EECE 4353 Image Processing

Lecture Notes: Frequency Filtering

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2016

Convolution Property of the Fourier Transform

Let functions $\mathbf{f}(r,c)$ and $\mathbf{g}(r,c)$ have Fourier Transforms $\mathrm{F}(u,v)$ and $\mathrm{G}(u,v)$.

Then,

$$\mathfrak{F}\{\mathbf{f} * \mathbf{g}\} = \mathbf{F} \cdot \mathbf{G}.$$

Moreover,

$$\mathfrak{F}\{\mathbf{f} \cdot \mathbf{g}\} = \mathbf{F} * \mathbf{G}.$$

* = convolution
· = multiplication

The Fourier Transform of a convolution equals the product of the Fourier Transforms. Similarly, the Fourier Transform of a product is the convolution of the Fourier Transforms

Convolution via Fourier Transform

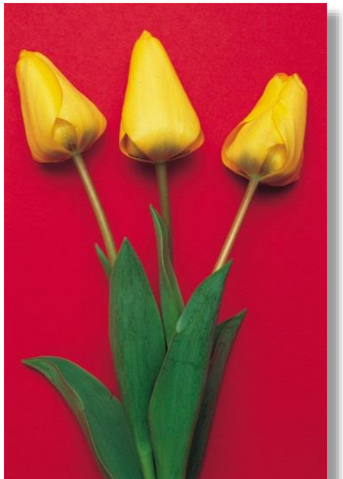
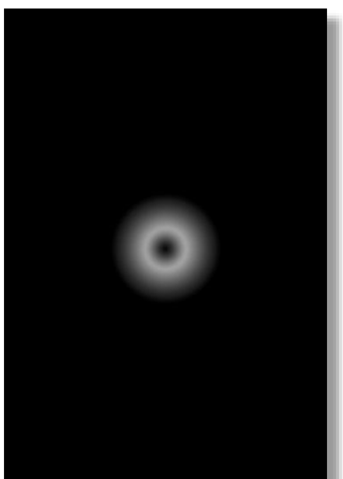
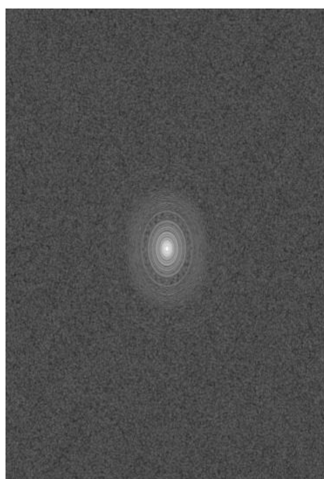


Image & Mask

Transforms





Pixel-wise Product

Inverse Transform

How to Convolve via FT in Matlab

1. Read the image from a file into a variable, say `I`.
2. Read in or create the convolution matrix, `h`.
3. Compute the sum of the matrix: `s = sum(h(:));`
4. If `s == 0,` set `s = 1;`
5. Replace `h` with `h = h/s;`
6. Create: `H = zeros(size(I));`
7. Copy `h` into the middle of `H`.
8. Shift `H` into position: `H = ifftshift(H);`
9. Take the 2D FT of `I` and `H`: `FI=fft2(I); FH=fft2(H);`
10. Pointwise multiply the FTs: `FJ=FI.*FH;`
11. Compute the inverse FT: `J = real(ifft2(FJ));`

The matrix is usually 1-band

If h is a one-band matrix and I is multi-band, you must copy h into all the bands of H.

How to Convolve via FT in Matlab

1. Read the image from a file into a variable, say `I`.
2. Read in or create the convolution matrix, `h`.
3. Compute the sum of the matrix: `s = sum(h(:));`
4. If `s == 0,` set `s = 1;`
5. Replace `h` with `h = h/s;`
6. Create: `H = zeros(size(I));`
7. Copy `h` into the middle of `H`.
8. Shift `H` into position: `H = ifftshift(H);`
9. Take the 2D FT of `I` and `H`: `FI=fft2(I); FH=fft2(H);`
10. Pointwise multiply the FTs: `FJ=FI.*FH;`
11. Compute the inverse FT: `J = real(ifft2(FJ));`

fftshift and ifftshift must be done separately for each band. fft2 transforms all the bands of a multiband image separately.

Coordinate Origin of the FFT

Center = (floor(R/2)+1, floor(C/2)+1)

Even

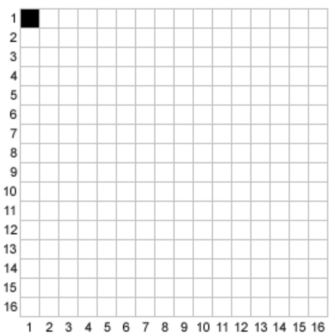


Image Origin

Odd

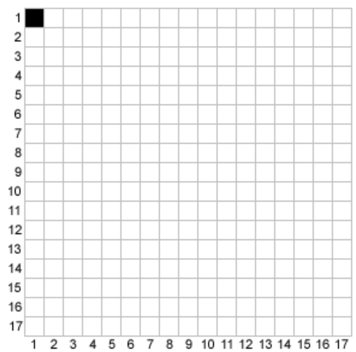
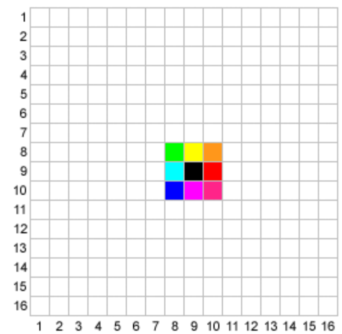


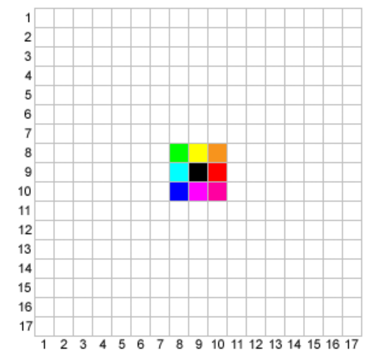
Image Origin

Even

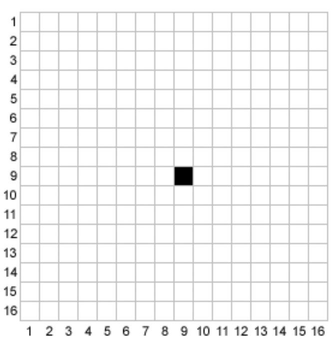


Weight Matrix Origin

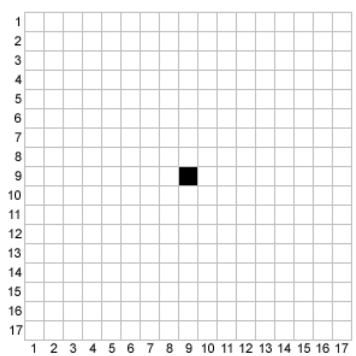
Odd



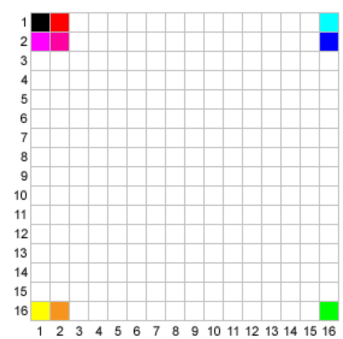
Weight Matrix Origin



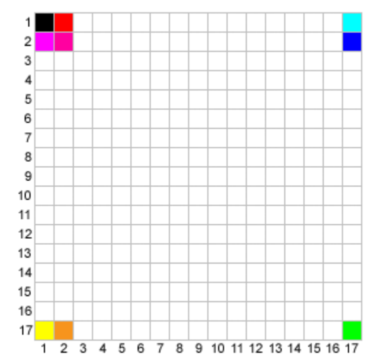
After FFT shift



After FFT shift



After IFFT shift

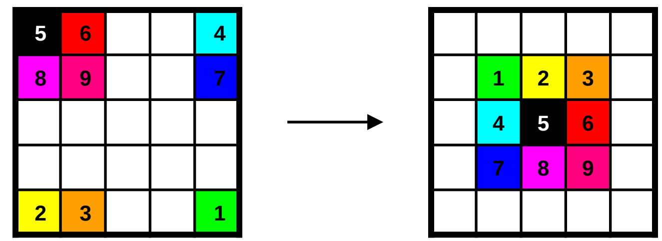


After IFFT shift

Matlab's `fftshift` and `ifftshift`

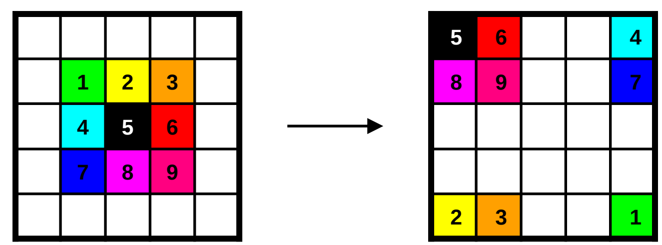
`J = fftshift(I):`

$$\mathbf{I}\,(1,1) \rightarrow \mathbf{J}\,(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1)$$



`I = ifftshift(J):`

$$\mathbf{J}\,(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1) \rightarrow \mathbf{I}\,(1,1)$$



where $\lfloor x \rfloor$ = `floor(x)` = the largest integer smaller than x.

Blurring: Averaging / Lowpass Filtering

Blurring results from:

- Pixel averaging in the spatial domain:
  - Each pixel in the output is a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 1.
- Lowpass filtering in the frequency domain:
  - High frequencies are diminished or eliminated
  - Individual frequency components are multiplied by a non-increasing function of ω such as $1/\omega = 1/\sqrt{(u^2+v^2)}$.

The values of the output image are all non-negative.

Sharpening: Differencing / Highpass Filtering

Sharpening results from adding to the image, a copy of itself that has been:

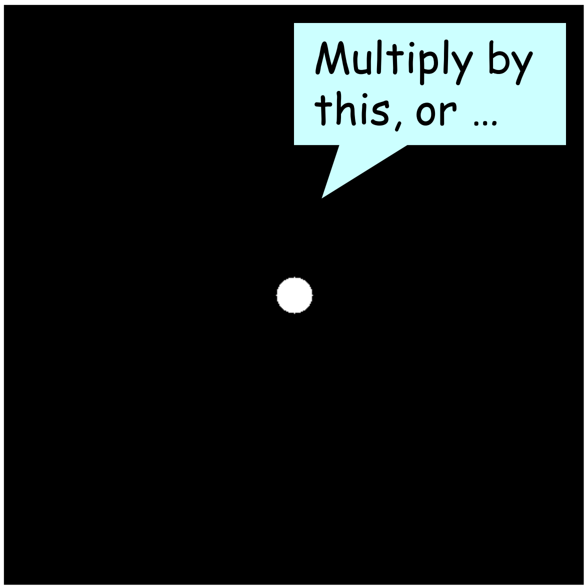
- Pixel-differenced in the spatial domain:
  - Each pixel in the output is a difference between itself and a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 0.
- Highpass filtered in the frequency domain:
  - High frequencies are enhanced or amplified.
  - Individual frequency components are multiplied by an increasing function of ω such as $\alpha\omega = \alpha\sqrt{(u^2+v^2)}$, where α is a constant.

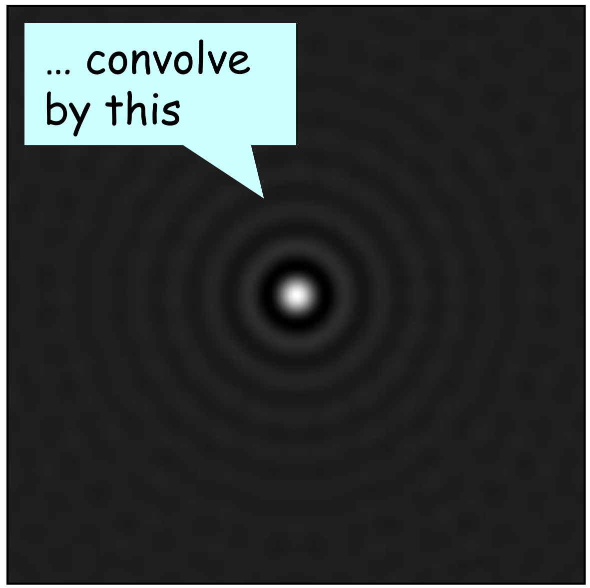
The values of the output image could be negative.

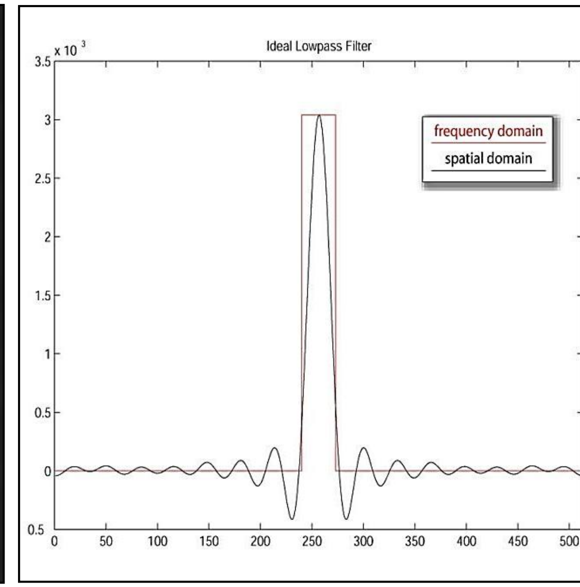
Ideal Lowpass Filter

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 16







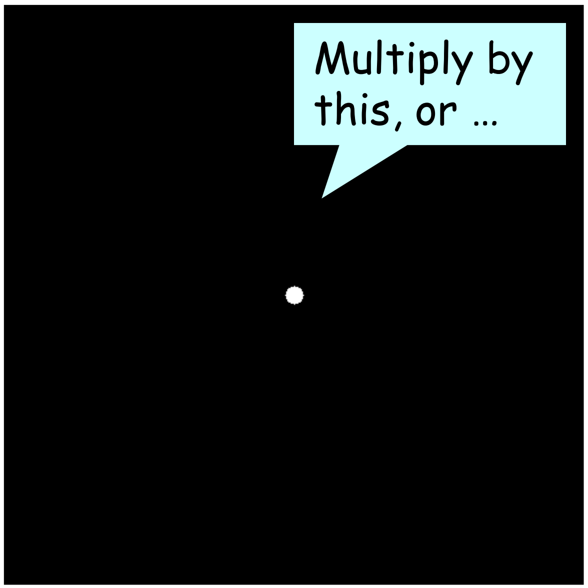
Fourier Domain Rep.

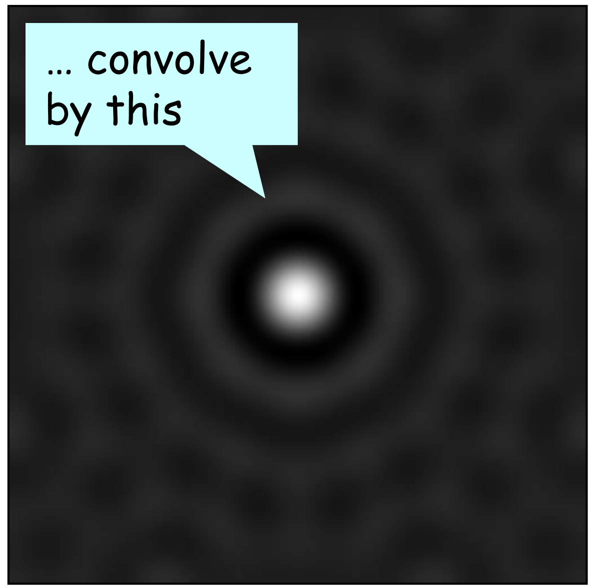
Spatial Representation

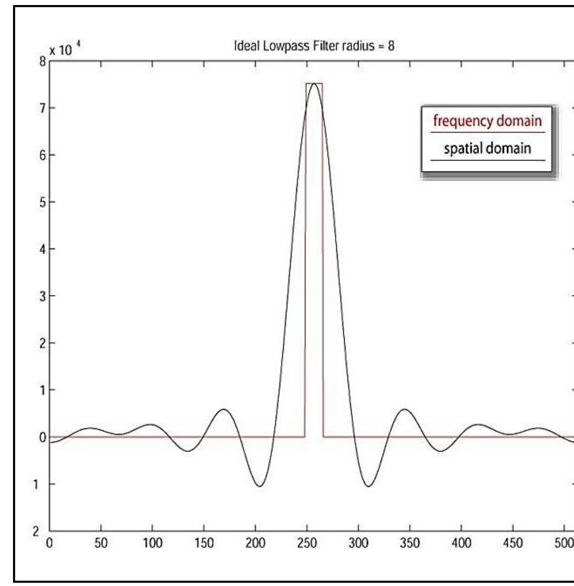
Central Profile

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 8





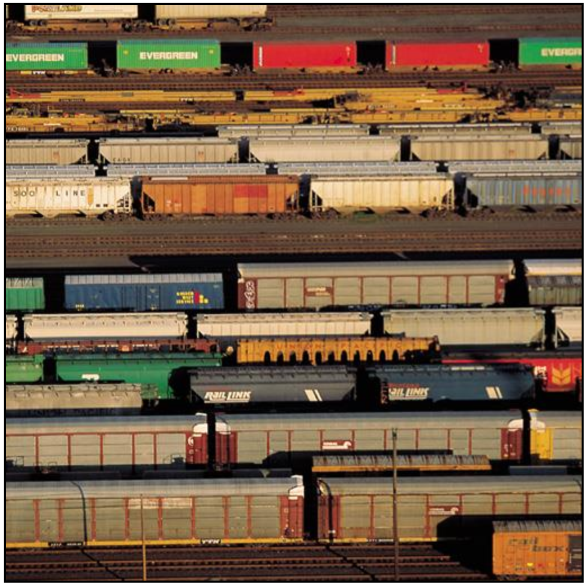


Fourier Domain Rep.

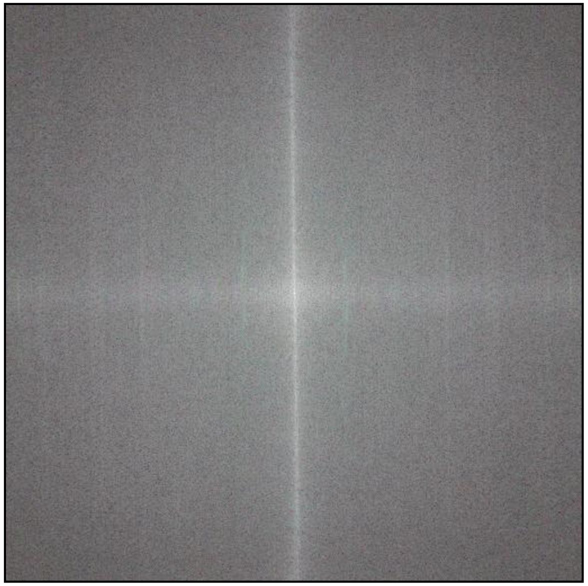
Spatial Representation

Central Profile

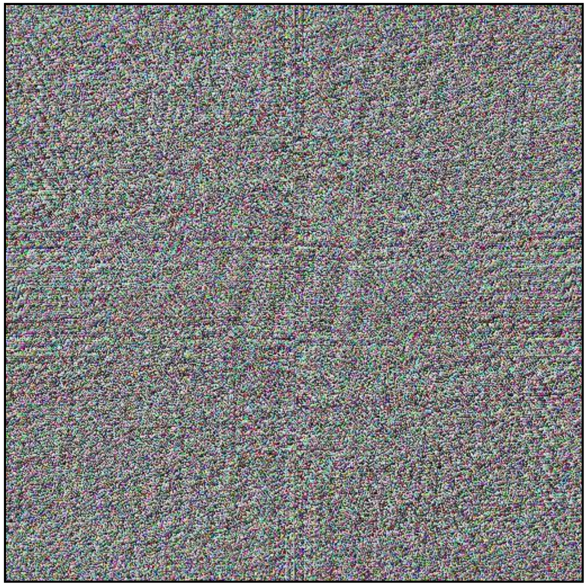
Power Spectrum and Phase of an Image



Original Image



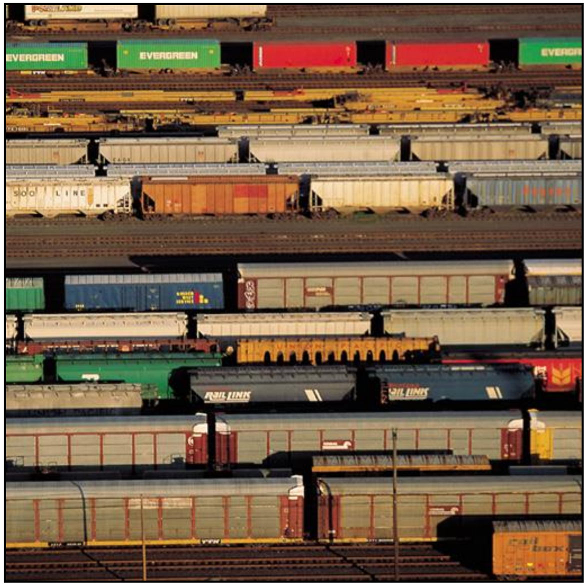
Power Spectrum



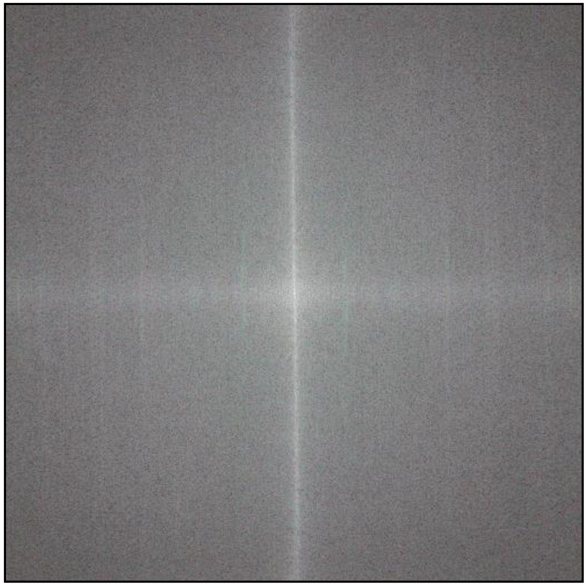
Phase

Ideal Lowpass Filter

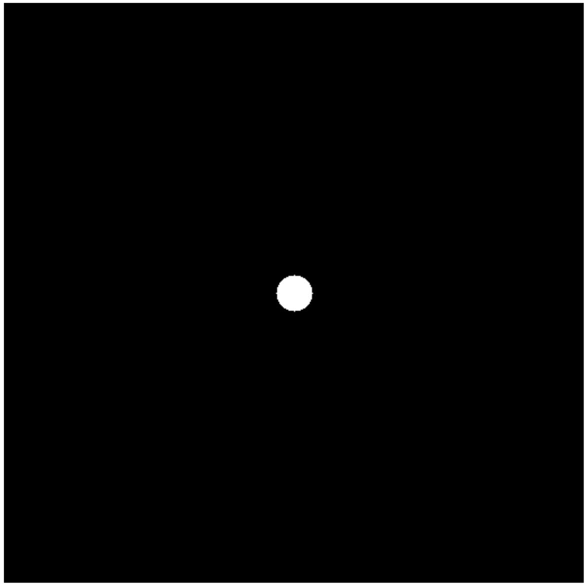
Image size: 512x512
FD filter radius: 16



