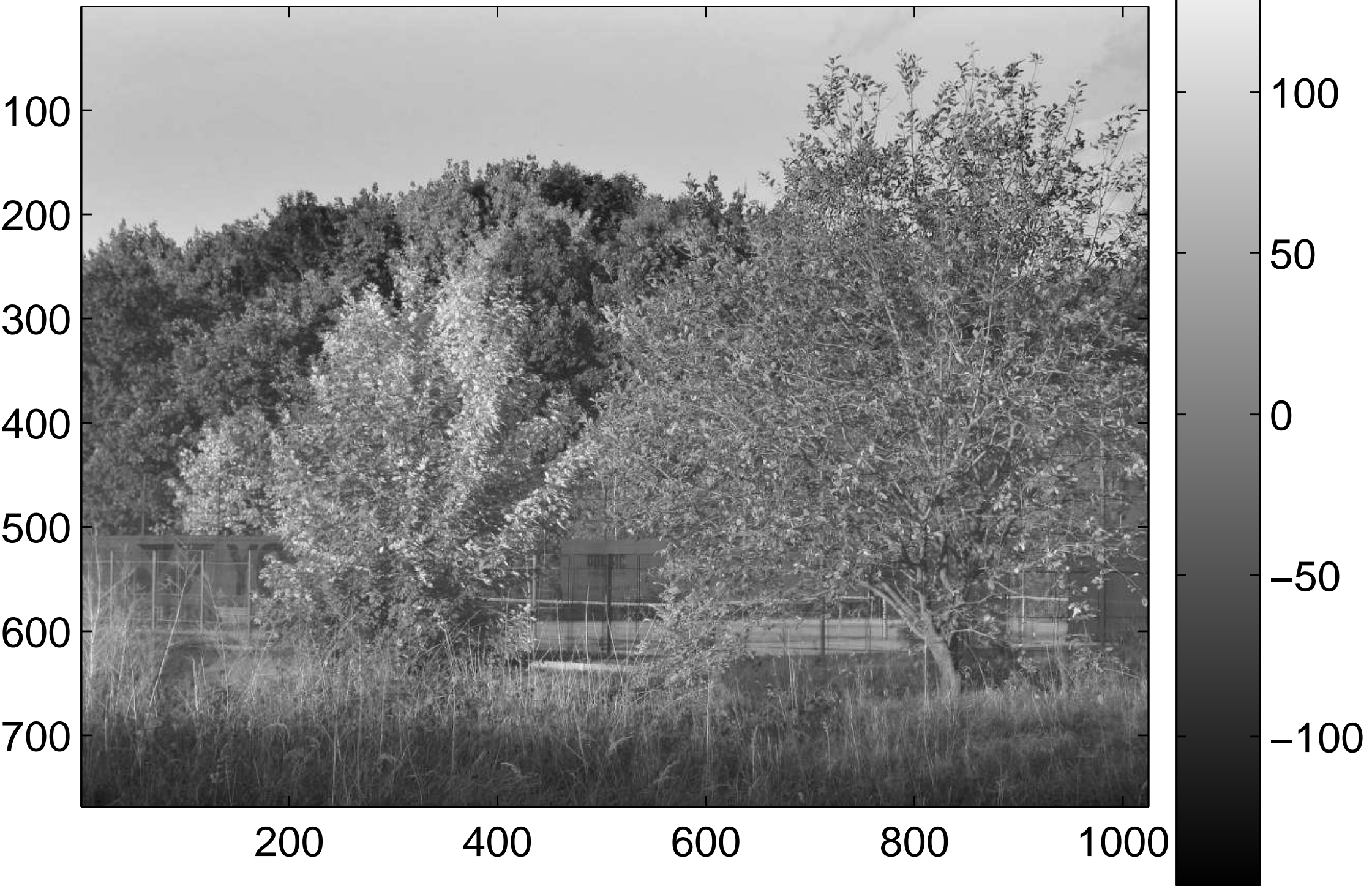
filtered image



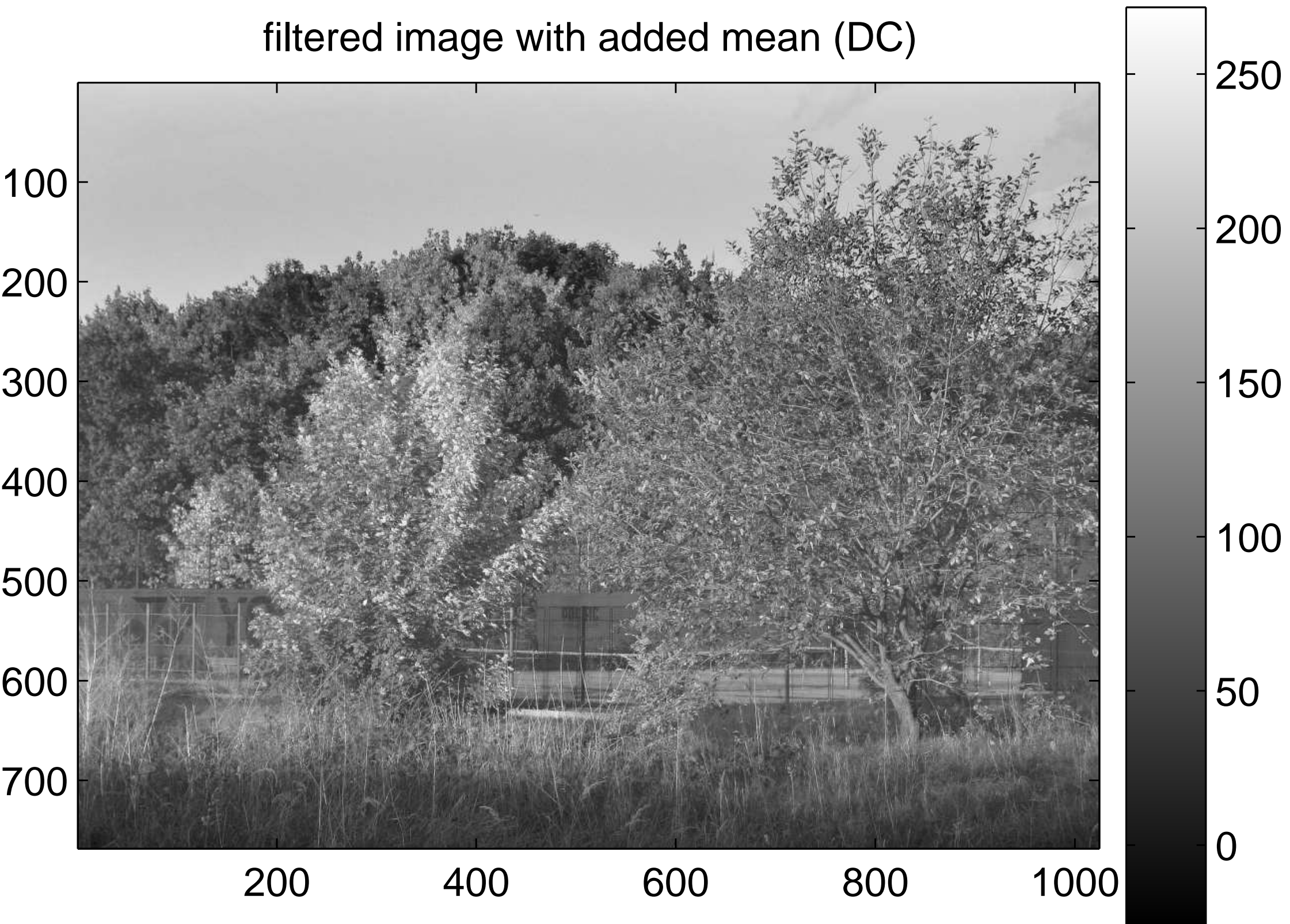
