

# Blur an Image in the Spatial Frequency Domain with a Low Pass Filter

## ▼ Introduction

Here, we will blur in an image with a Gaussian Filter (effectively a low-pass filter) applied in the spatial frequency domain.

- First an image is imported
- The fourier transform of the image is then computed, and the image periodogram plotted.
- The fourier transform is multiplied by a Gaussian filter (this attenuates higher spatial frequencies - i.e. the finer detail is removed, leaving only the broad outline).
- The resulted is inverted to the image domain, giving a blurry image

> with( SignalProcessing ) :  
with( ImageTools ) :

## ▼ Import an Image

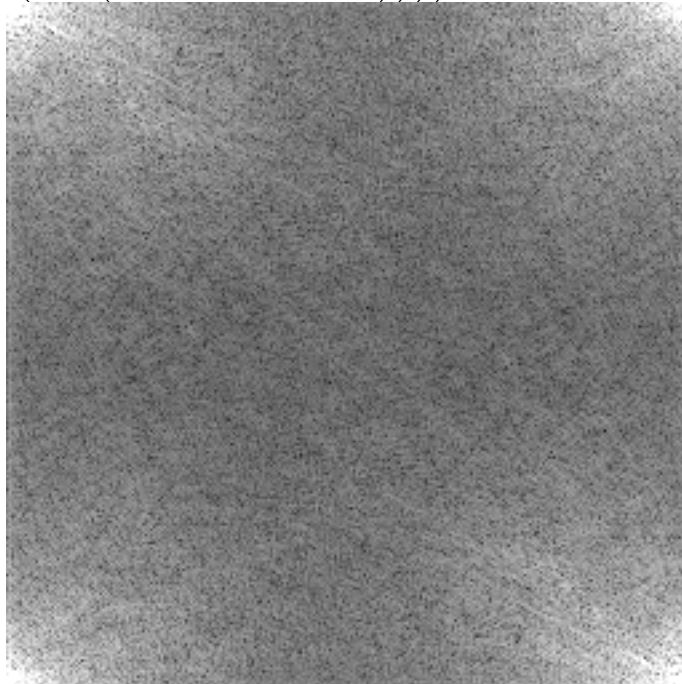
> img := Read( FileTools:-JoinPath( [ kernelopts( datadir ), "images", "phone.jpg" ] ) ) :  
Embed( img)



## ▼ Plot the Image Periodogram

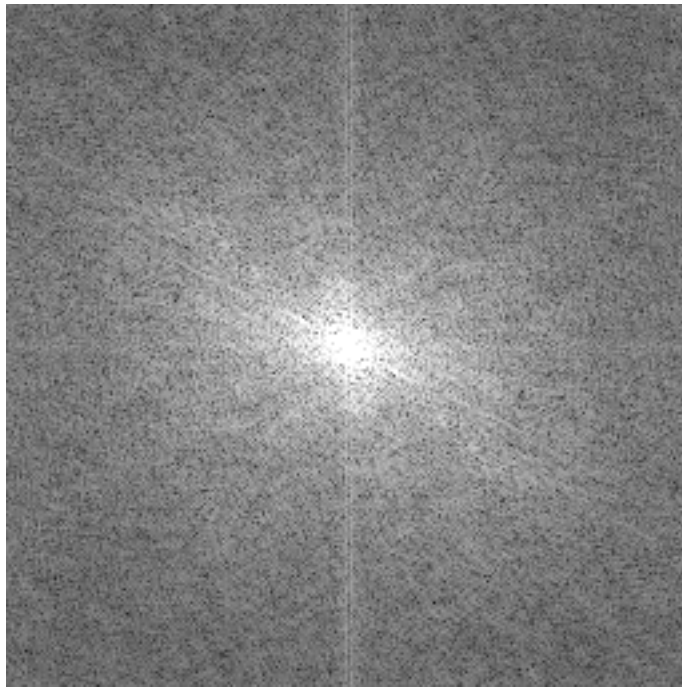
Calculate the Fourier transform of the image, and then visualize the spectra

```
> img_fourier_noshift := FFT( img ) :  
  Embed( Create( sqrt~( abs~( img_fourier_noshift ) ) ) )
```



Calculate the Fourier transform of the image, move the zero-frequency data to the center with FFT, and then visualize the spatial spectra,

```
> img_fourier_shift := FFTShift( FFT( img ) ) :  
  Embed( Create( sqrt~( abs~( img_fourier_shift ) ) ) )
```



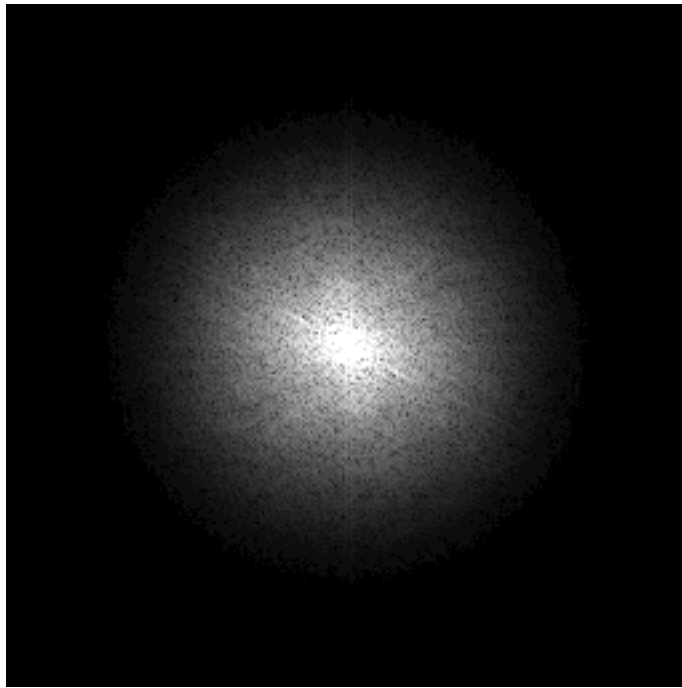
## ▼ Apply a Low Pass Filter

Multiply the spatial frequencies by a Gaussian mask (this is similar to a low pass filter that attenuates higher frequency data).

>  $\sigma := 20$  :  
 nRows, nCols := LinearAlgebra:-Dimension( img ) :

img\_fourier\_blur := img\_fourier\_shift · ~Matrix( nRows, nCols, ( i, j )  
 $\rightarrow \text{evalf}\left( e^{-\frac{(i - \text{nRows}/2)^2 + (j - \text{nCols}/2)^2}{2.0 \cdot \sigma^2}} \right), \text{datatype} = \text{float}[8] \right) :$

Embed( Create( sqrt~( abs~( img\_fourier\_blur ) ) ) )



## ▼ Invert the Spatial Frequencies back to the Image Domain

Move the zero frequency data back to their original positions, and then invert the spatial frequency data back to the image domain.

```
> img_low_pass := Re~( InverseFFT( FFTShift( img_fourier_blur ) ) ) :  
  Embed( FitIntensity( Create( img_low_pass ) ) )
```