Original Image



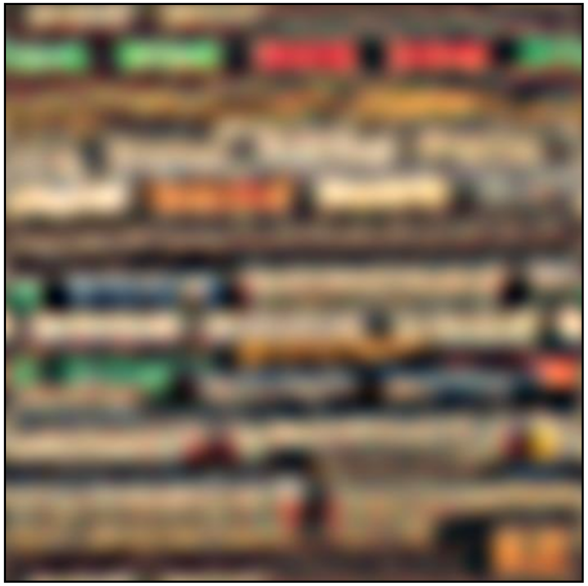
Power Spectrum



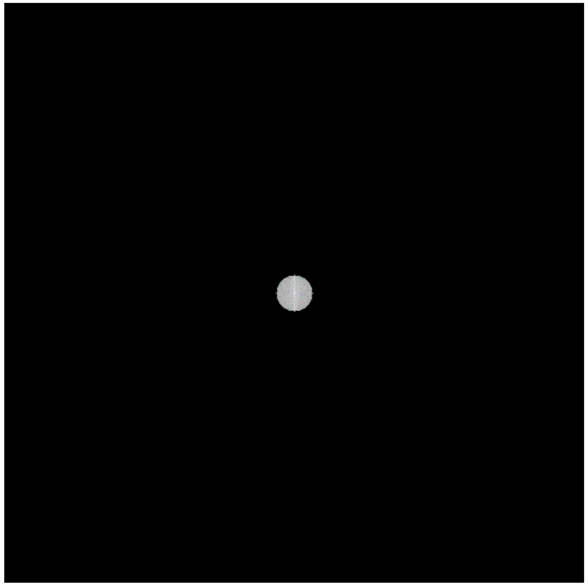
Ideal LPF in FD

Ideal Lowpass Filter

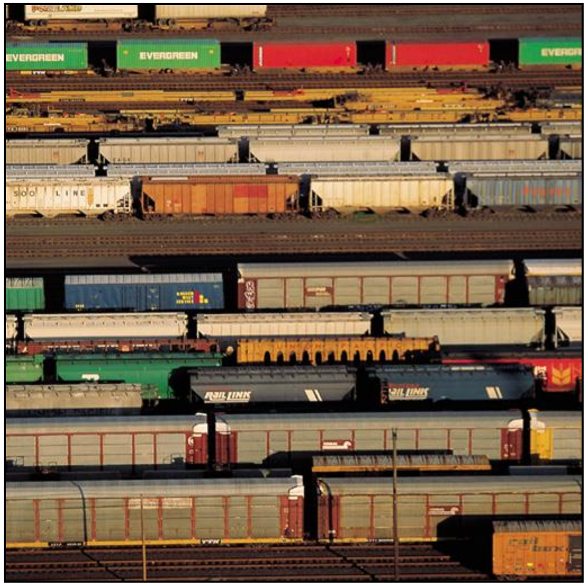
Image size: 512x512
FD filter radius: 16



Filtered Image



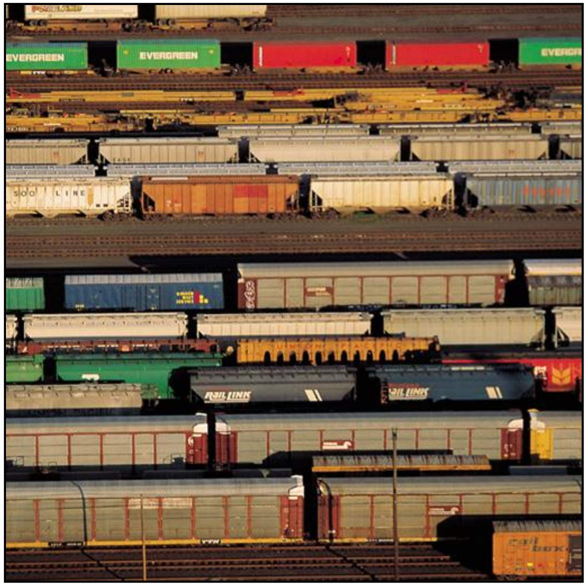
Filtered Power Spectrum

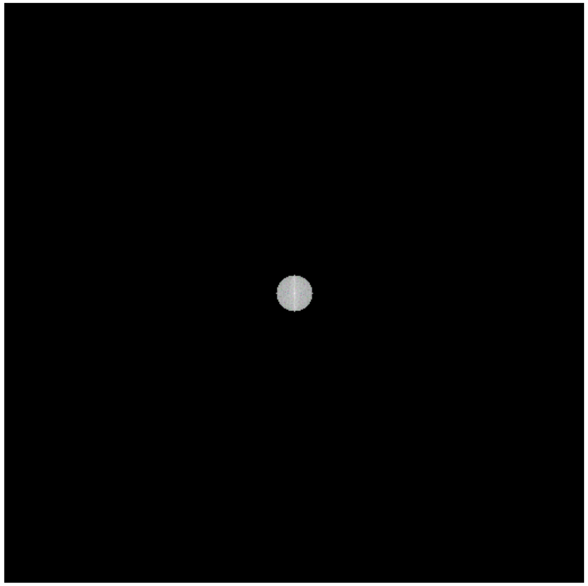


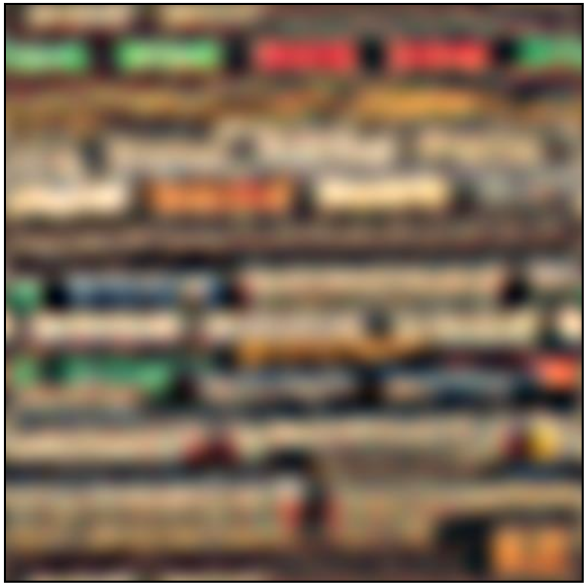
Original Image

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 16







Original Image

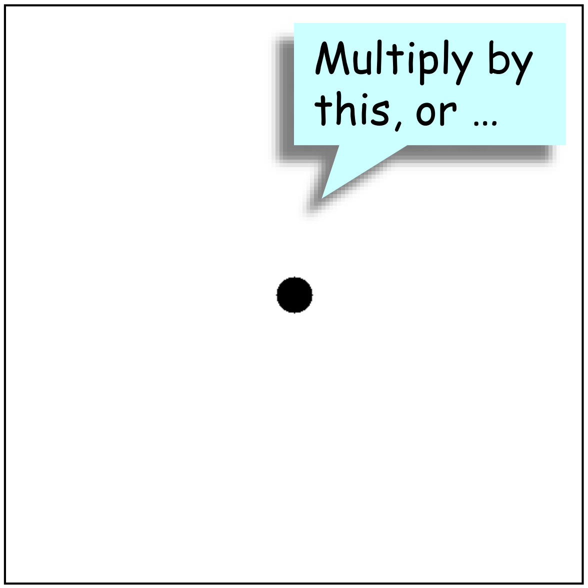
Filtered Power Spectrum

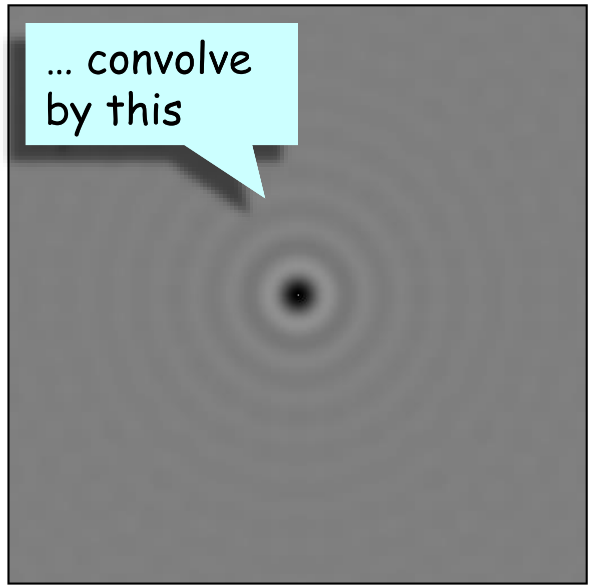
Filtered Image

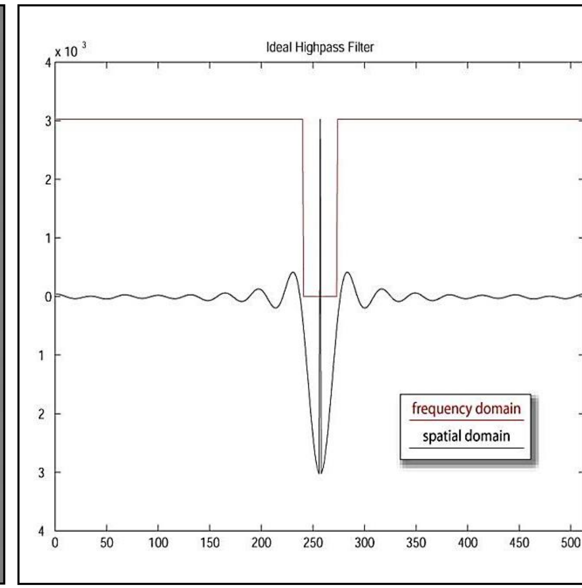
Ideal Highpass Filter

Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16







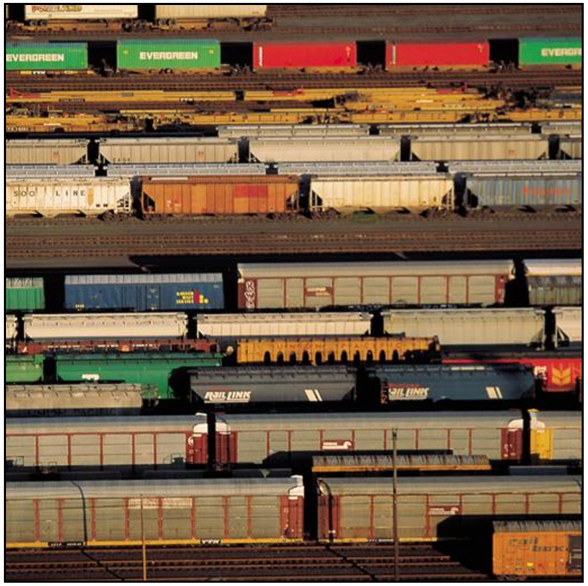
Fourier Domain Rep.

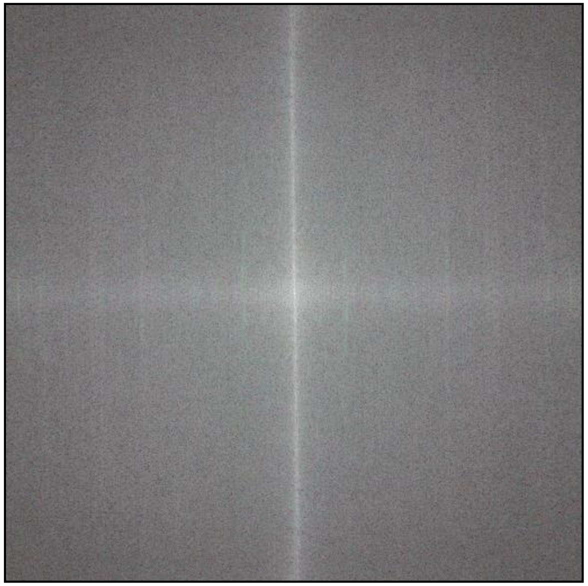
Spatial Representation

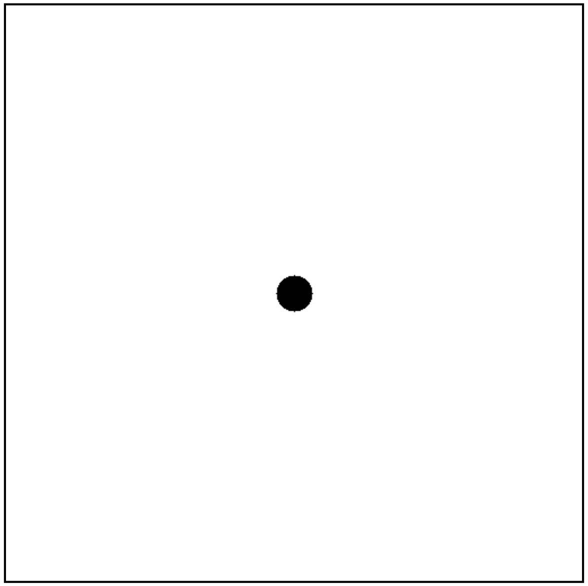
Central Profile

Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16







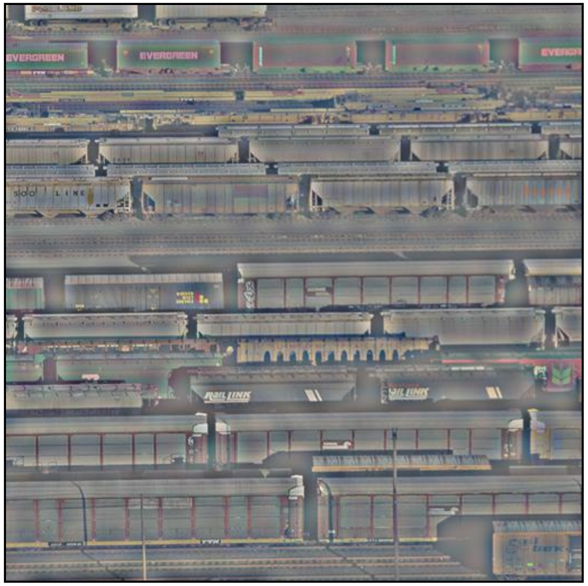
Original Image

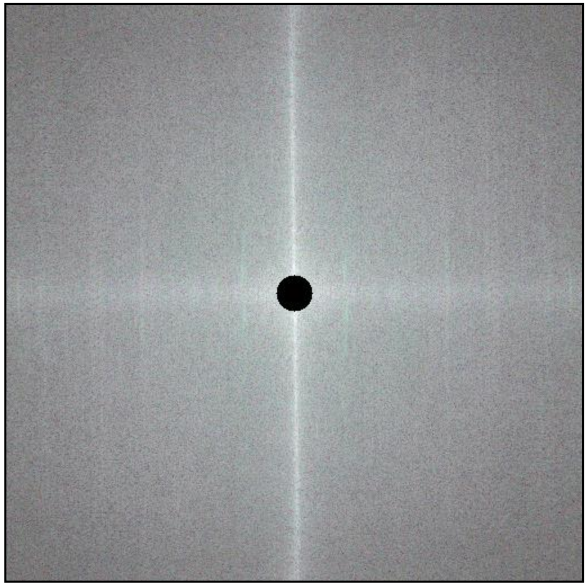
Power Spectrum

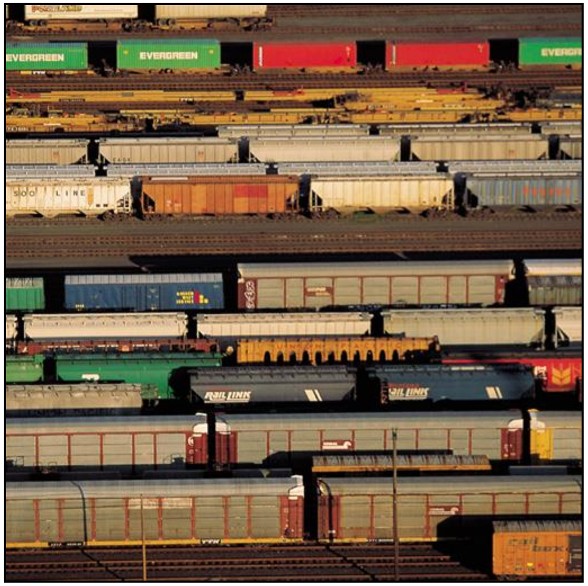
Ideal HPF in FD

Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16







Filtered Image*

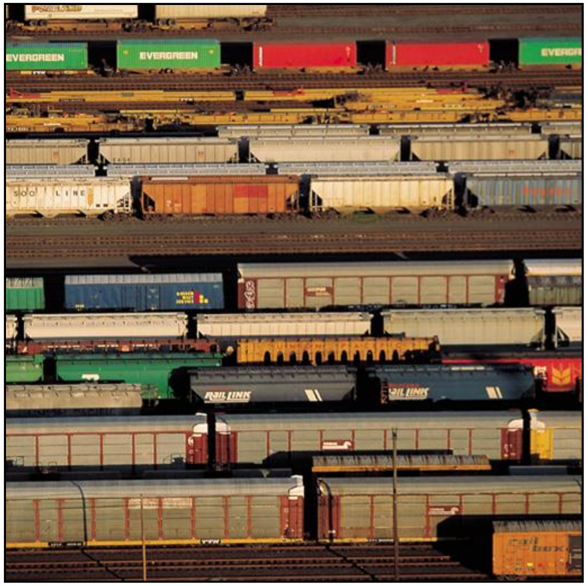
Filtered Power Spectrum

Original Image

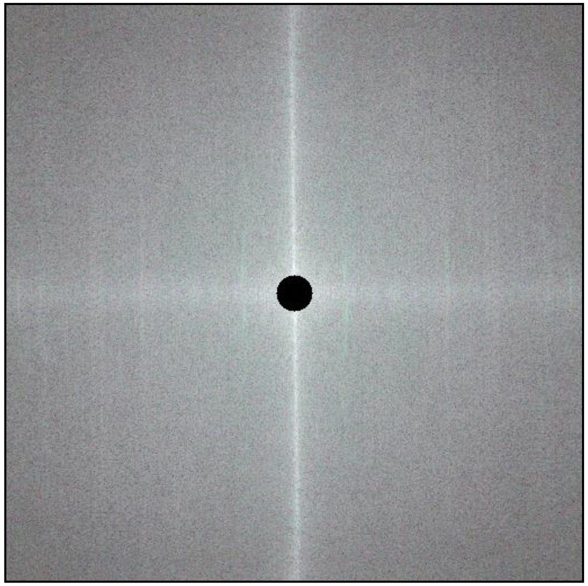
*signed image:
0 mapped to 128

Ideal Highpass Filter

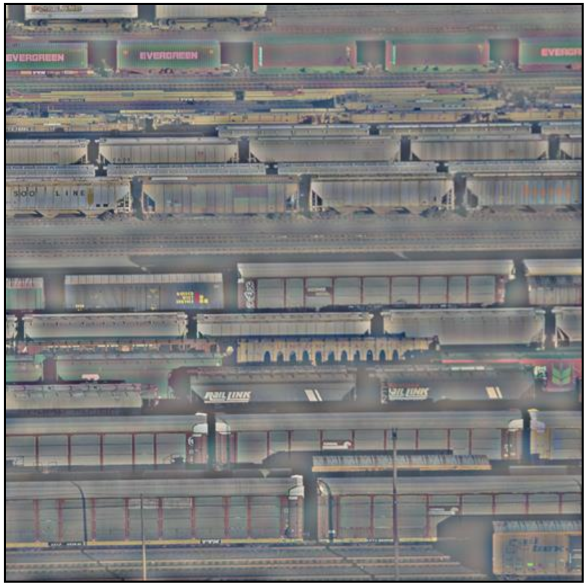
Image size: 512x512
FD notch radius: 16



Original Image



Filtered Power Spectrum



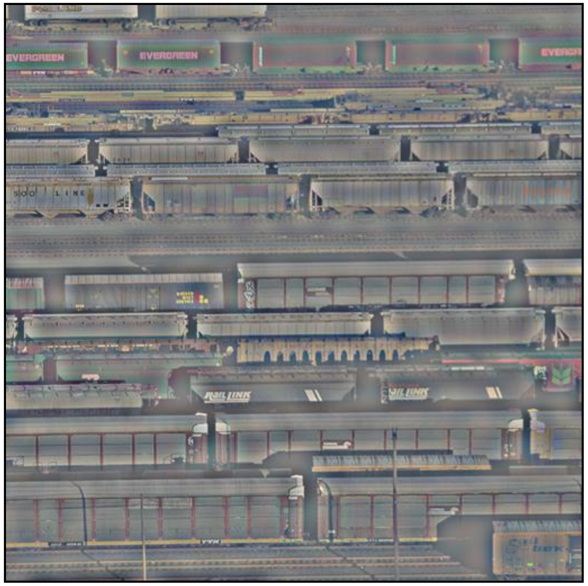
Filtered Image*

*signed image:
0 mapped to 128

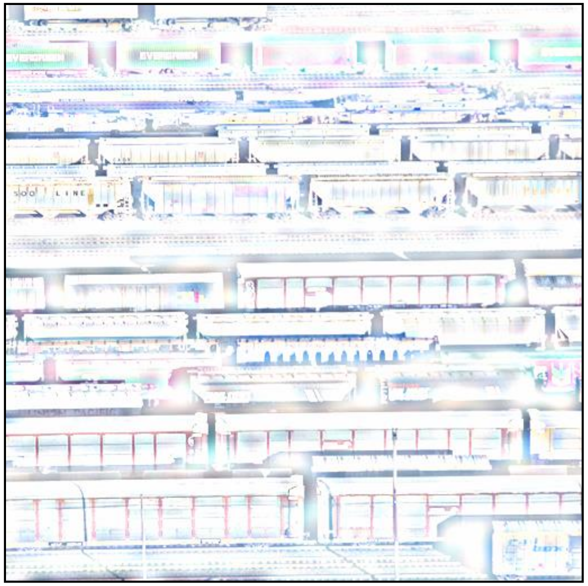
Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16

Positive Pixels



Filtered Image*



Negative Pixels

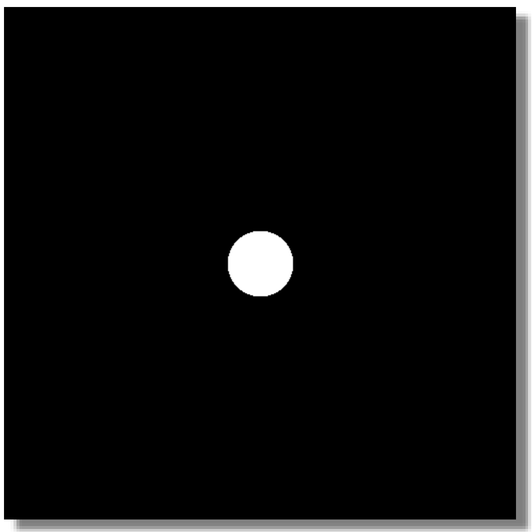
Ideal Bandpass Filter

Ideal Bandpass Filter

A bandpass filter is created by

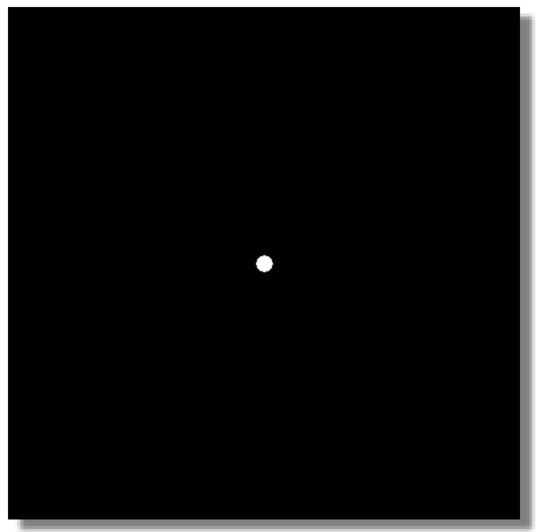
(1) subtracting a FD radius ρ_2 lowpass filtered image from a FD radius ρ_1 lowpass filtered image, where $\rho_2 < \rho_1$, or

(2) convolving the image with a matrix that is the difference of the two lowpass matrixs.