filtered image with added mean (DC)



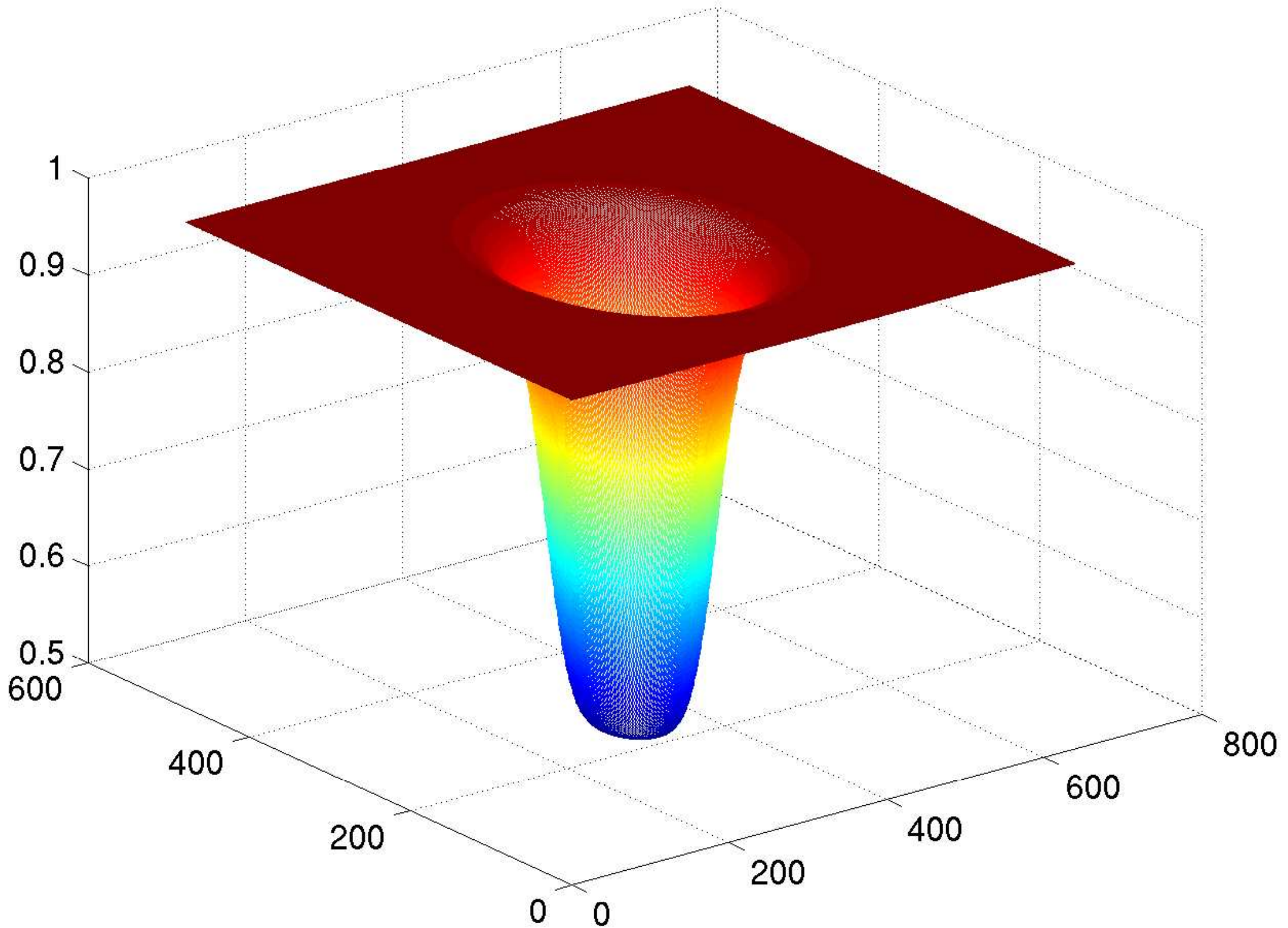
filtered image