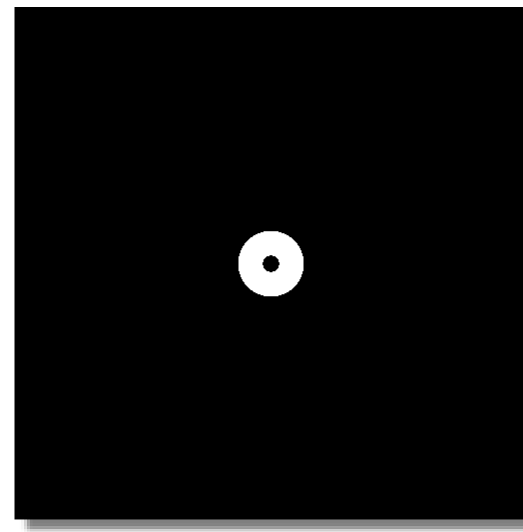
FD LP matrix with radius ρ_1

\-



FD LP matrix with radius ρ_2

=



FD BP matrix ρ_1 - ρ_2

Ideal Bandpass Filter

image LPF radius ρ1

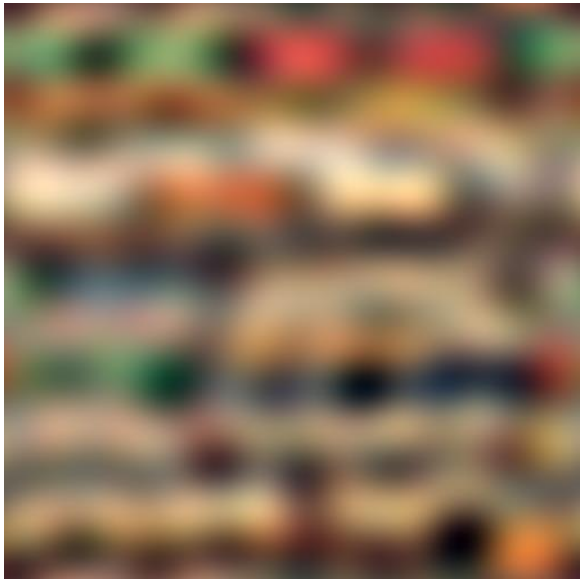


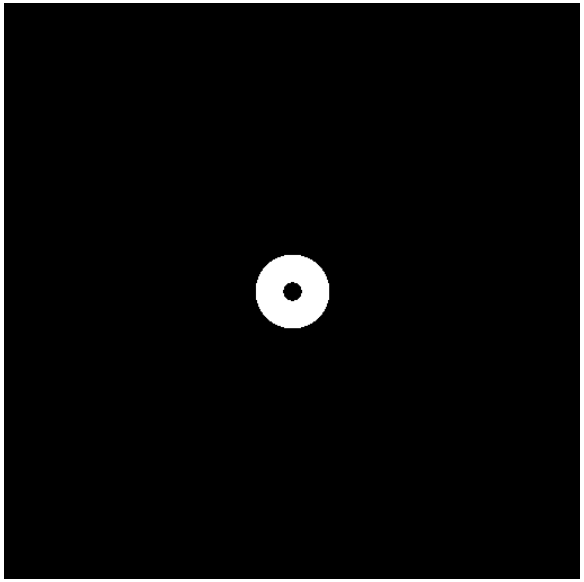
image LPF radius ρ2

image BPF radii ρ1, ρ2*

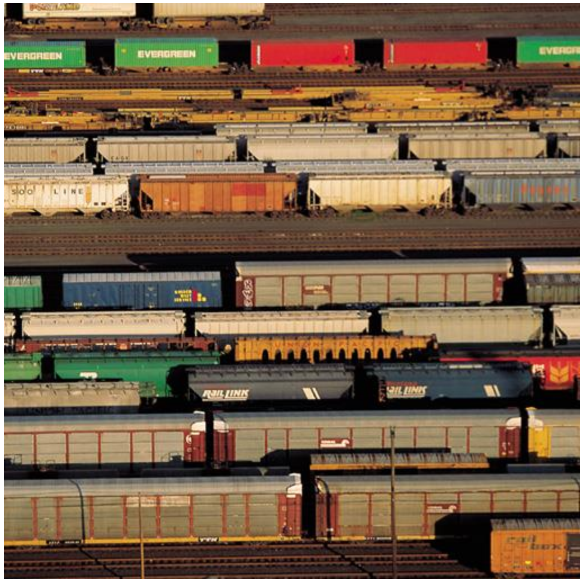
*signed image:
0 mapped to 128

Ideal Bandpass Filter

filtered image*



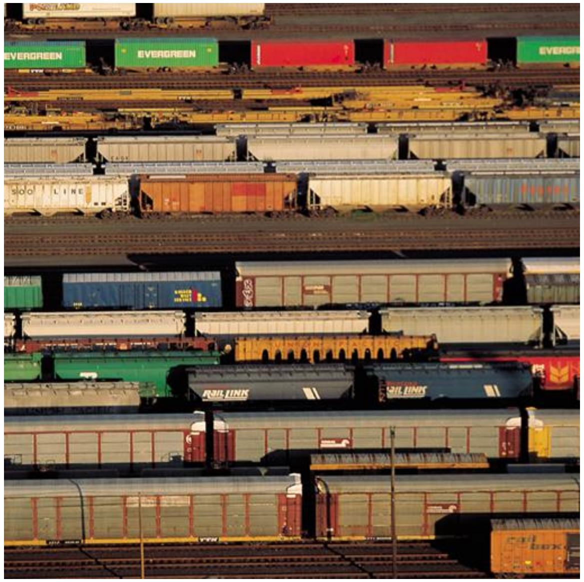
filter power spectrum



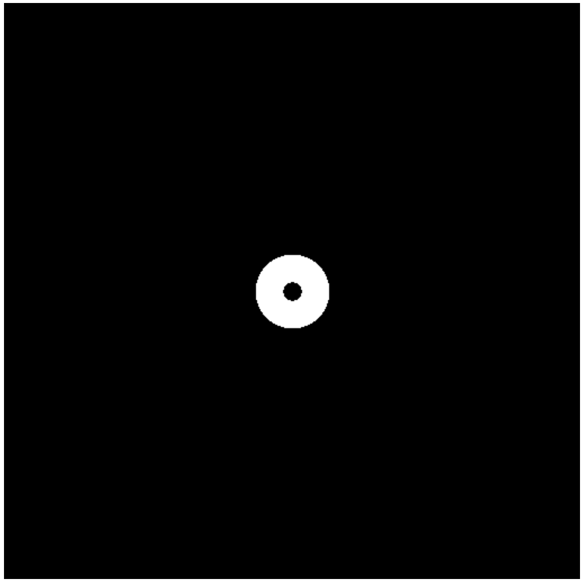
original image

*signed image:
0 mapped to 128

Ideal Bandpass Filter



original image

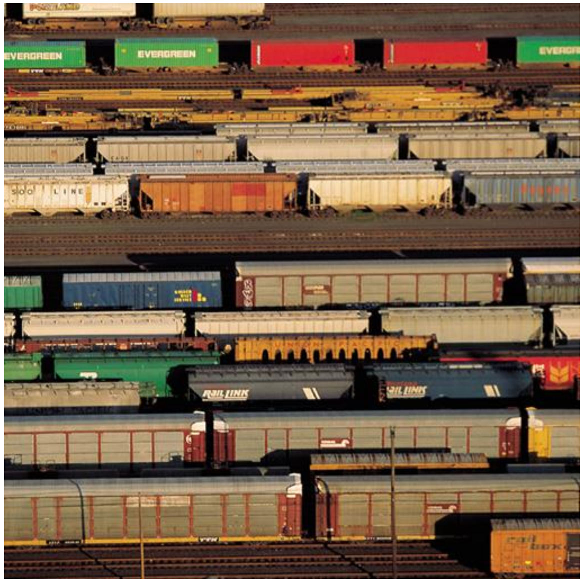


filter power spectrum

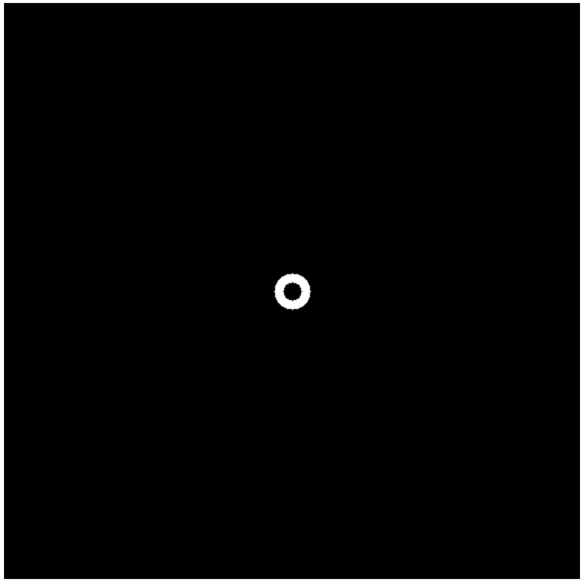
filtered image*

*signed image:
0 mapped to 128

A Different Ideal Bandpass Filter



original image

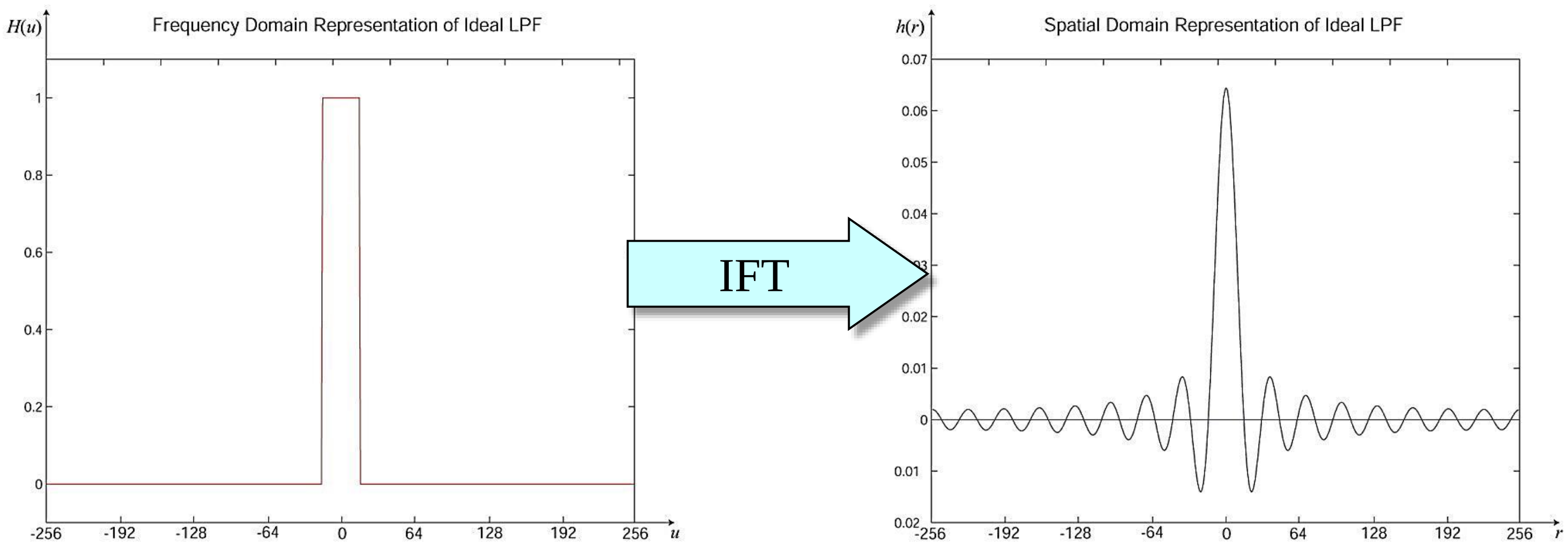


filter power spectrum

filtered image*

The Optimal Filter

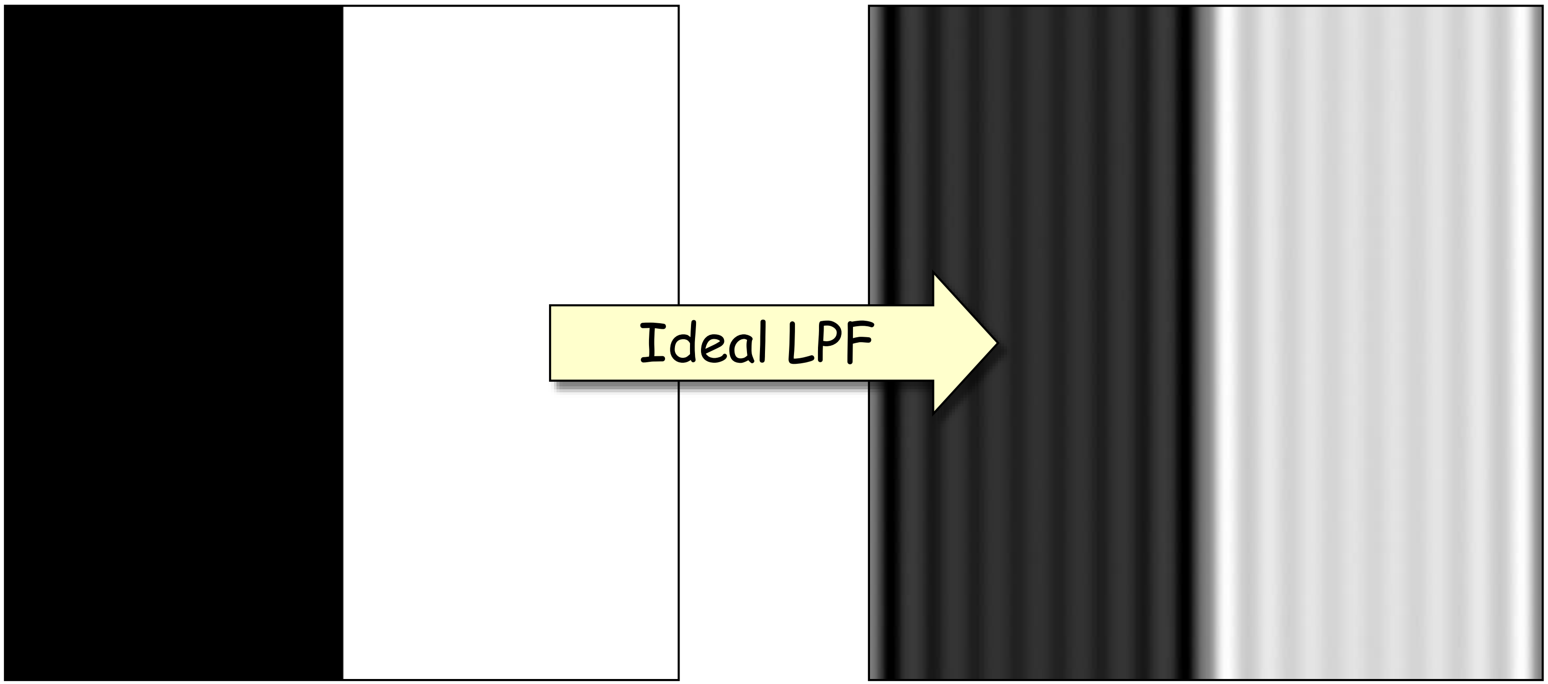
Ideal Filters Do Not Produce Ideal Results



A sharp cutoff in the frequency domain…

…causes ringing in the spatial domain.

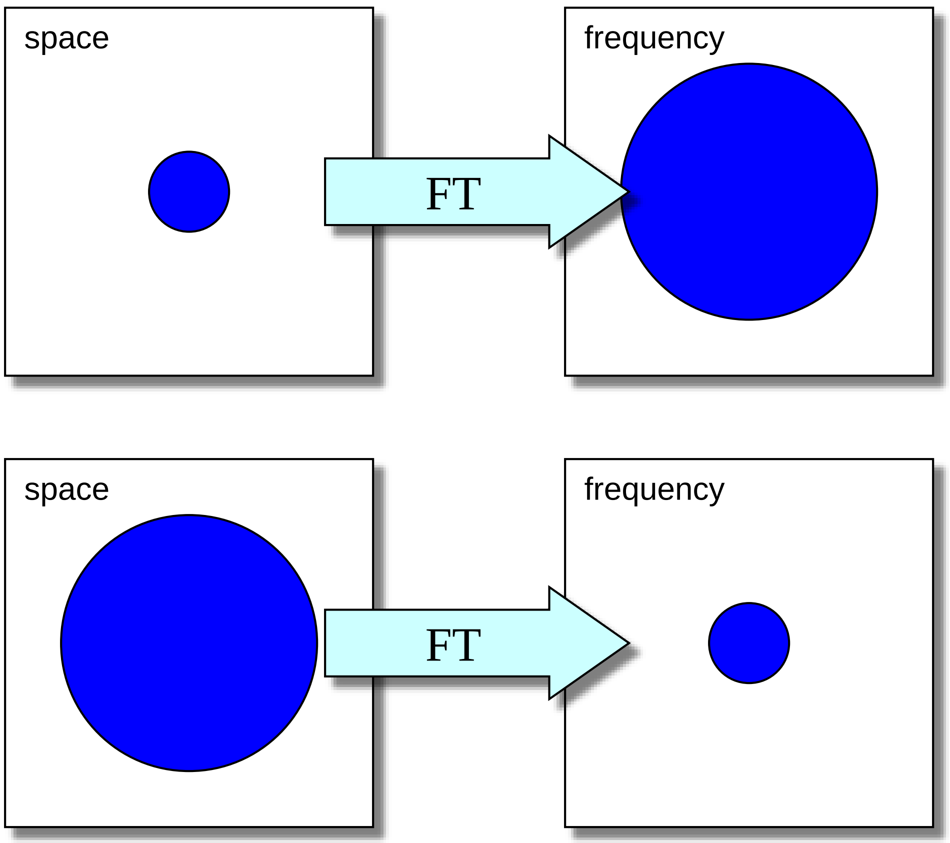
Ideal Filters Do Not Produce Ideal Results



Blurring the image above w/ an ideal lowpass filter…

…distorts the results with ringing or ghosting.

The Uncertainty Relation

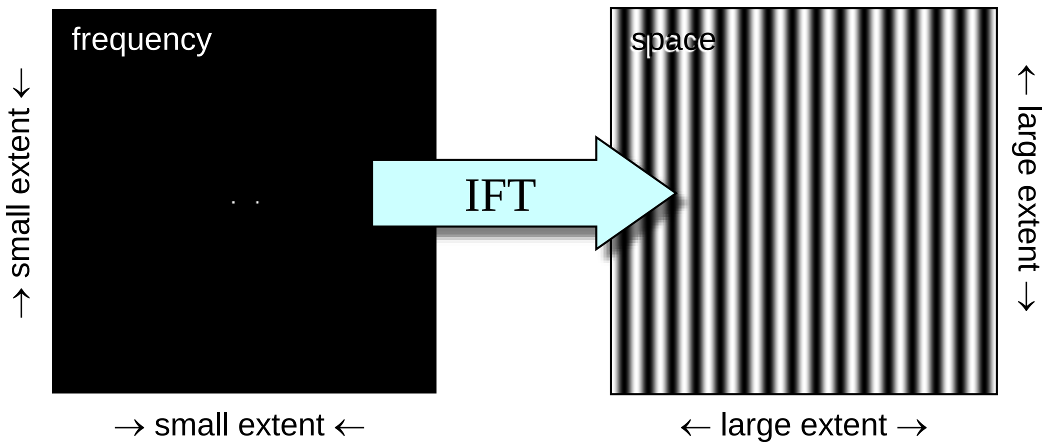


If $\Delta x \, \Delta y$ is the extent of the object in space and if $\Delta u \, \Delta v$ is its extent in frequency then,

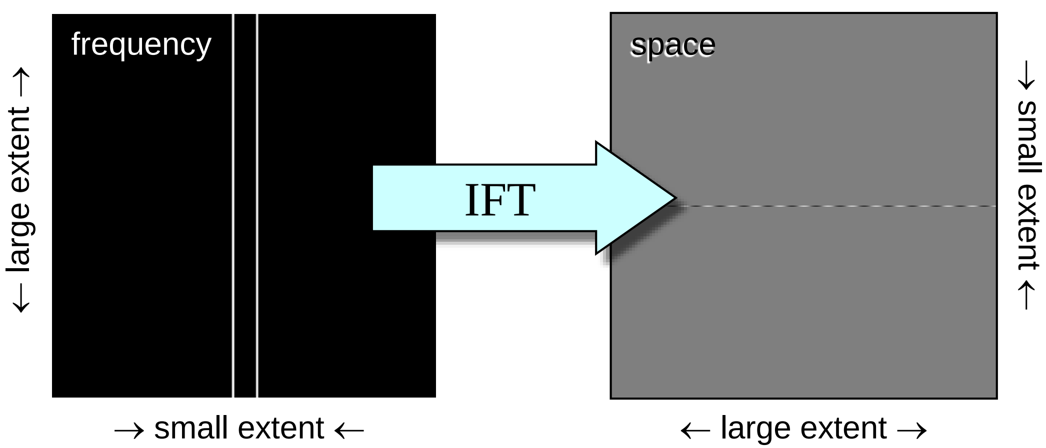
$$\Delta x \, \Delta y \cdot \Delta u \, \Delta v \geq \frac{1}{16\pi^2}$$

A small object in space has a large frequency extent and vice-versa.

The Uncertainty Relation

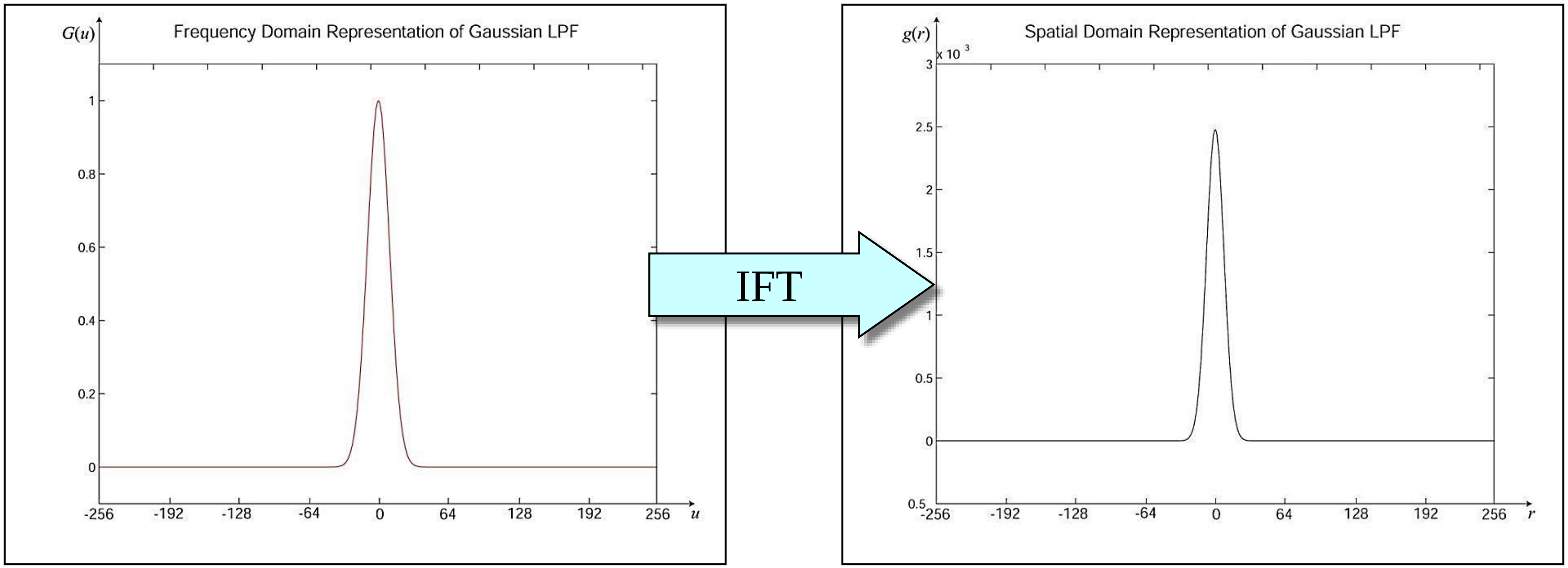


Recall: a symmetric pair of impulses in the frequency domain becomes a sinusoid in the spatial domain.



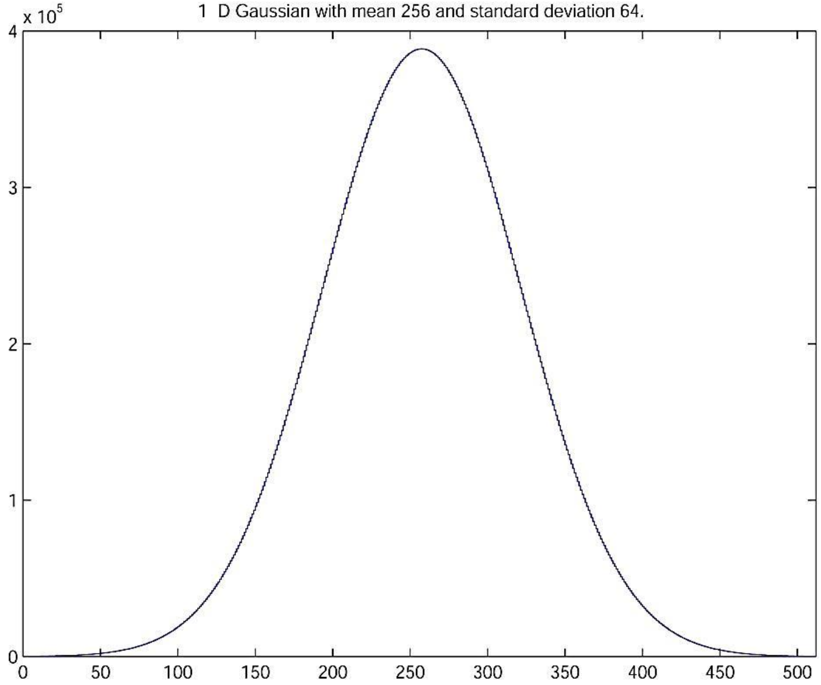
A symmetric pair of lines in the frequency domain becomes a sinusoidal line in the spatial domain.

Optimal Filter: The Gaussian



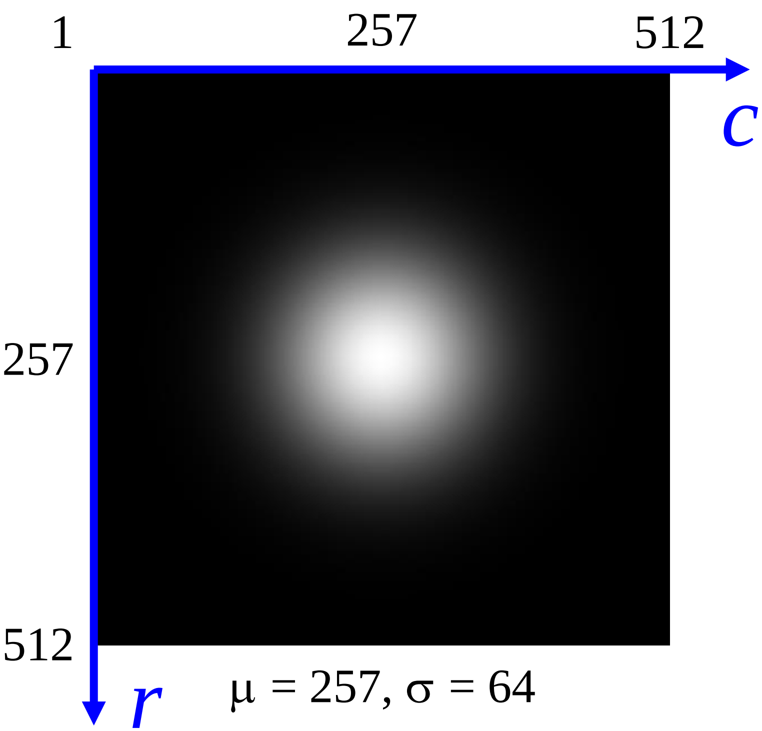
The Gaussian filter optimizes the uncertainty relation. It provides the sharpest cutoff possible without ringing.

One-Dimensional Gaussian



$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Two-Dimensional Gaussian



$\mu = 257$, $\sigma = 64$

If μ and σ are different for r & c…

$$g(r,c) = g(r)g(c)$$

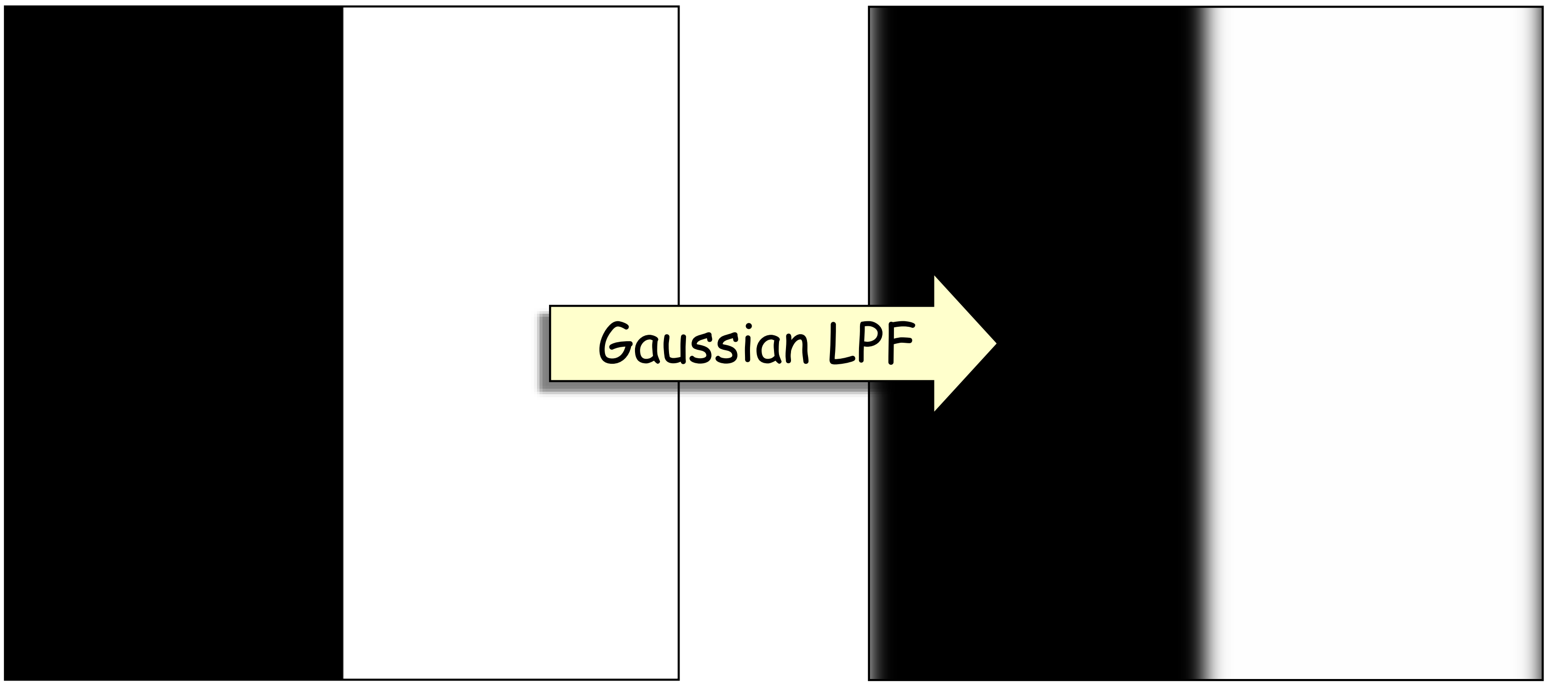
$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{(r-\mu_r)^2}{2\sigma_r^2} - \frac{(y-\mu_c)^2}{2\sigma_c^2}}$$

$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{\sigma_c^2 (x-\mu_r)^2 + \sigma_r^2 (y-\mu_c)^2}{2\sigma_r^2 \sigma_c^2}}$$

…or if μ and σ are the same for r & c.

$$g(r,c) = \frac{1}{\sigma^2 2\pi} e^{-\frac{(r-\mu)^2 + (c-\mu)^2}{2\sigma^2}}$$

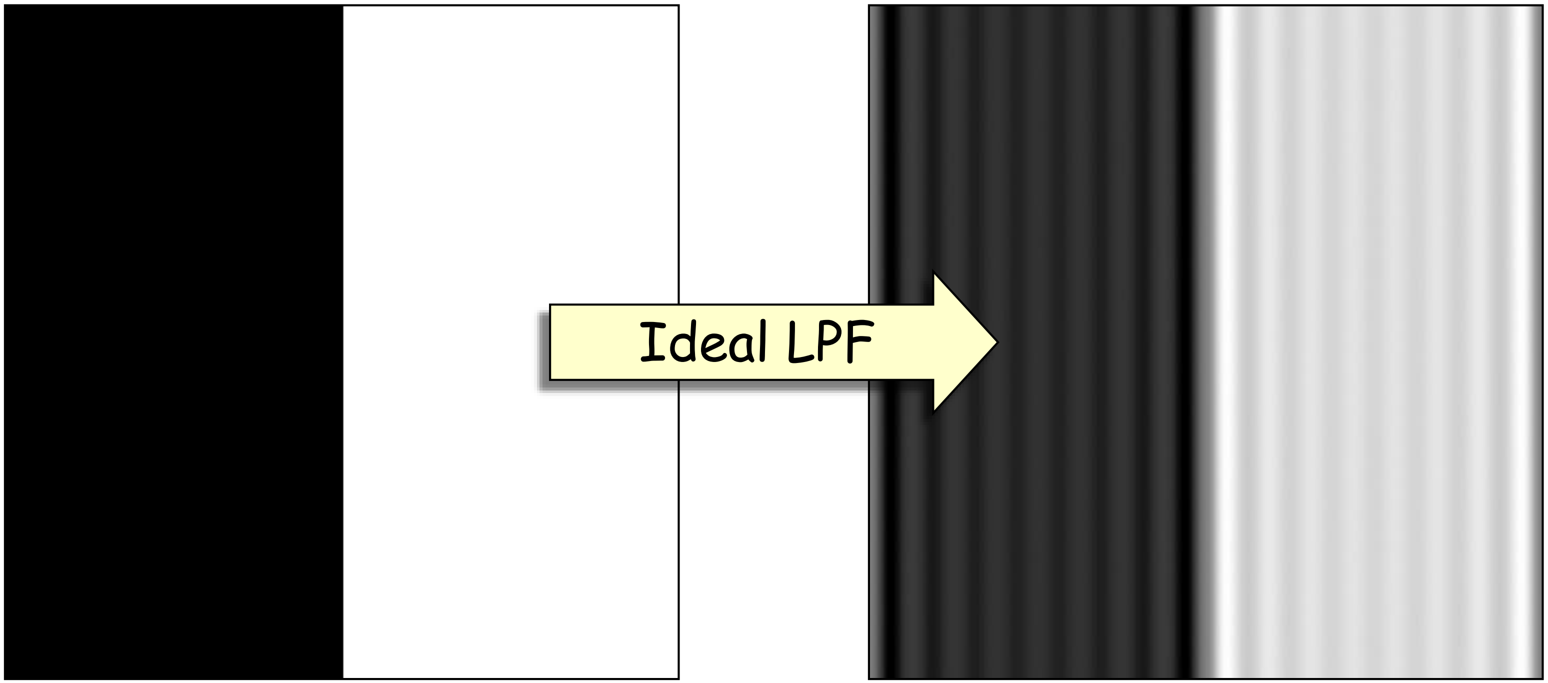
Optimal Filter: The Gaussian



With a gaussian lowpass filter, the image above …

… is blurred without ringing or ghosting.

Compare with an "Ideal" LPF



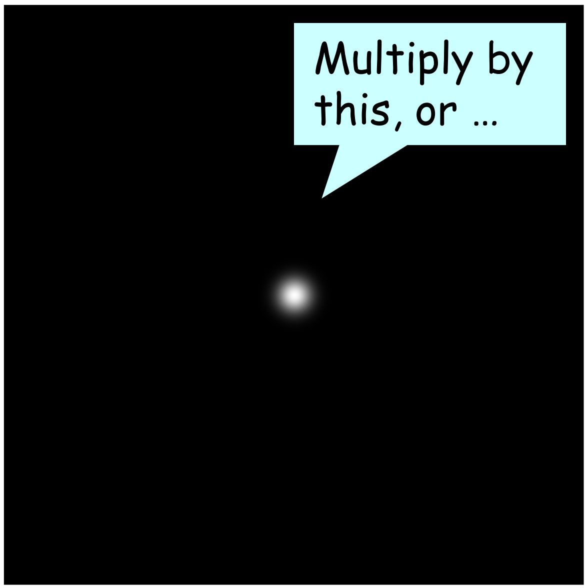
Blurring the image above w/ an ideal lowpass filter…

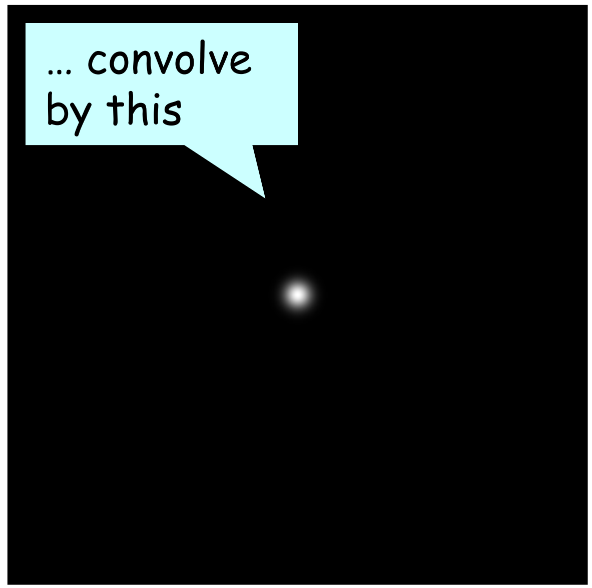
…distorts the results with ringing or ghosting.

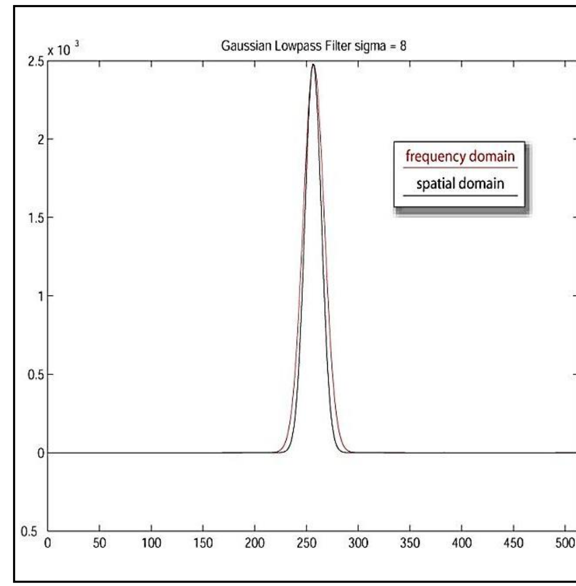
Gaussian Lowpass Filter

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8







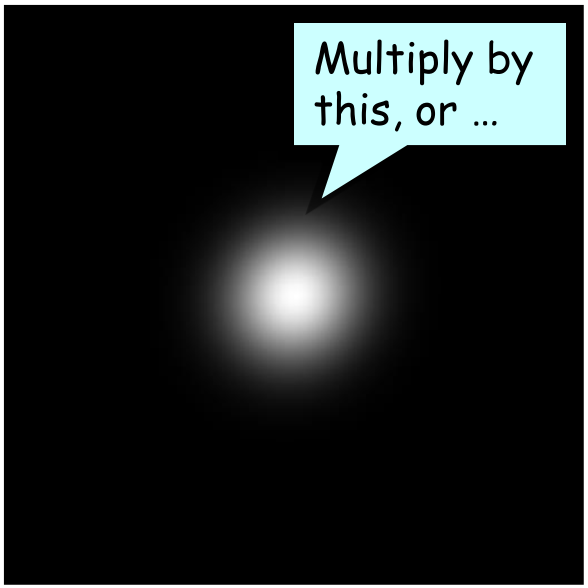
Fourier Domain Rep.

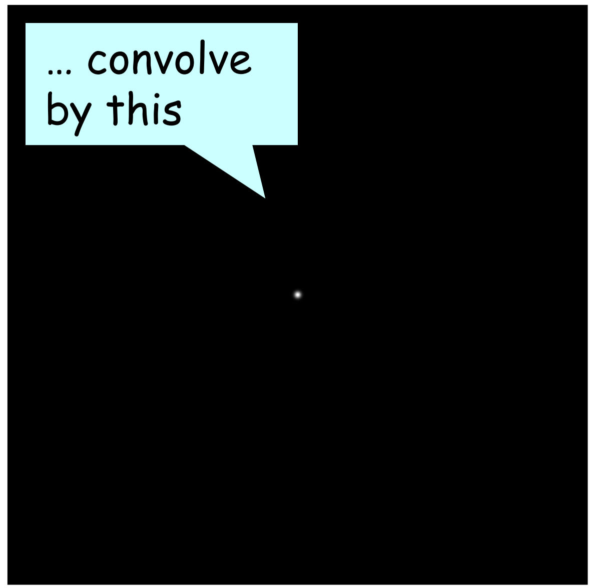
Spatial Representation

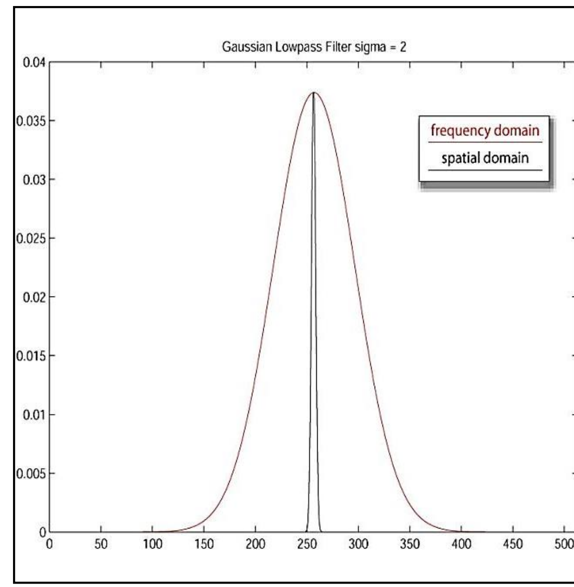
Central Profile

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 2







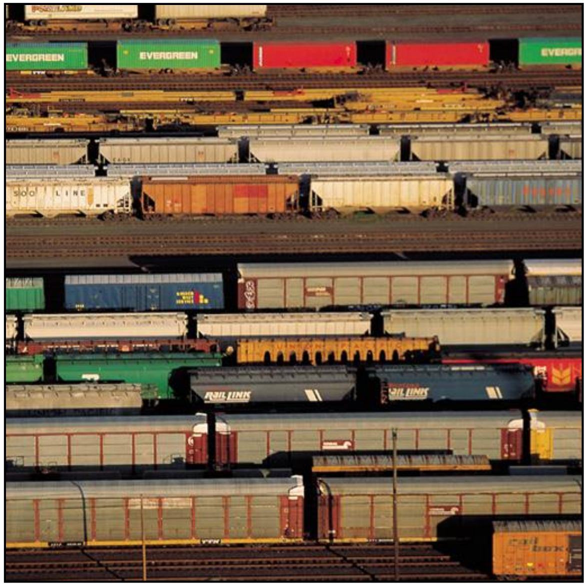
Fourier Domain Rep.