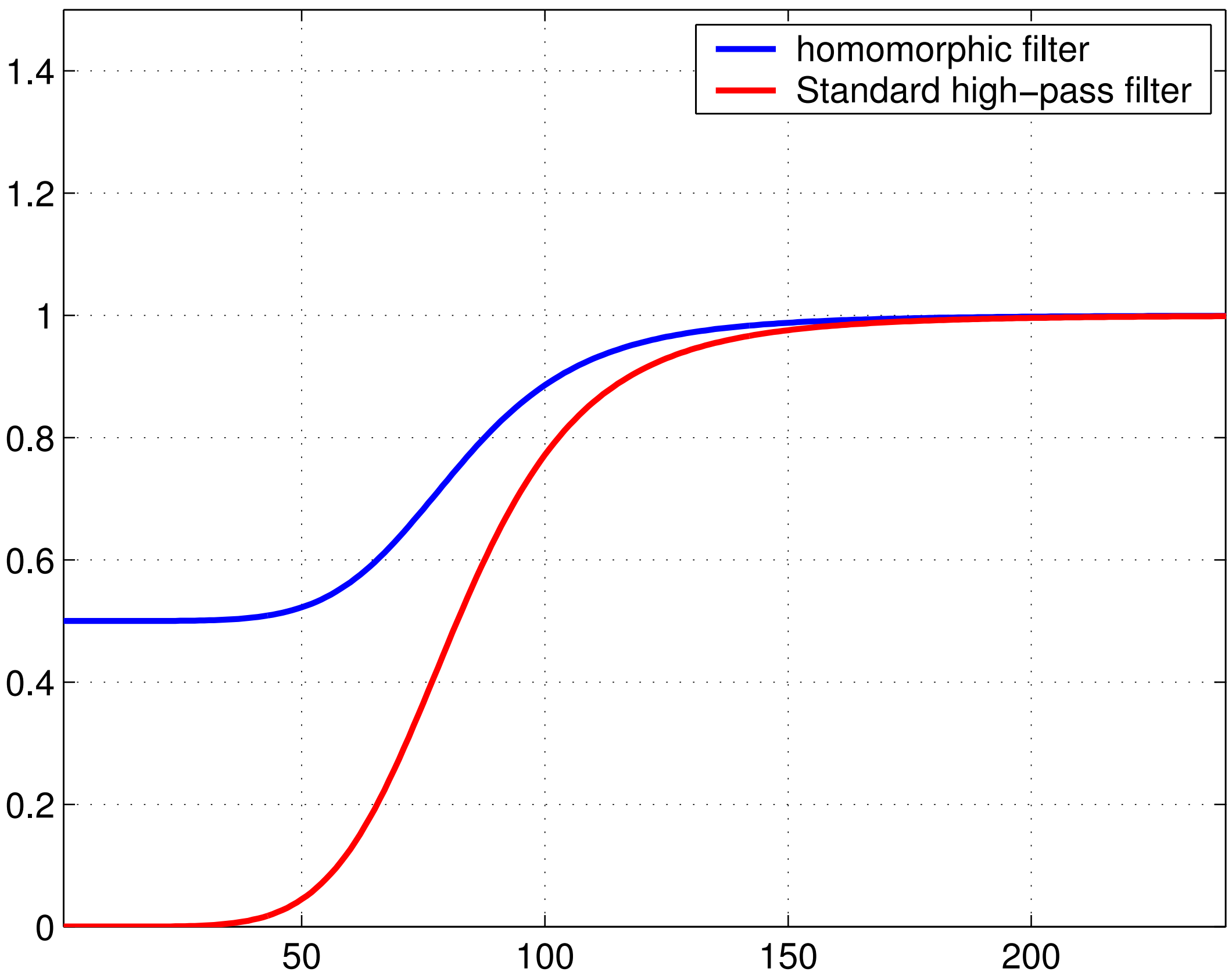


filtered image with added mean (DC)



Homomorphic filter made by adaptation of Butterworth highpass



















**OPPA European Social Fund
Prague & EU: We invest in your future.**