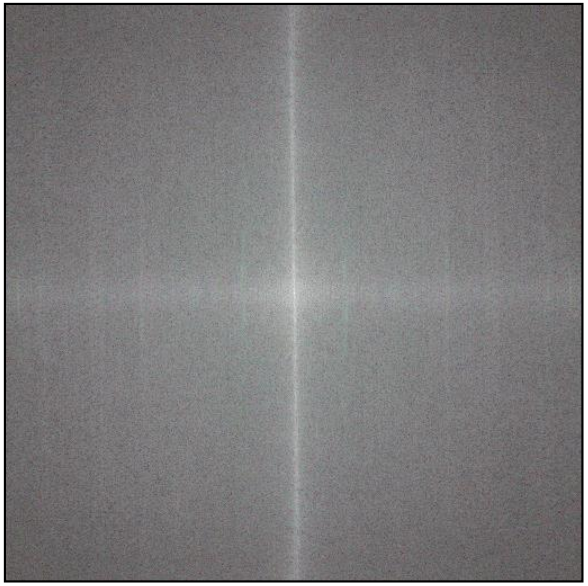
Spatial Representation

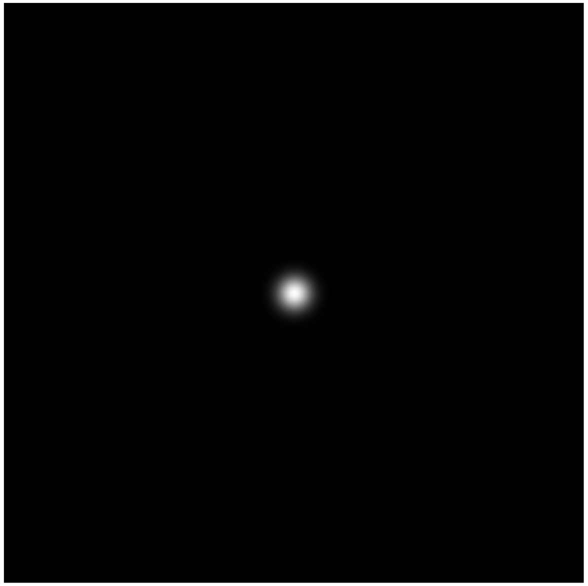
Central Profile

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8







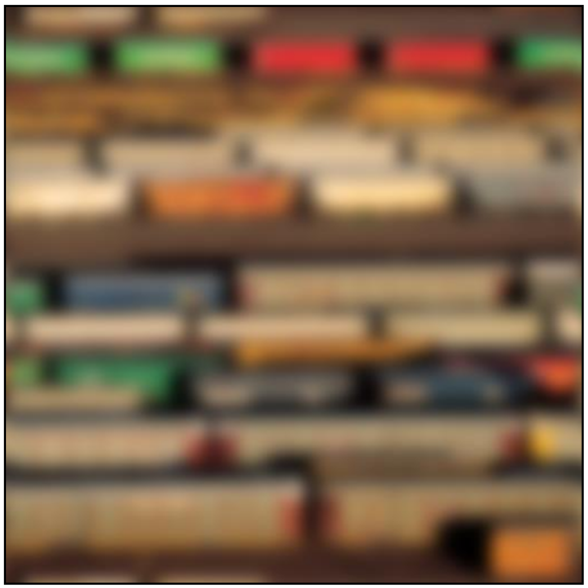
Original Image

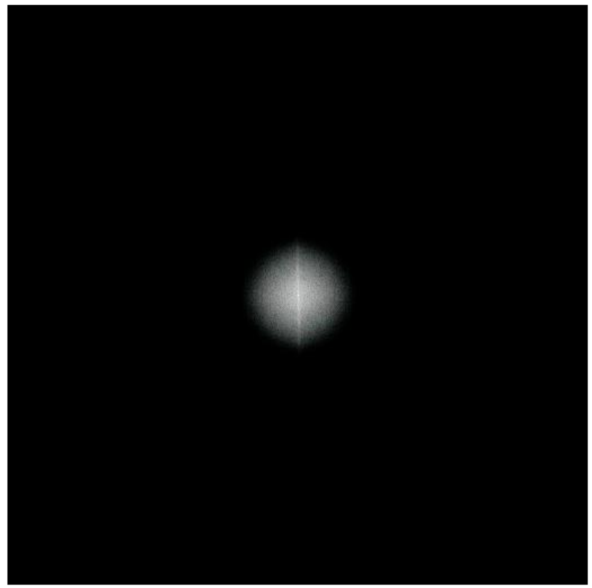
Power Spectrum

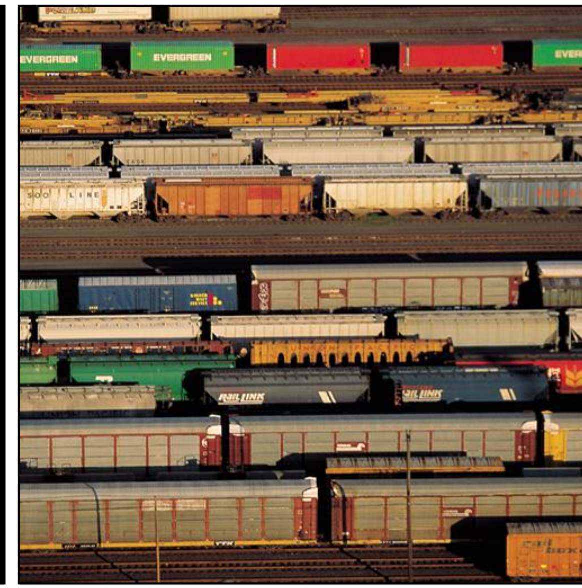
Gaussian LPF in FD

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8







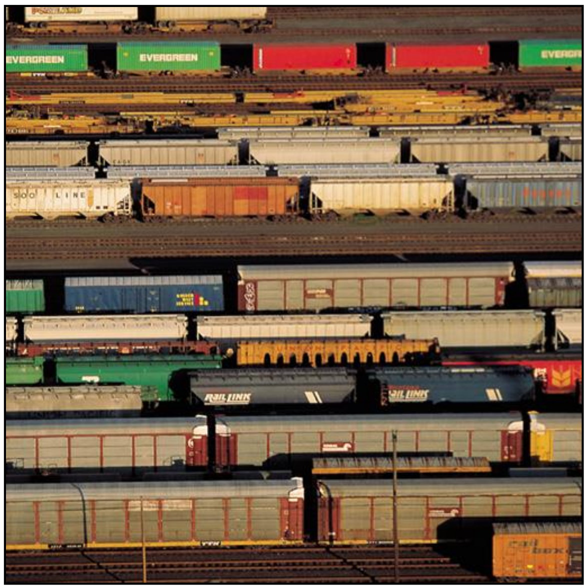
Filtered Image

Filtered Power Spectrum

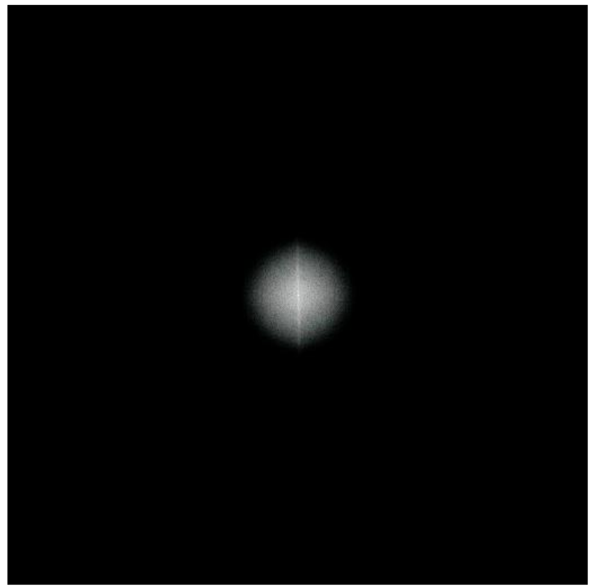
Original Image

Gaussian Lowpass Filter

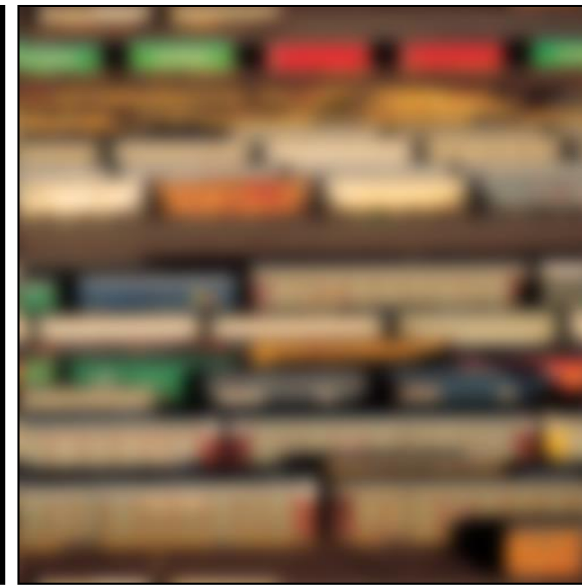
Image size: 512x512
SD filter sigma = 8



Original Image



Filtered Power Spectrum



Filtered Image

Resolution Sequence

Original Image

$\sigma_0 = 0$

Resolution Sequence

Gaussian LPF

$\sigma_1 = 1$

Resolution Sequence

Gaussian LPF

$\sigma_2 = 2$

Resolution Sequence

Gaussian LPF

$\sigma_3 = 4$

Resolution Sequence

Gaussian LPF

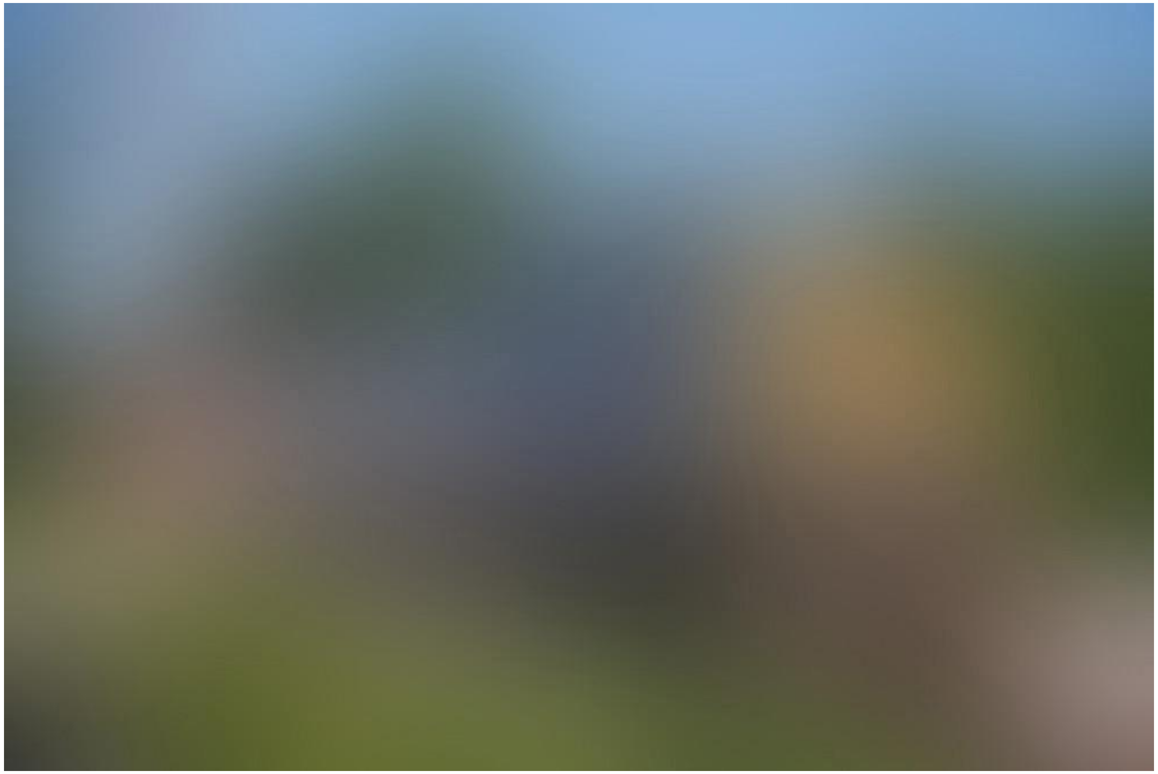
$\sigma_4 = 8$

Resolution Sequence

Gaussian LPF

$\sigma_5 = 16$

Resolution Sequence



Gaussian LPF

$\sigma_6 = 32$

Resolution Sequence

Gaussian LPF

$\sigma_7 = 64$

Resolution Sequence

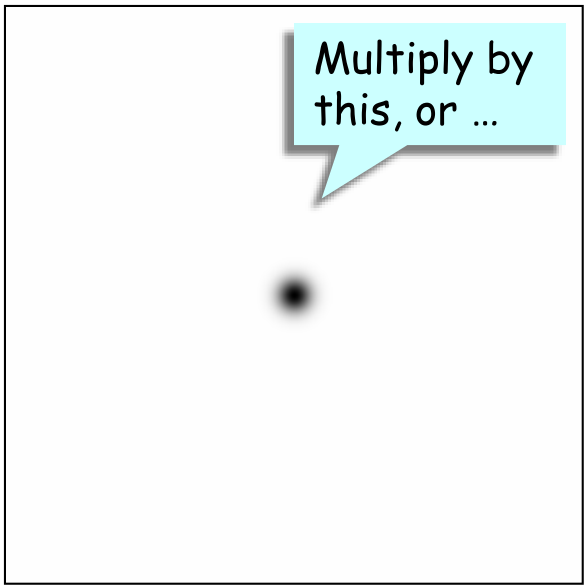
Gaussian LPF

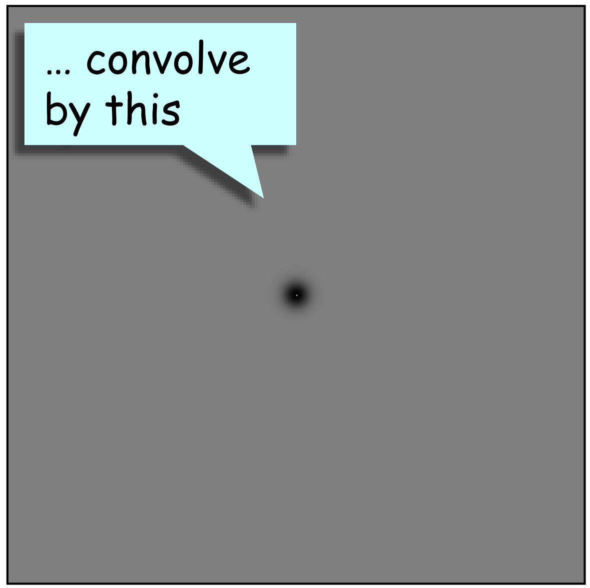
$\sigma_8 = 128$

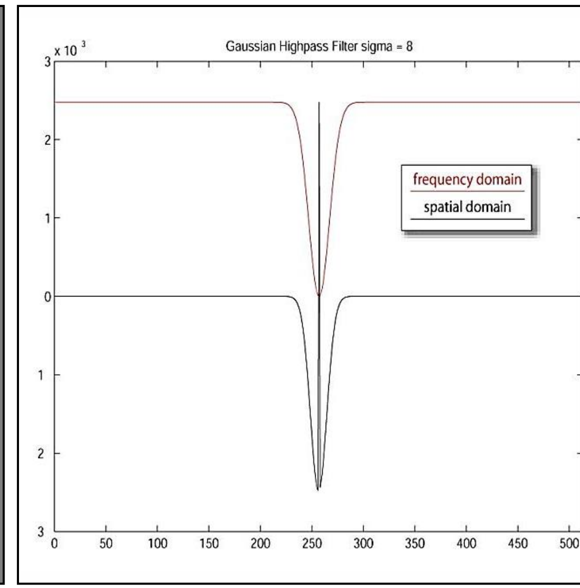
Gaussian Highpass Filter

Gaussian Highpass Filter

Image size: 512x512
FD notch sigma = 8







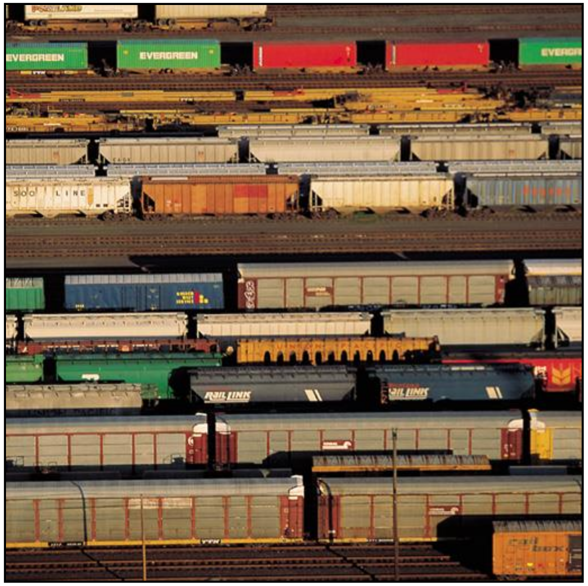
Fourier Domain Rep.

Spatial Representation

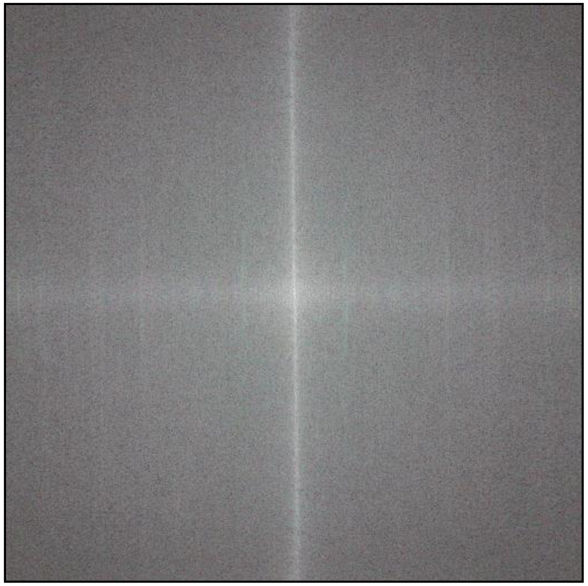
Central Profile

Gaussian Highpass Filter

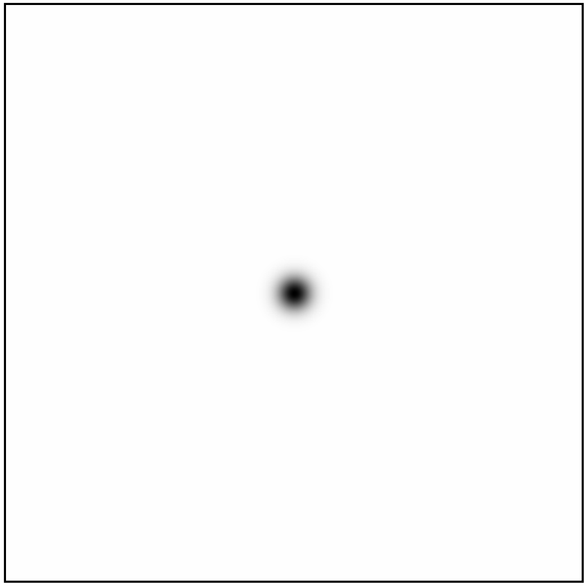
Image size: 512x512
FD notch sigma = 8



Original Image



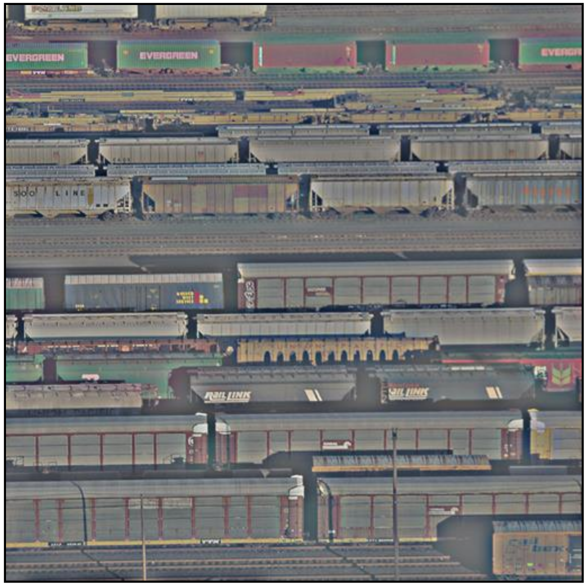
Power Spectrum



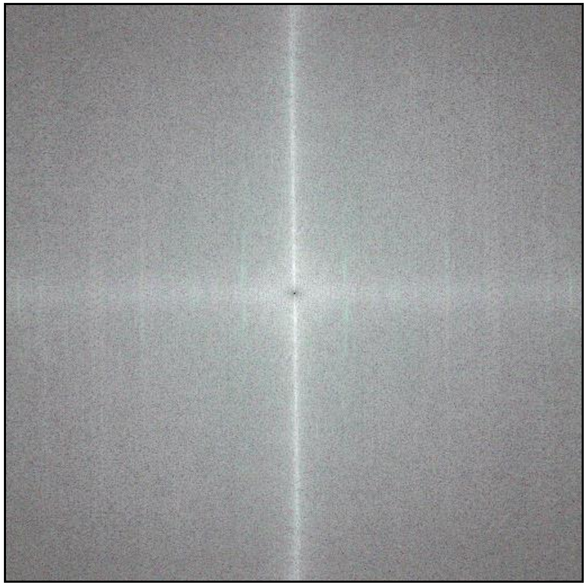
Gaussian HPF in FD

Gaussian Highpass Filter

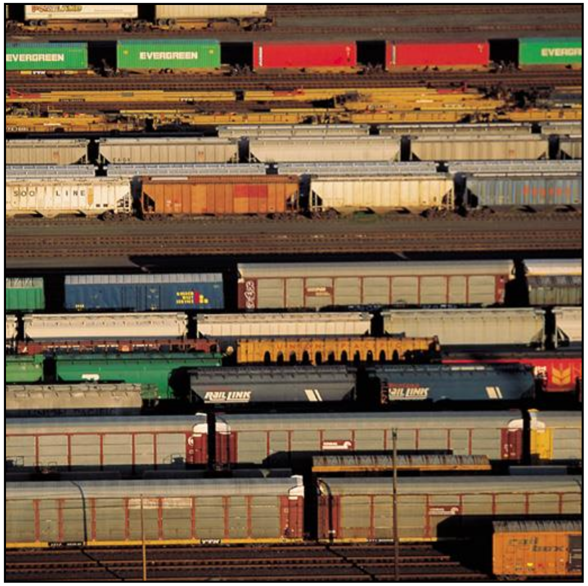
Image size: 512x512
FD notch sigma = 8



Filtered Image*



Filtered Power Spectrum

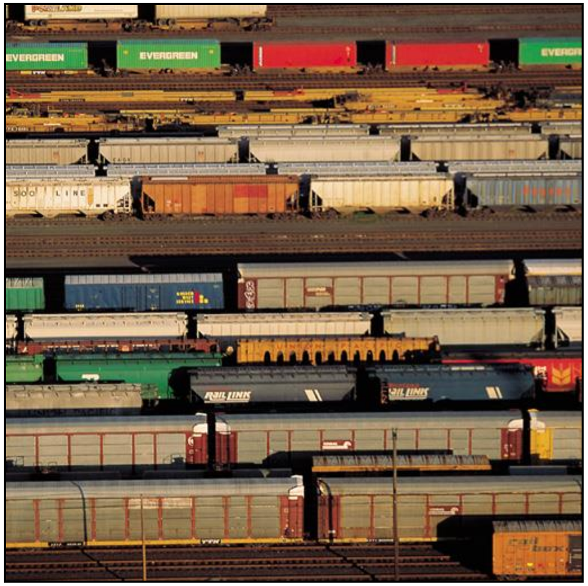


Original Image

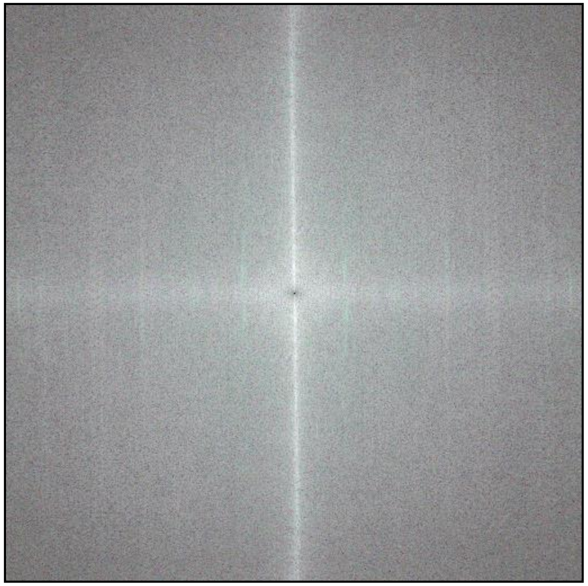
*signed image:
0 mapped to 128

Gaussian Highpass Filter

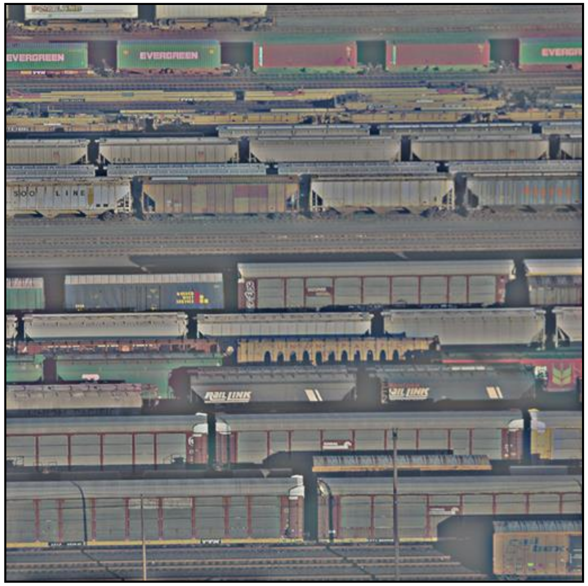
Image size: 512x512
FD notch sigma = 8



Original Image



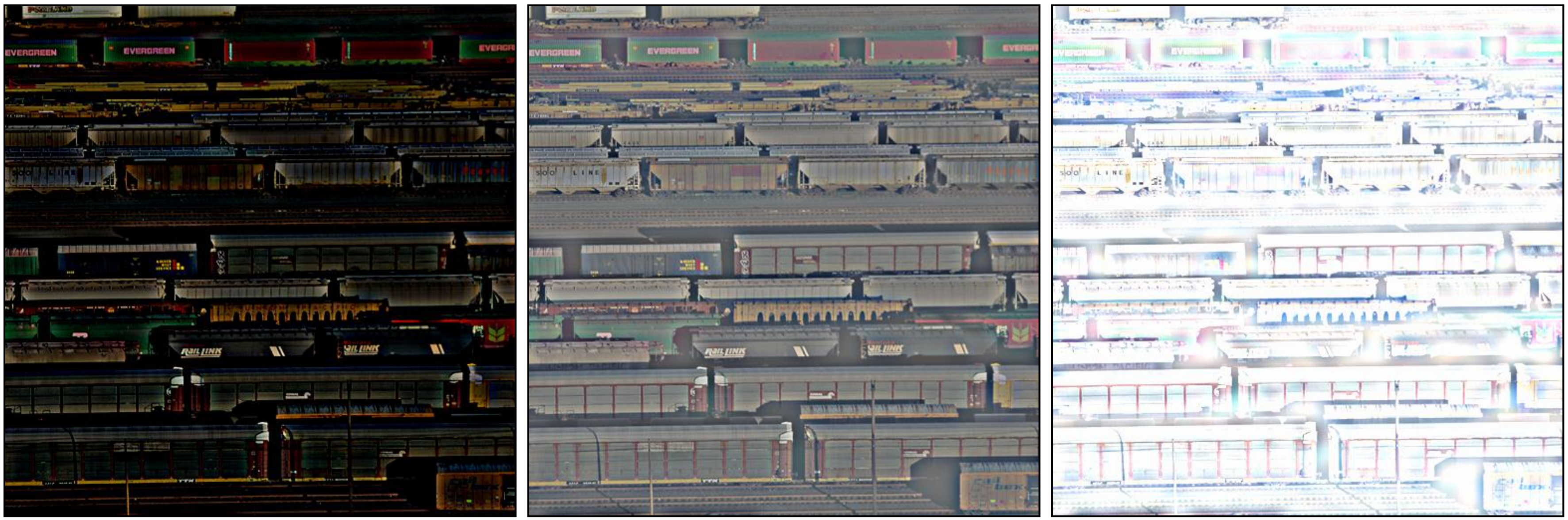
Filtered Power Spectrum



Filtered Image*

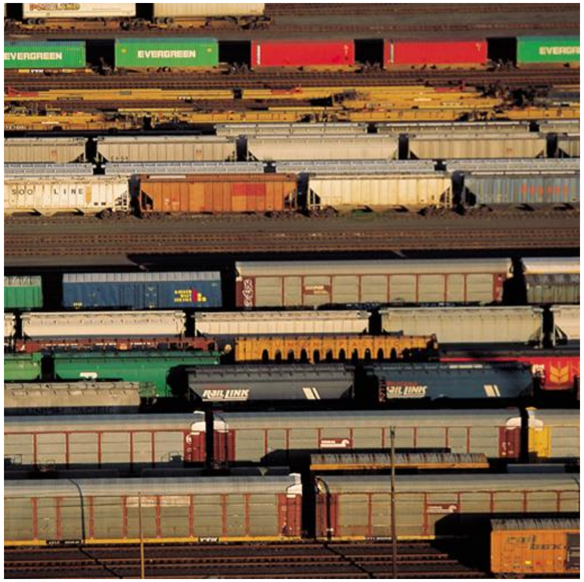
Gaussian Highpass Filter

Image size: 512x512
FD notch sigma = 8

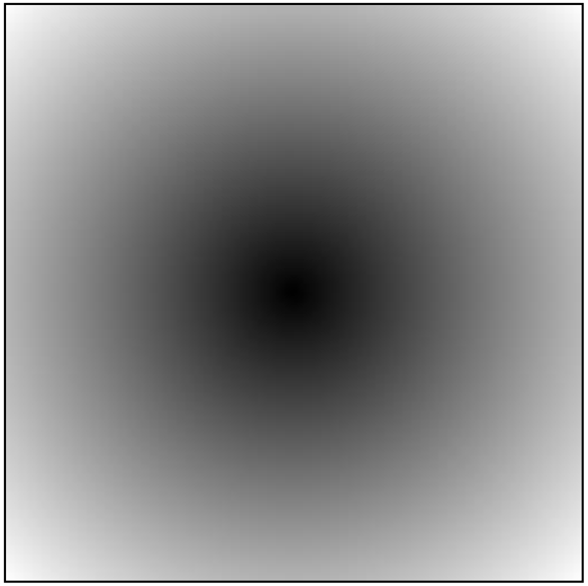


Positive Pixels | Filtered Image* | Negative Pixels

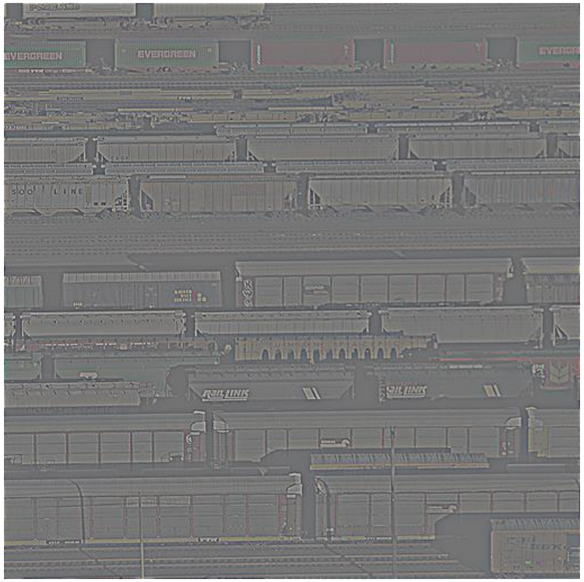
Another Gaussian Highpass Filter



Original Image



Filter Power Spectrum



Filtered Image*

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_9 = 256$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_9\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_8 = 128$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_8\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_7 = 64$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_7\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_6 = 32$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_6\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_5 = 16$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_5\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_4 = 8$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_4\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_3 = 4$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_3\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_2 = 2$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_2\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_1 = 1$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_1\right) \right].$$

Highpass Sequence

Original Image

$\sigma_0 = 0$

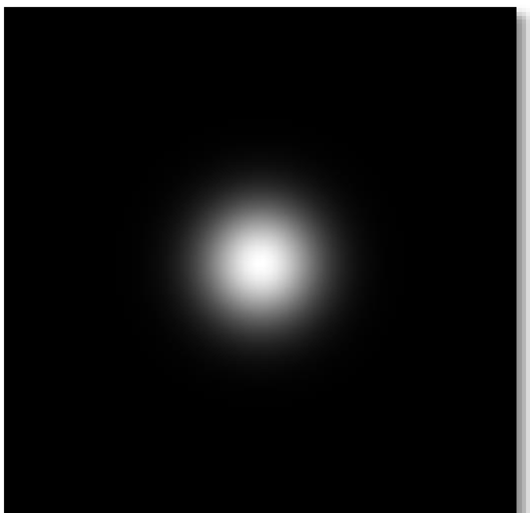
Gaussian Bandpass Filter

Gaussian Bandpass Filter

A bandpass filter is created by

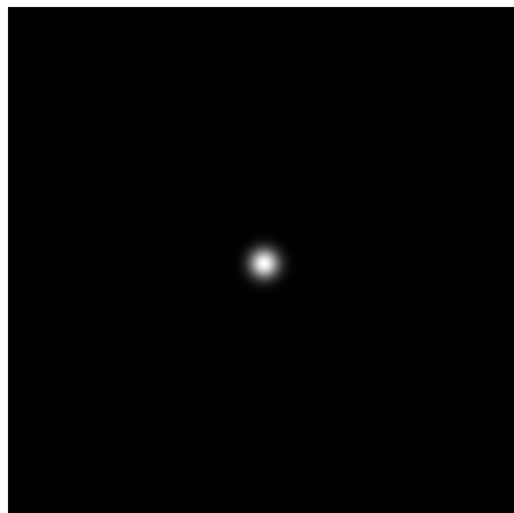
(1) subtracting a FD radius ρ_2 lowpass filtered image from a FD radius ρ_1 lowpass filtered image, where $\rho_2 < \rho_1$, or

(2) convolving the image with a matrix that is the difference of the two lowpass matrixs.



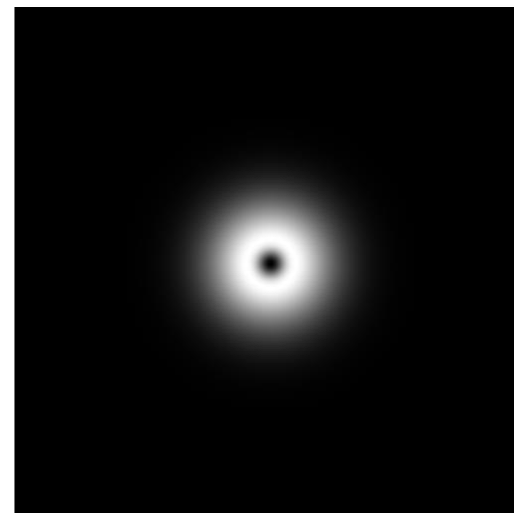
FD LP matrix with radius σ1

-



FD LP matrix with radius σ2

=



FD BP matrix σ1 - σ2

*signed image: 0 mapped to 128

Gaussian Bandpass Filter

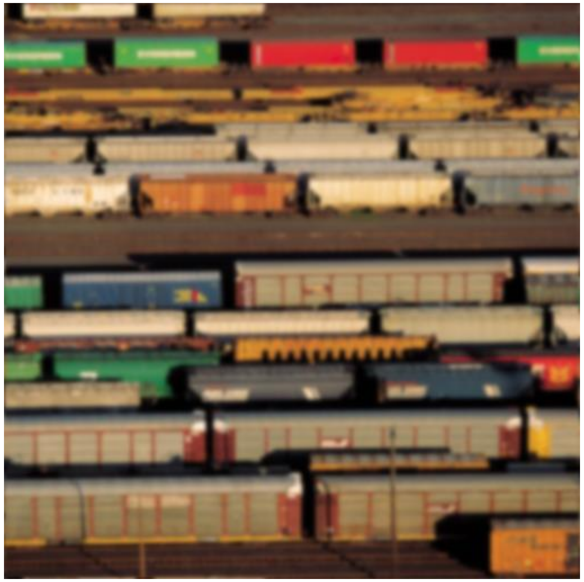


image LPF radius ρ1

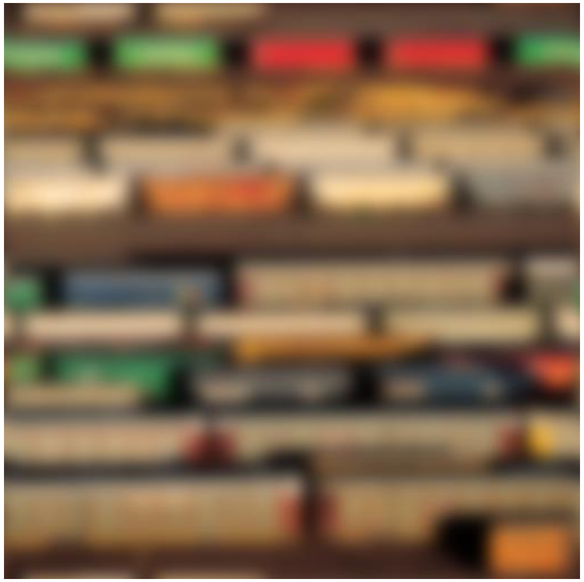


image LPF radius ρ2

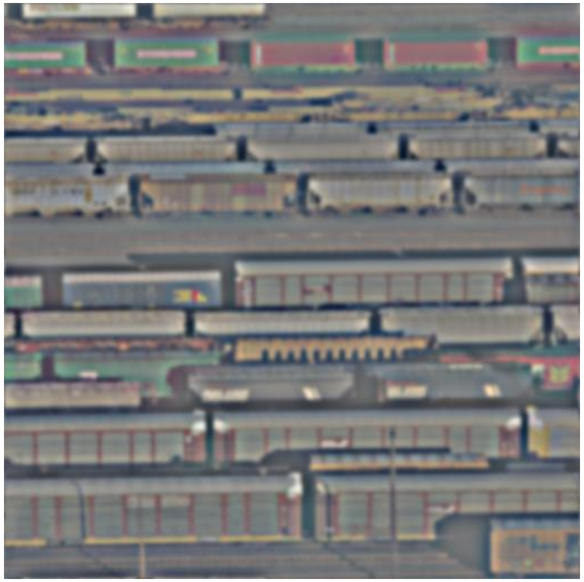
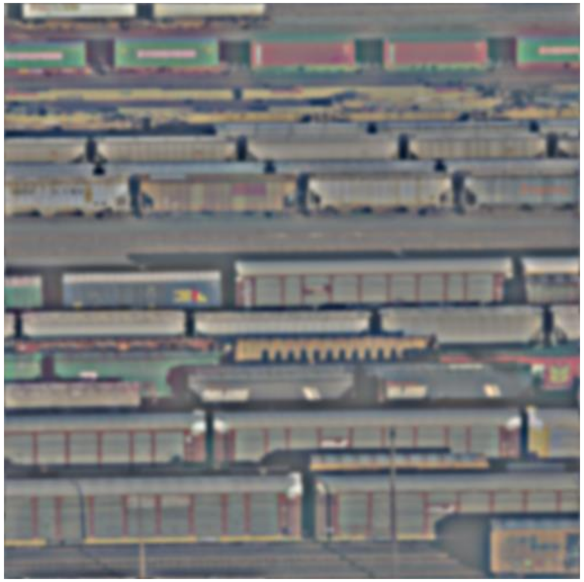
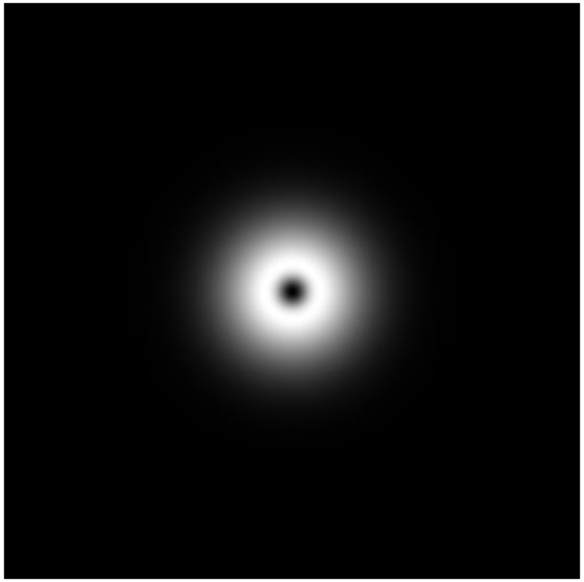


image BPF radii ρ1, ρ2*

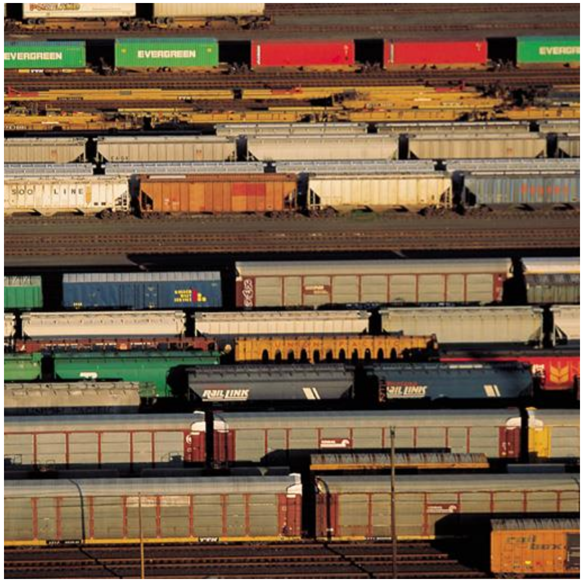
Gaussian Bandpass Filter



filtered image*

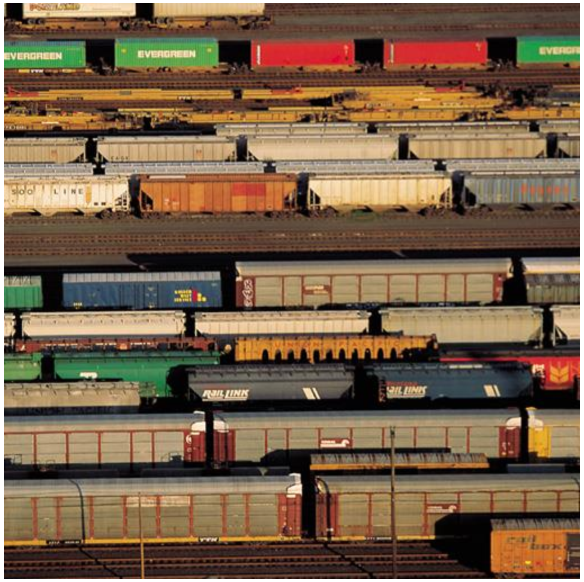
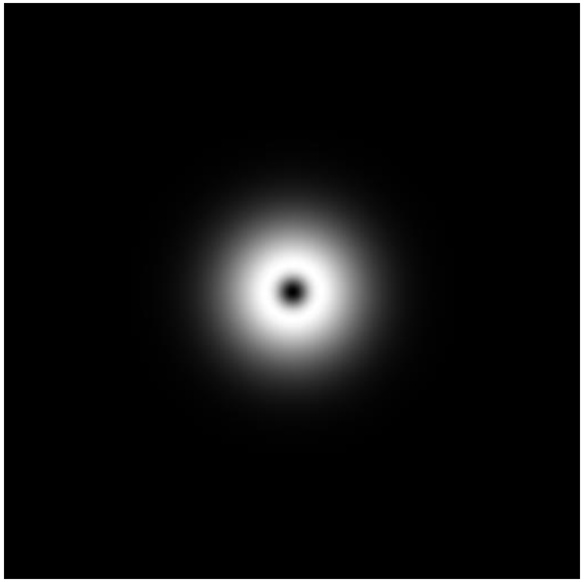
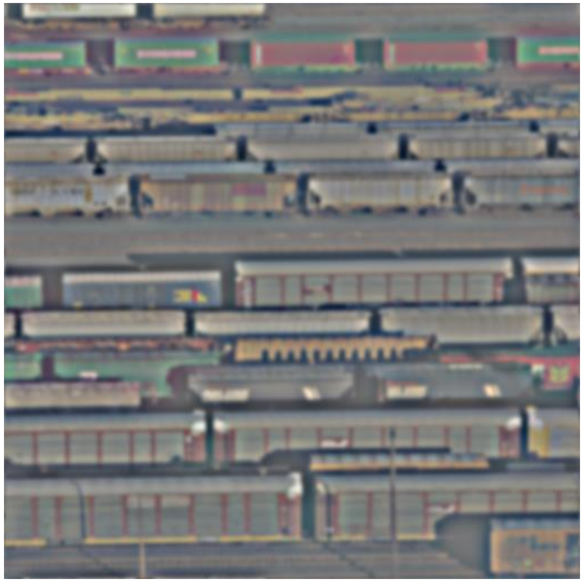


filter power spectrum



original image

Ideal Bandpass Filter

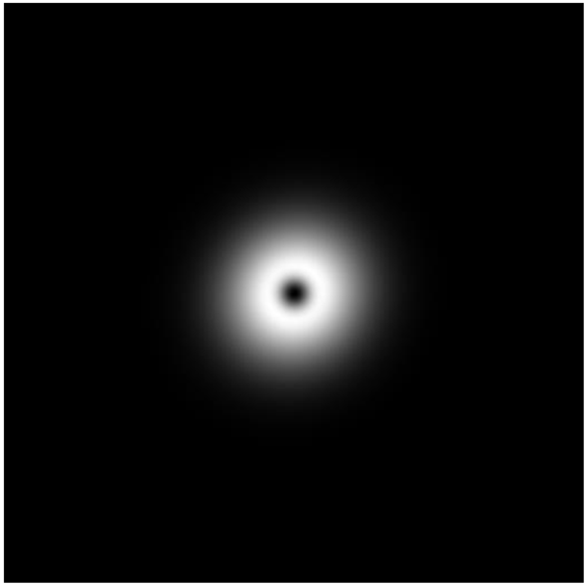
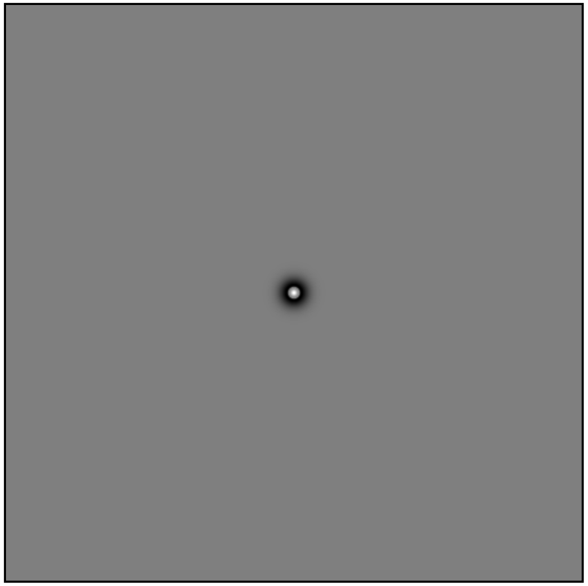
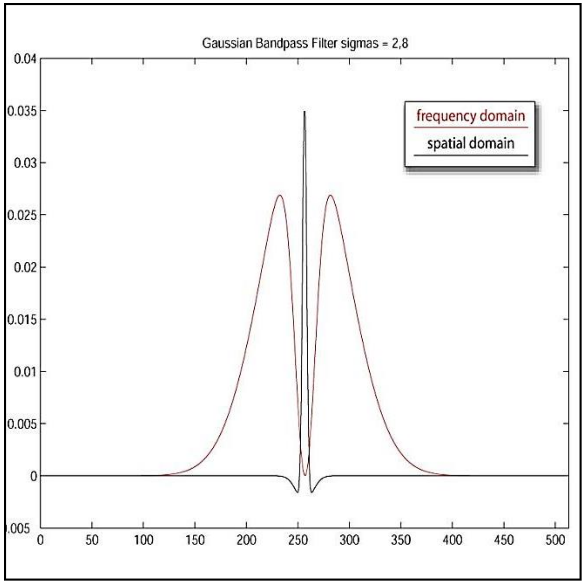
original image

filter power spectrum

filtered image*

Gaussian Bandpass Filter

Image size: 512x512
sigma = 2 - sigma = 8

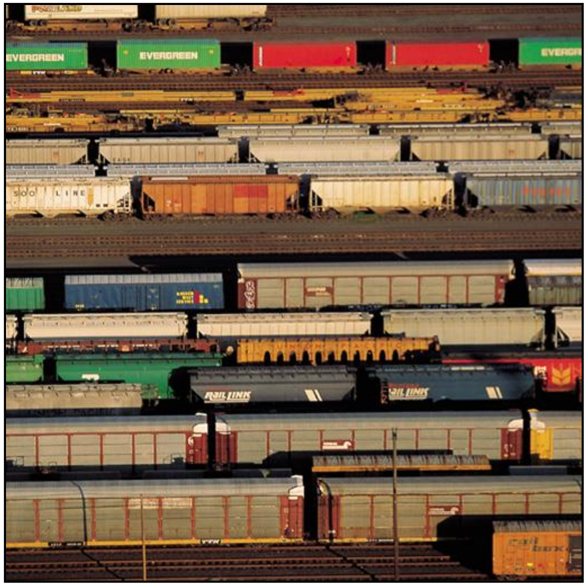
Fourier Domain Rep.

Spatial Representation

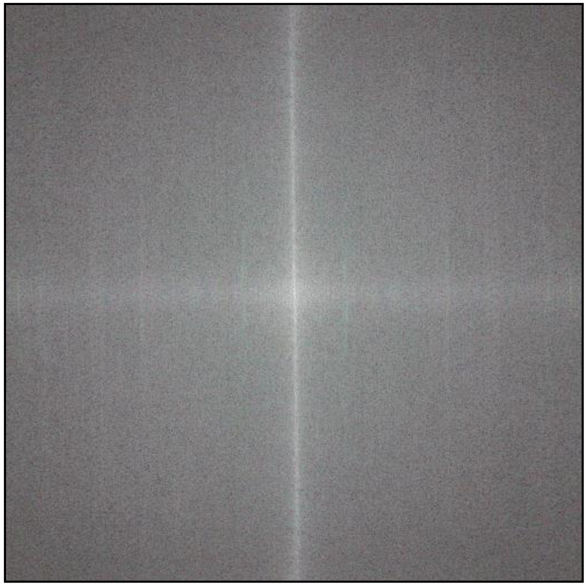
Central Profile

Gaussian Bandpass Filter

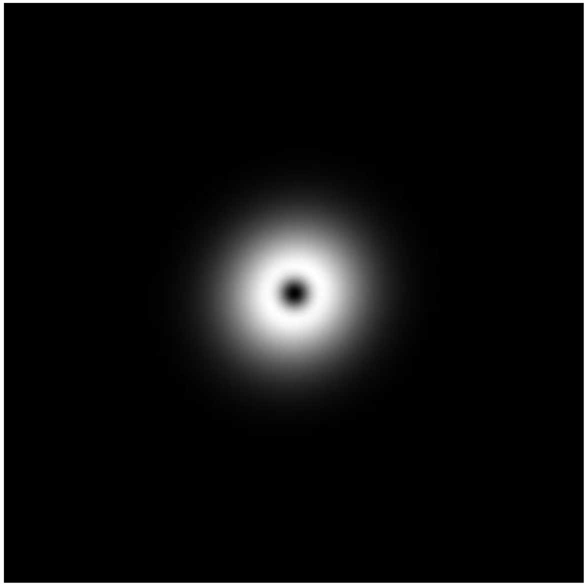
Image size: 512x512
sigma = 2 - sigma = 8



Original Image



Power Spectrum

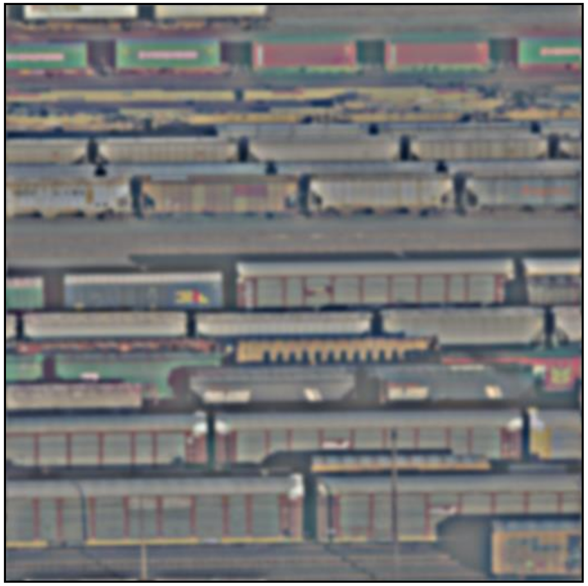


Gaussian BPF in FD

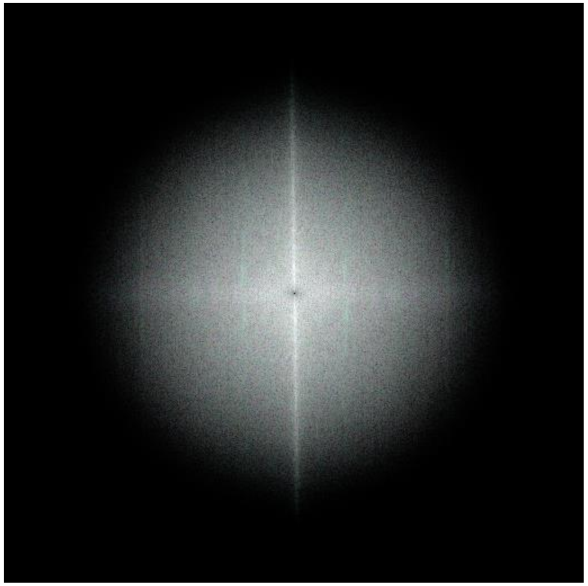
*signed image:
0 mapped to 128

Gaussian Bandpass Filter

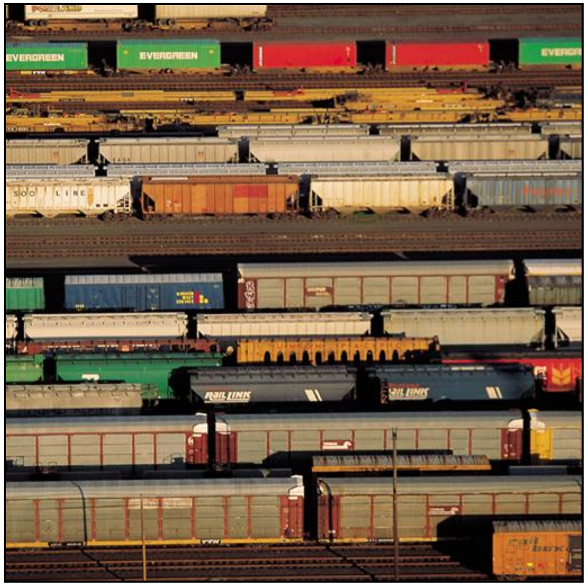
Image size: 512x512
sigma = 2 - sigma = 8



Filtered Image*



Filtered Power Spectrum

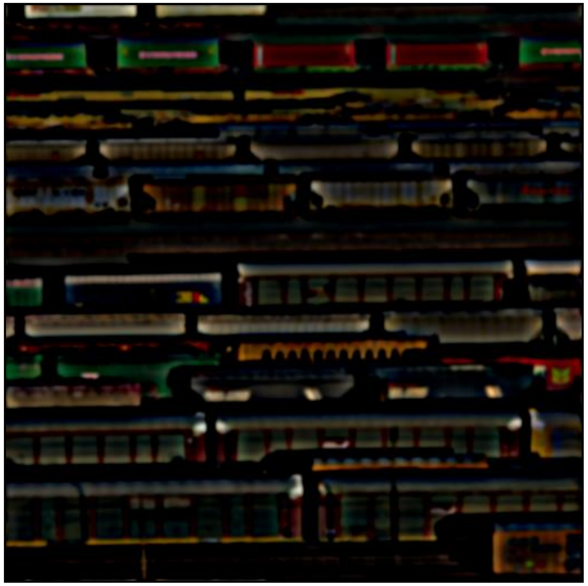


Original Image

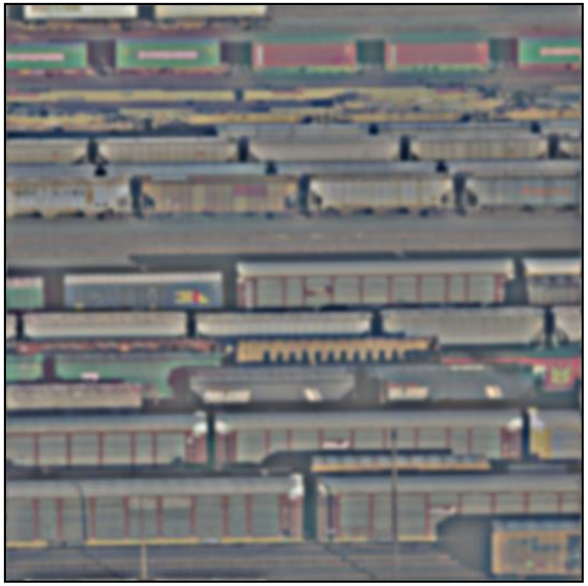
*signed image:
0 mapped to 128

Gaussian Bandpass Filter

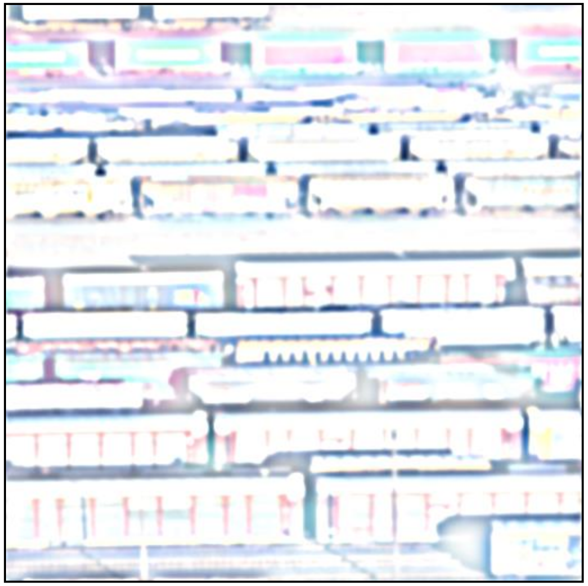
Image size: 512x512
sigma = 2 - sigma = 8



Positive Pixels

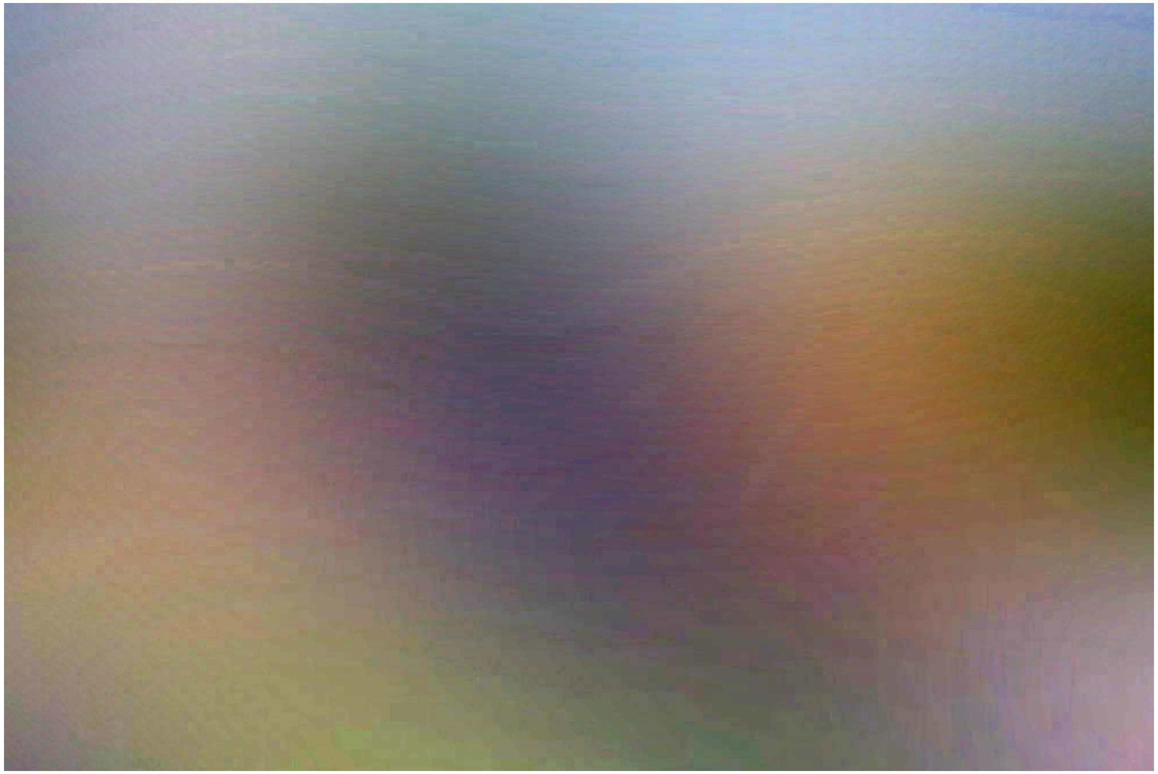


Filtered Image*



Negative Pixels

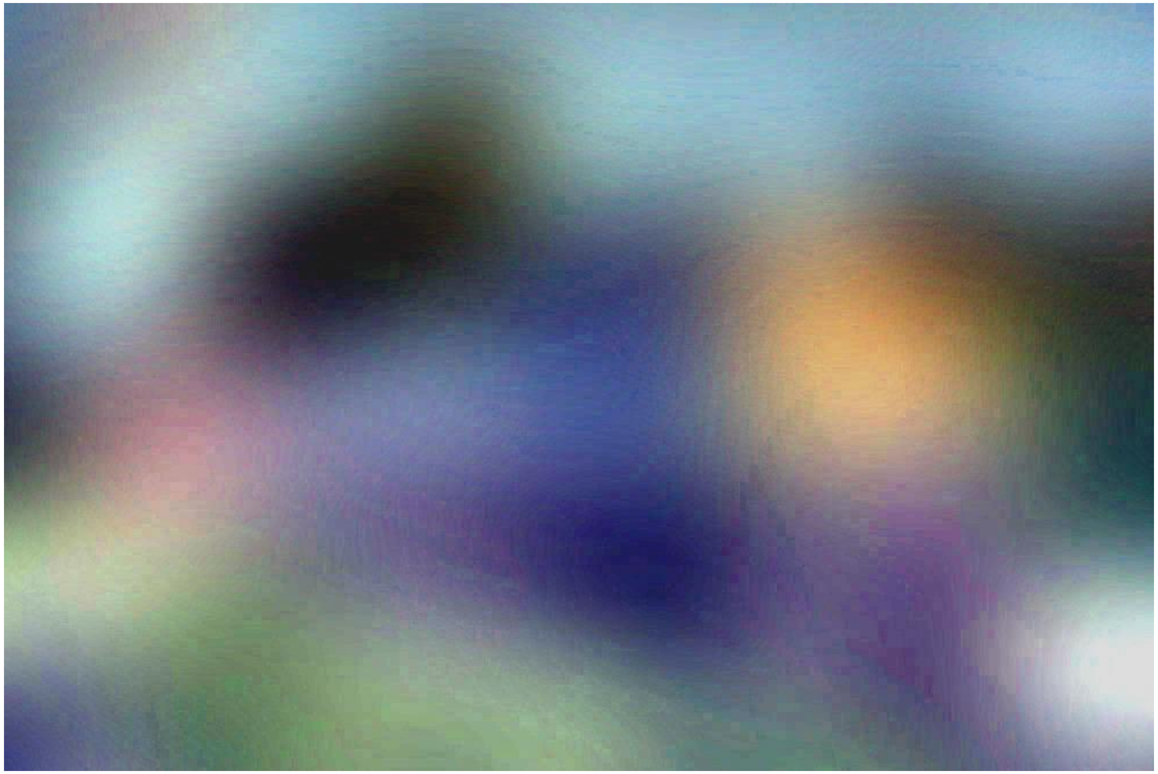
Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_8 = 128$ and $\sigma_9 = 256$.

$$\mathbf{J} = \mathbf{I} * \left[g(\sigma_8) - g(\sigma_9) \right] = \left[\mathbf{I} * g(\sigma_8) \right] - \left[\mathbf{I} * g(\sigma_9) \right].$$

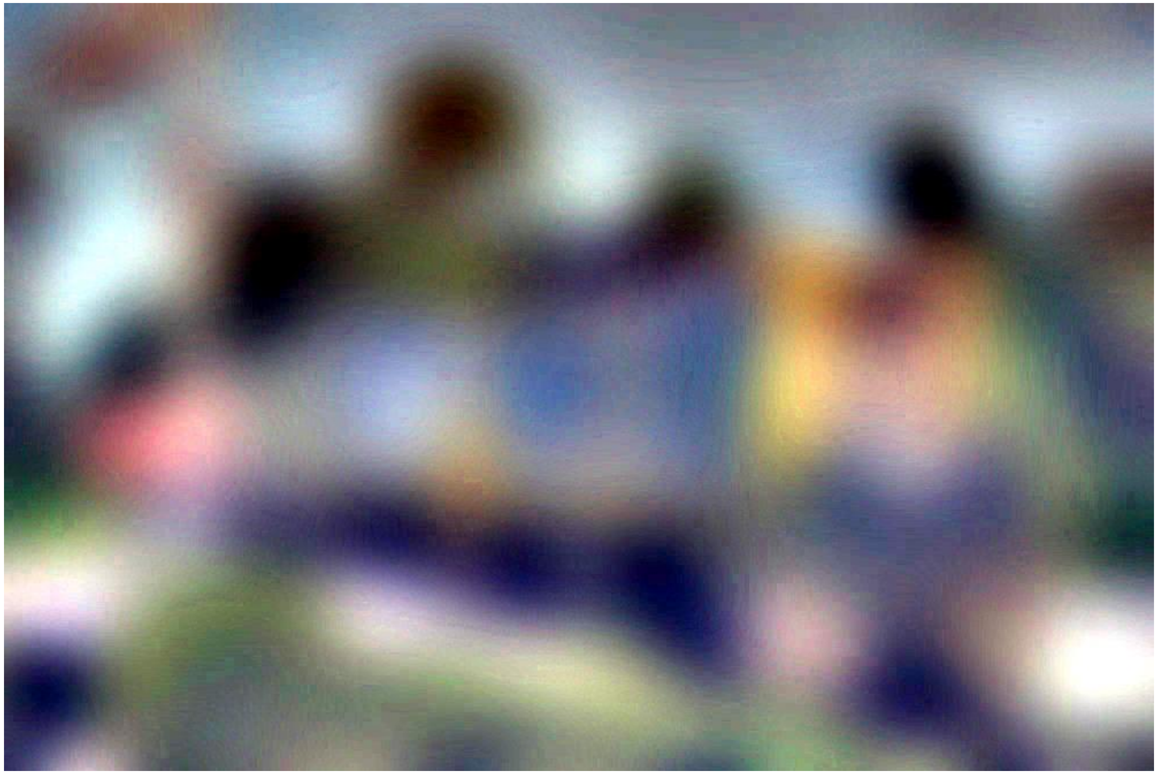
Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_7 = 64$ and $\sigma_8 = 128$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_6\right) - g\left(\sigma_7\right) \right] = \left[\mathbf{I} * g\left(\sigma_6\right) \right] - \left[\mathbf{I} * g\left(\sigma_7\right) \right].$$

Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_6 = 32$ and $\sigma_7 = 64$.

$$\mathbf{J} = \mathbf{I} * \left[g(\sigma_6) - g(\sigma_7) \right] = \left[\mathbf{I} * g(\sigma_6) \right] - \left[\mathbf{I} * g(\sigma_7) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_5 = 16$ and $\sigma_6 = 32$.

$$\mathbf{J} = \mathbf{I} * \left[g(\sigma_5) - g(\sigma_6) \right] = \left[\mathbf{I} * g(\sigma_5) \right] - \left[\mathbf{I} * g(\sigma_6) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_4 = 8$ and $\sigma_5 = 16$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_4\right) - g\left(\sigma_5\right) \right] = \left[\mathbf{I} * g\left(\sigma_4\right) \right] - \left[\mathbf{I} * g\left(\sigma_5\right) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_3 = 4$ and $\sigma_4 = 8$.

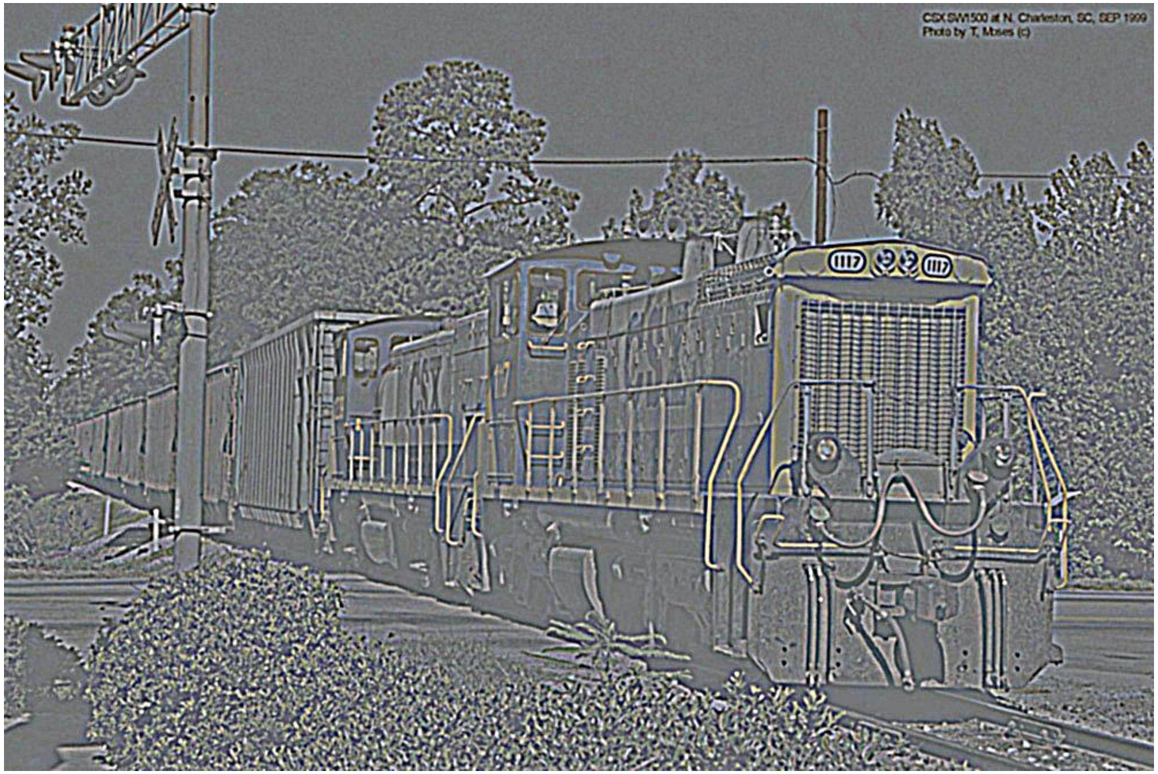
$$\mathbf{J} = \mathbf{I} * \left[g(\sigma_3) - g(\sigma_4) \right] = \left[\mathbf{I} * g(\sigma_3) \right] - \left[\mathbf{I} * g(\sigma_4) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_2 = 2$ and $\sigma_3 = 4$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_2\right) - g\left(\sigma_3\right) \right] = \left[\mathbf{I} * g\left(\sigma_2\right) \right] - \left[\mathbf{I} * g\left(\sigma_3\right) \right].$$

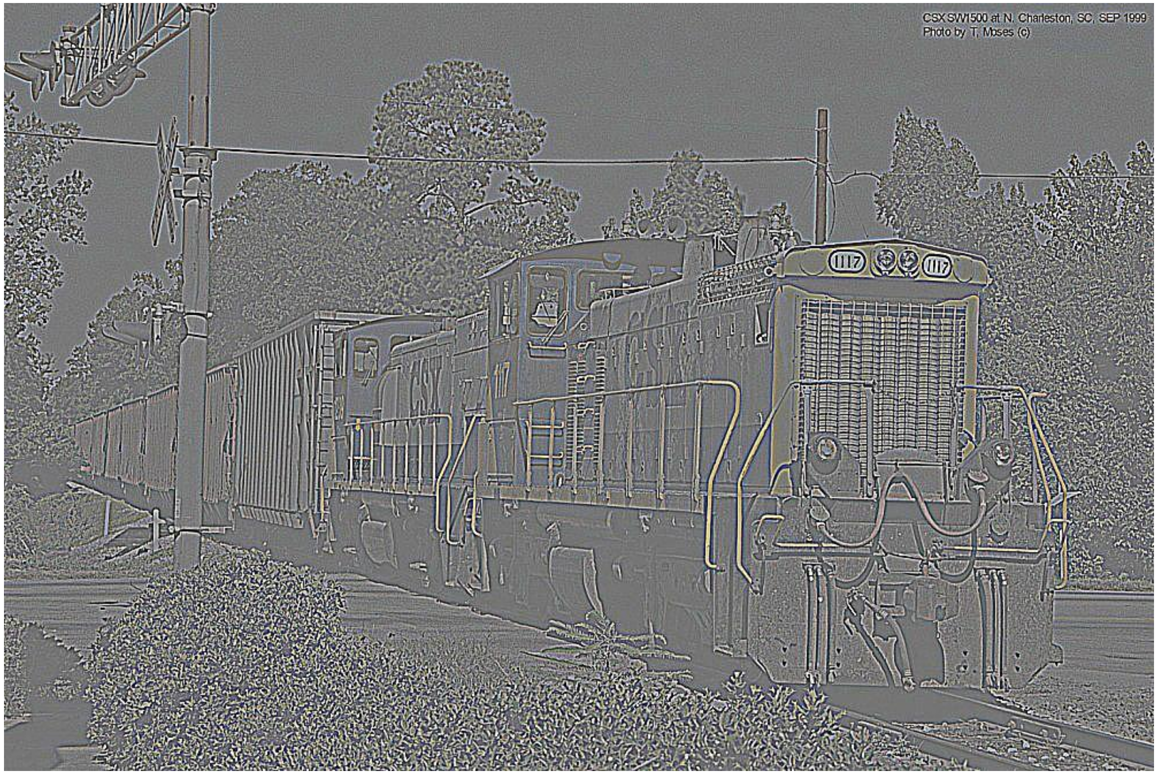
Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_1 = 1$ and $\sigma_2 = 2$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_1\right) - g\left(\sigma_2\right) \right] = \left[\mathbf{I} * g\left(\sigma_1\right) \right] - \left[\mathbf{I} * g\left(\sigma_2\right) \right].$$

Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_0 = 0$ and $\sigma_1 = 1$.

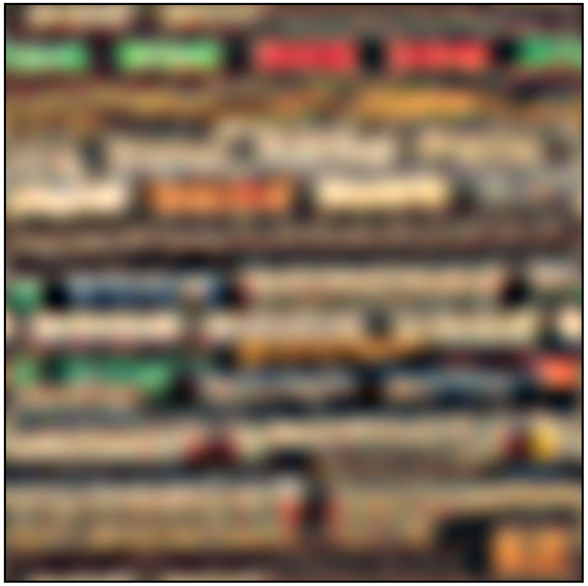
$$\mathbf{J} = \mathbf{I} * \left[\delta(0) - g\left(\sigma_1\right)\right] = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_1\right)\right].$$

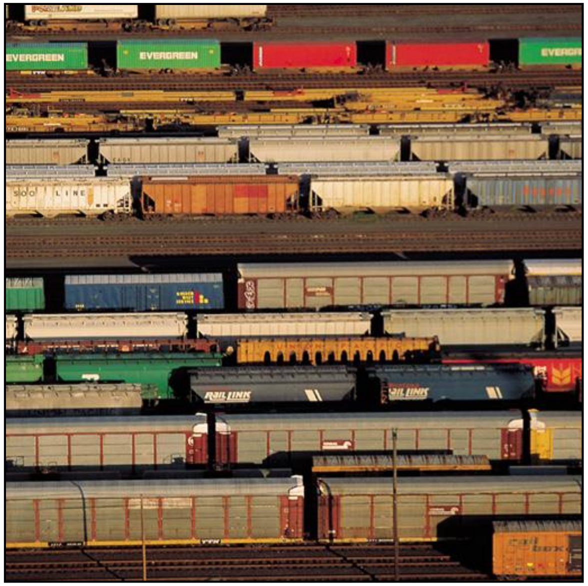
Bandpass Sequence

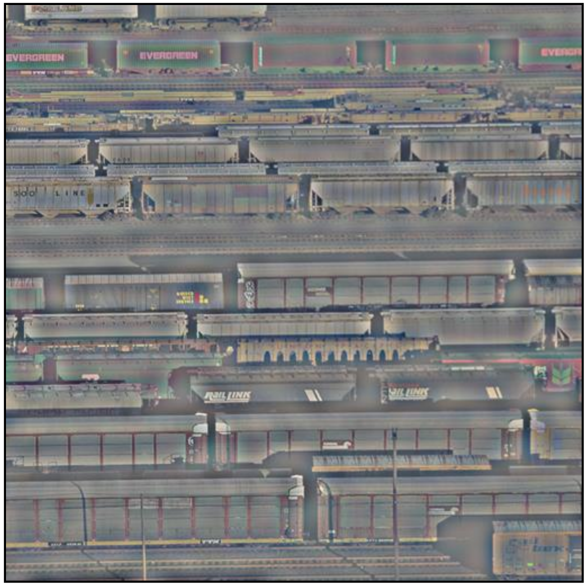
Original Image

Ideal vs. Gaussian Filters

Ideal Lowpass and Highpass Filters







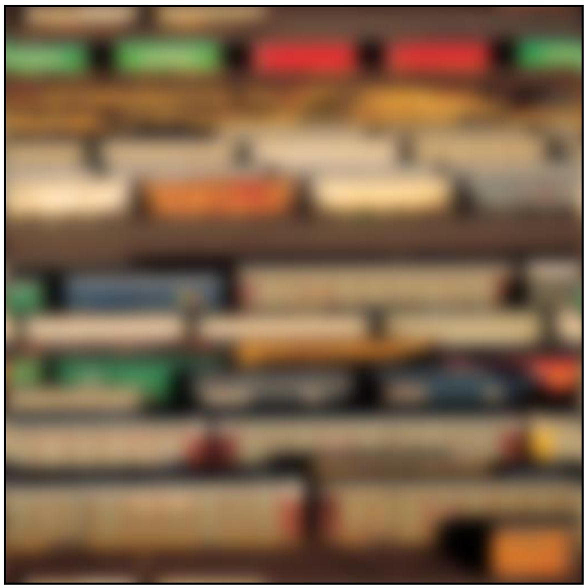
Ideal LPF

Original Image

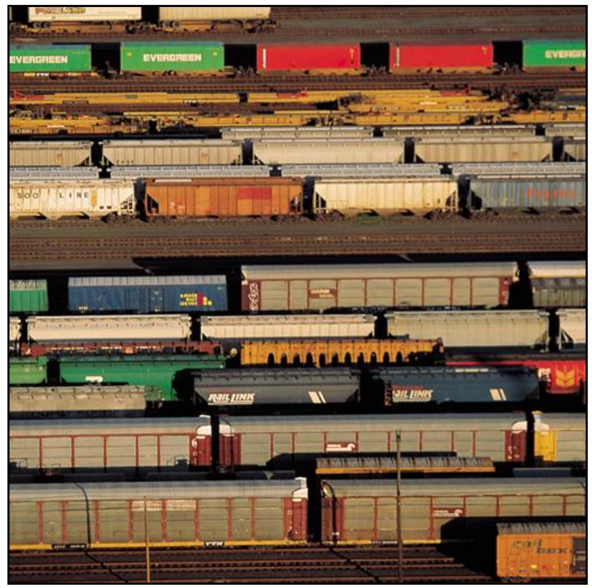
Ideal HPF*

*signed image:
0 mapped to 128

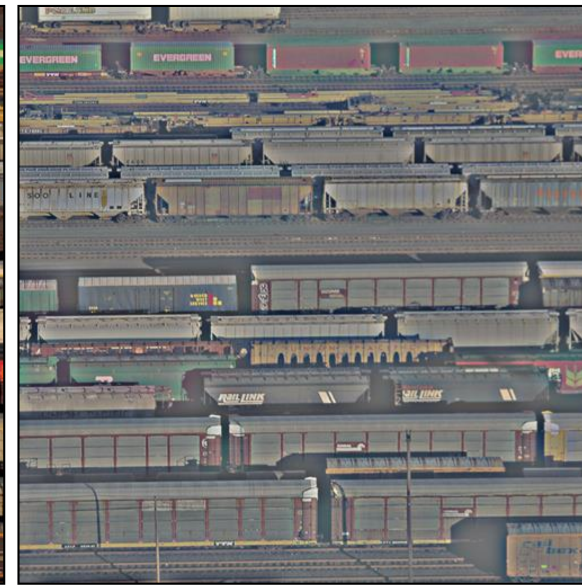
Gaussian Lowpass and Highpass Filters



Gaussian LPF



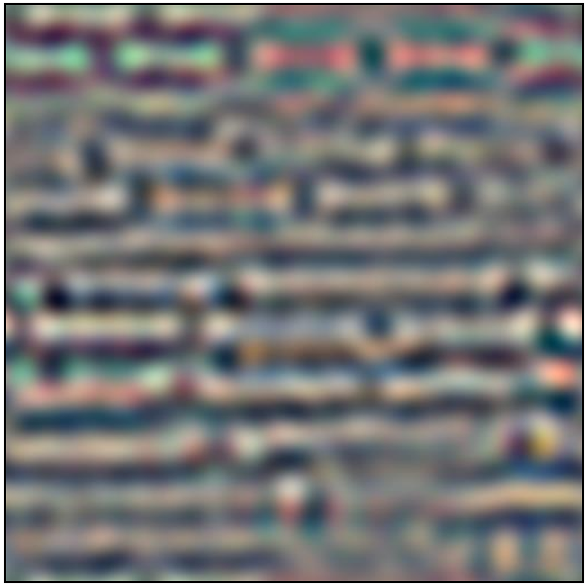
Original Image



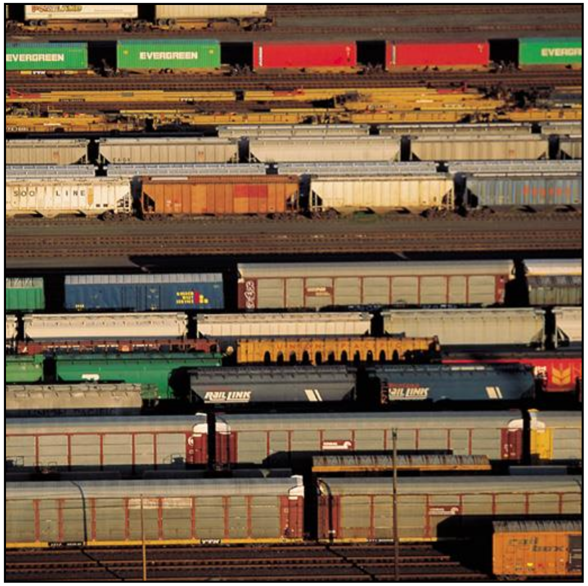
Gaussian HPF*

*signed image:
0 mapped to 128

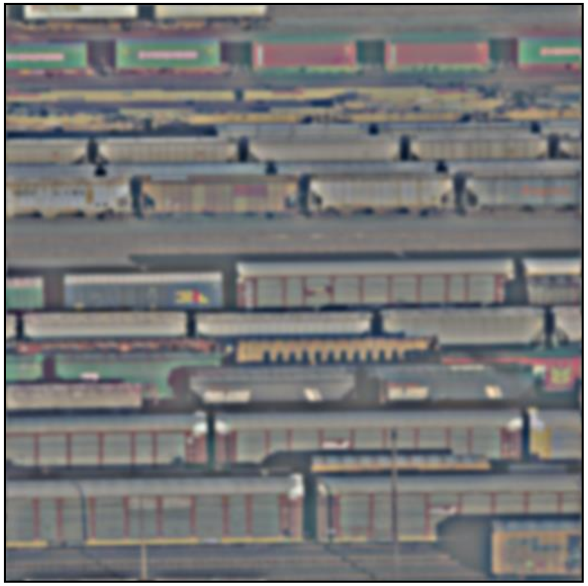
Ideal and Gaussian Bandpass Filters



Ideal BPF*



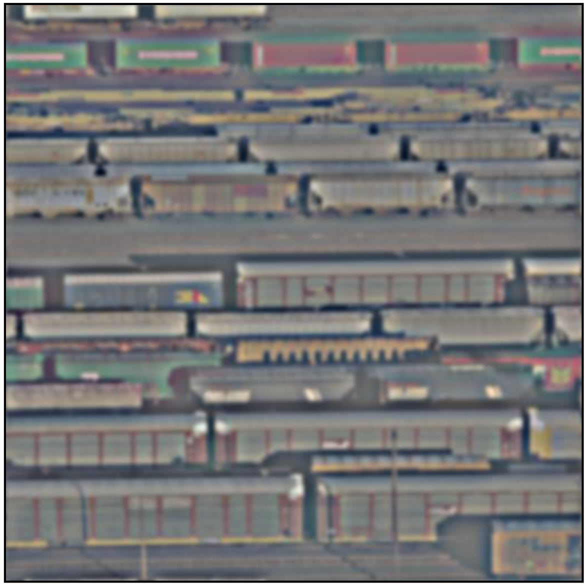
Original Image



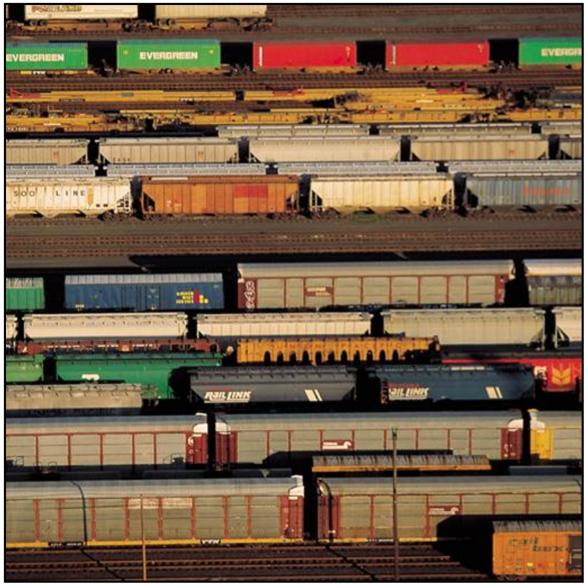
Gaussian BPF*

*signed image:
0 mapped to 128

Gaussian and Ideal Bandpass Filters



Gaussian BPF*

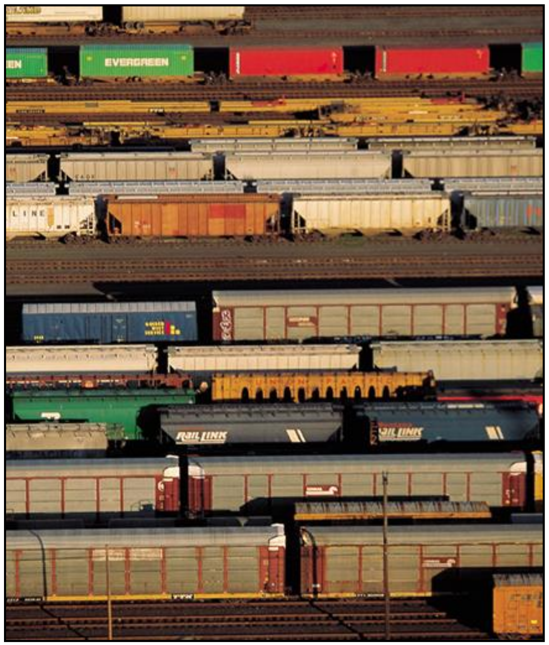


Original Image

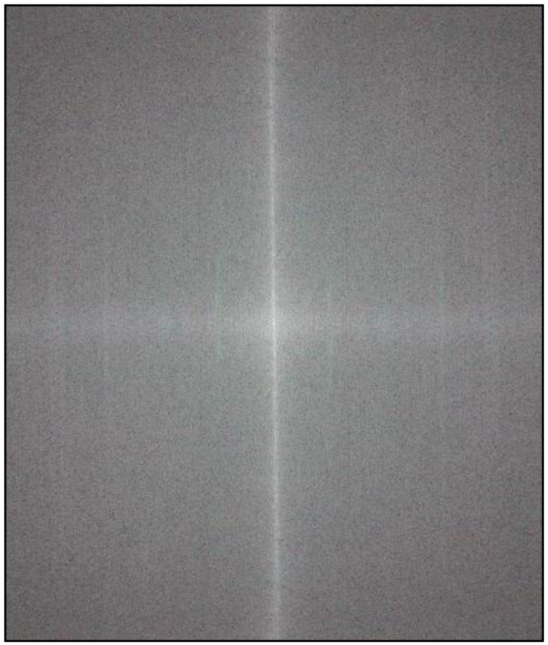
Ideal BPF*

Effects on Power Spectrum

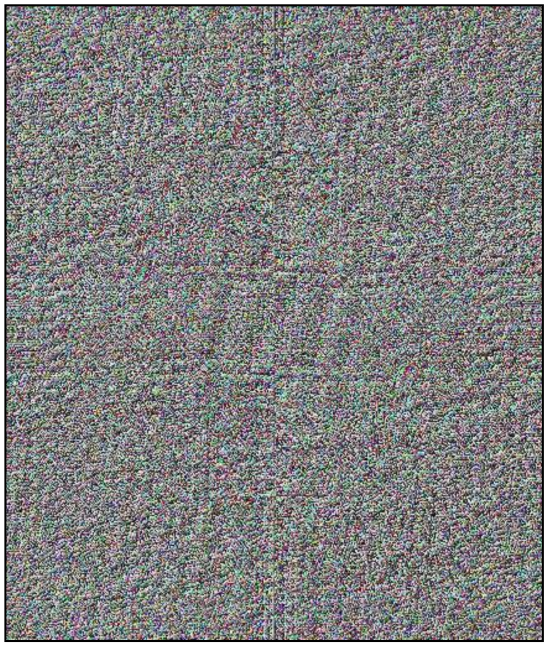
Power Spectrum and Phase of an Image



original image

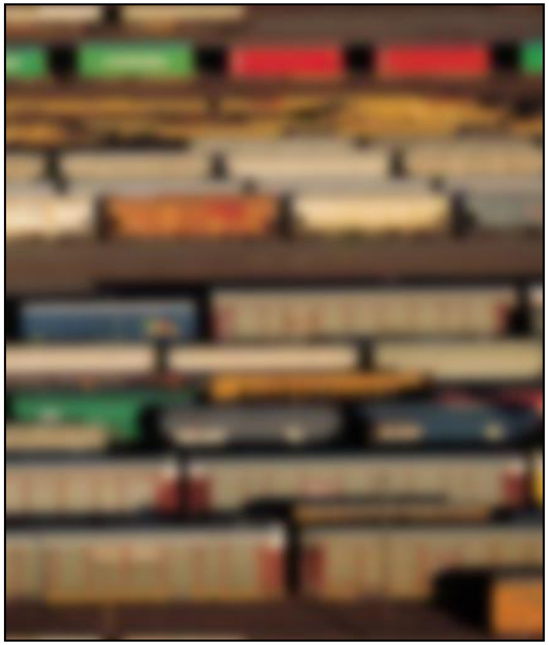


power spectrum

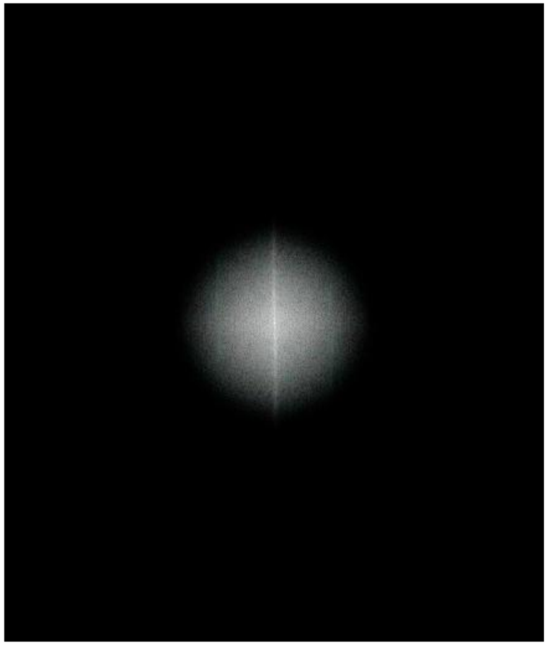


phase

Power Spectrum and Phase of a Blurred Image



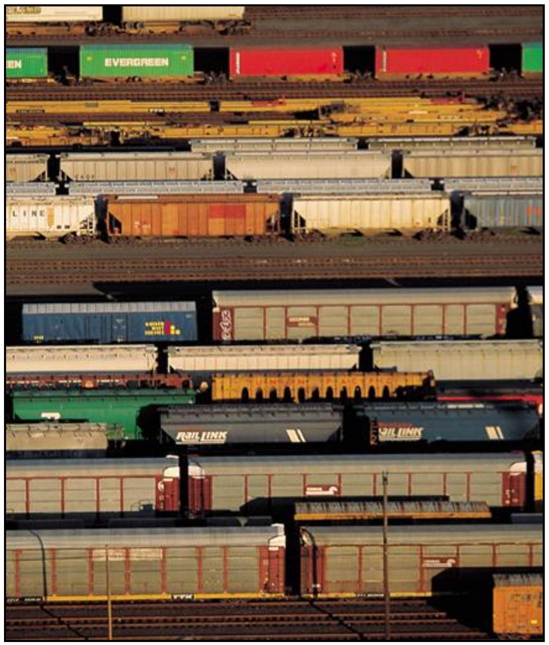
blurred image



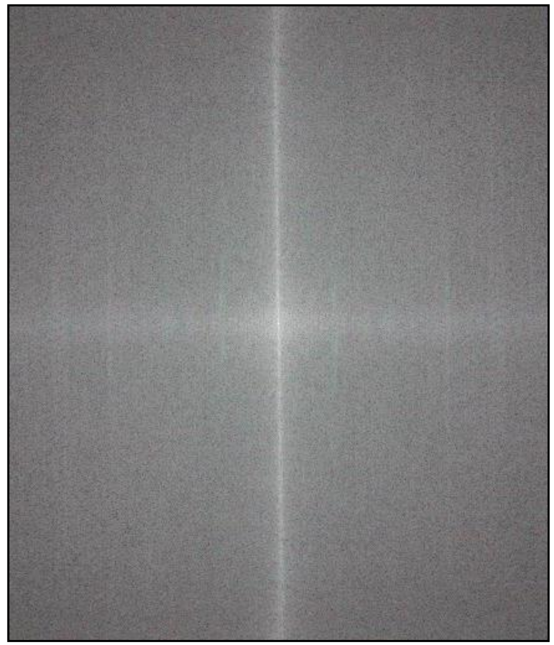
power spectrum

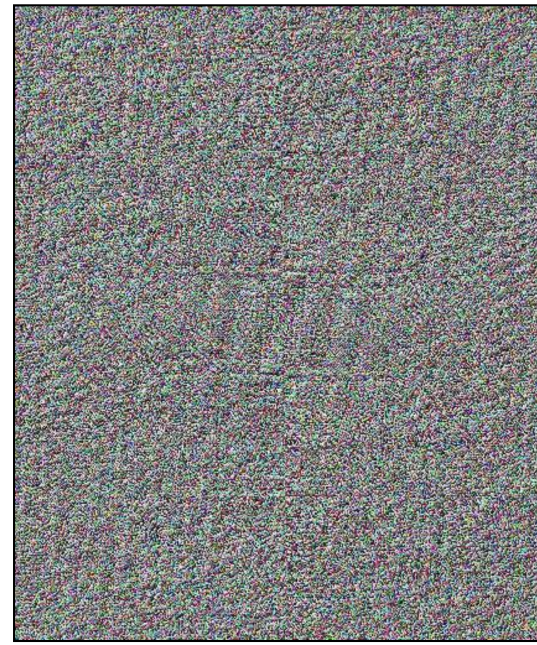
phase

Power Spectrum and Phase of an Image

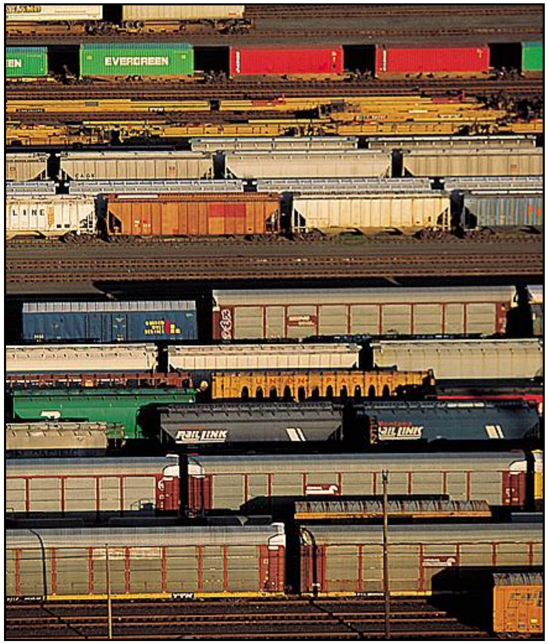
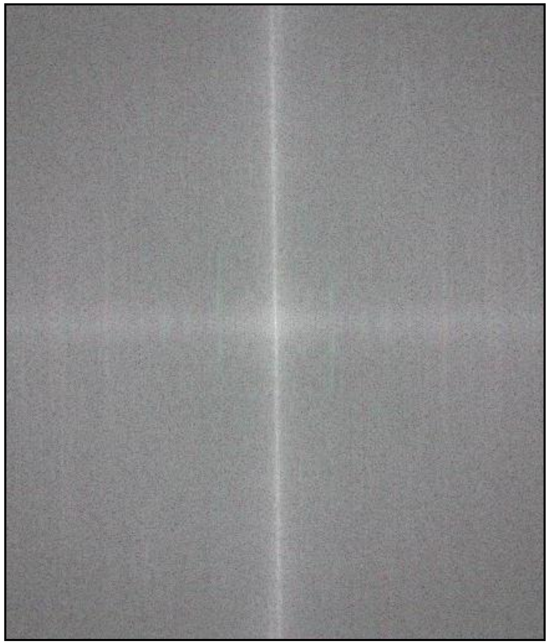
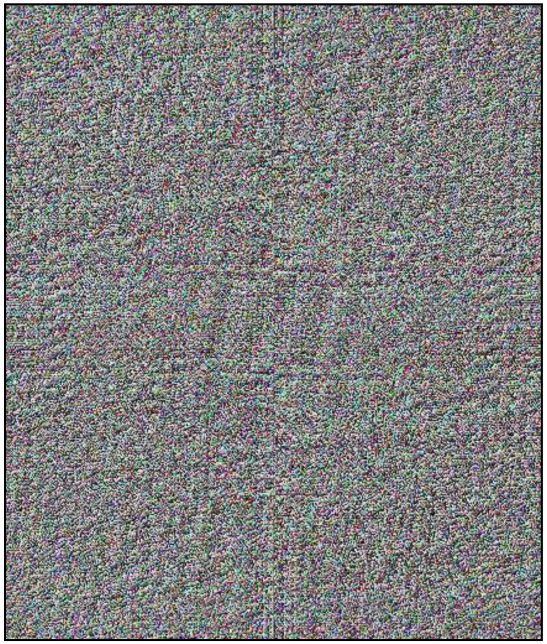


original image


power spectrum


phase

Power Spectrum and Phase of a Sharpened Image

sharpened image

power spectrum

phase