Università Ca' Foscari Venezia

Computer Vision

Filtering in the Frequency Domain

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco
DAIS, Ca' Foscari University of Venice
Academic year 2016/2017

Università Ca' Foscari Venezia

Introduction

During the past century, and especially in the past 50 years, entire industries and academic disciplines have flourished as a result of Fourier's ideas.

The "discovery" of a fast Fourier transform (FFT) algorithm in the early 1960s revolutionized the field of signal processing.

The goal of this lesson is to give a working knowledge of how the Fourier transform and the frequency domain can be used for image filtering

Università Ca' Foscari Venezia

Complex numbers

A complex number C is defined as

$$C = R + jI$$

Where R and I are real numbers, and j is an imaginary number so that $j^2 = -1$

R is called **real part** and I is called **imaginary part**

$$(a + jb) + (c + jd) = (a + c) + j(b + d)$$

$$(a + jb)(c + jd) = (ac - bd) + j(bc + ad)$$

$$\overline{a + jb} = a - jb$$

Complex numbers

A complex number can be represented in polar coordinates:

$$C = (R + jI) = \sqrt{R^2 + I^2}(\cos\theta + j\sin\theta)$$

$$\theta = \arctan(I/R)$$

Euler formula:

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$C = \sqrt{C\overline{C}}\, e^{j\theta}$$

Università Ca' Foscari Venezia

Fourier's basic idea

Any **periodic** function (with period T) can be expressed as the sum of sines and cosines of different frequencies, each multiplied by a different coefficient

> Fourier serie

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}x}$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(x) e^{-j\frac{2\pi n}{T}x} dx \qquad \text{for } n \in \mathbb{Z}$$

Università
Ca' Foscari
Venezia

Fourier's basic idea

Functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function.

> Fourier transform

$$f(x) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu x} d\mu$$

Continuous Fourier transform

Fourier transform:

$$F(\mu) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\mu x} dx$$

Even if f(x) is real, its transform in general is a complex function.

The domain of the fourier transform is called the *frequency domain*

Fourier transform is essentially a change of basis from spatial domain to frequency domain

Continuous Fourier transform

Fourier transform:

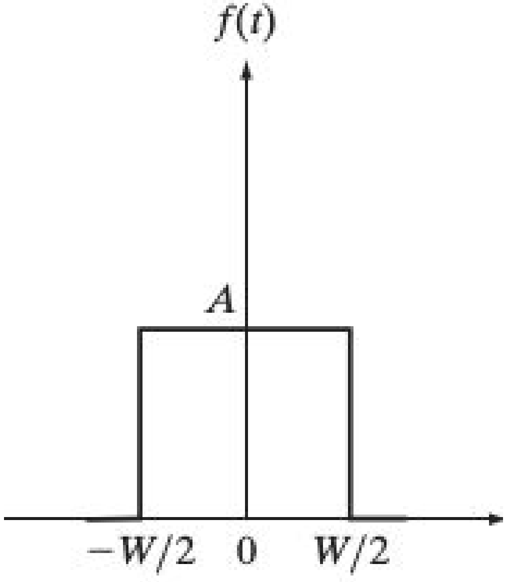
$$F(\mu) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\mu x} dx$$

Inverse Fourier transform:

$$f(x) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu x} d\mu$$

Continuous Fourier transform

Suppose that we want to compute the Fourier transform of the following function:



$$f(t) = \begin{cases} 0 & t < -W/2 \\ A & -W/2 \le t \le W/2 \\ 0 & t > W/2 \end{cases}$$

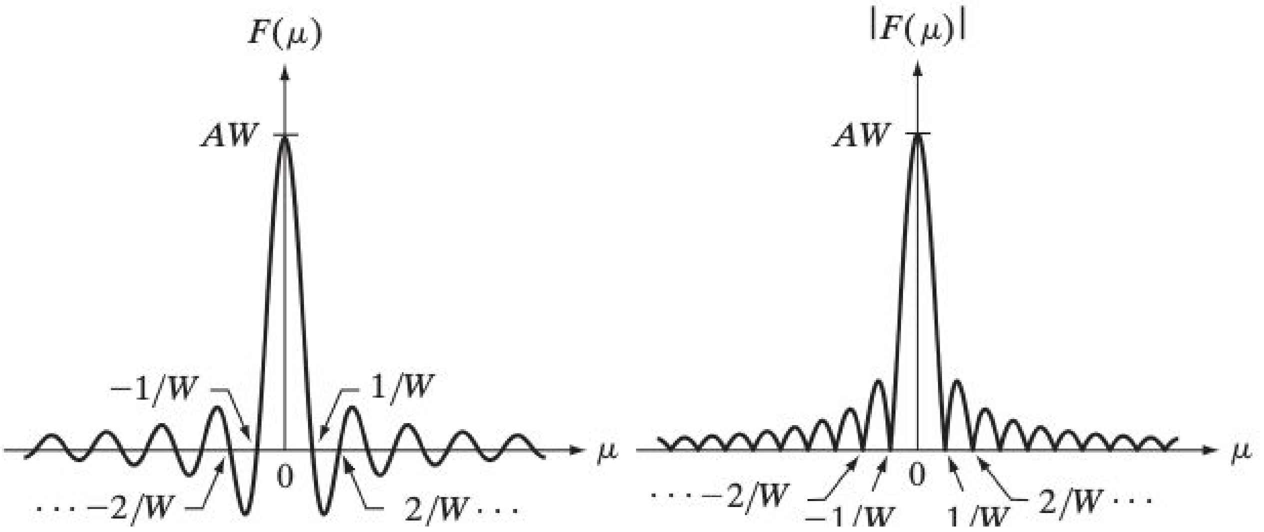
Continuous Fourier transform

$$F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi\mu t}\, dt = \int_{-W/2}^{W/2} A e^{-j2\pi\mu t}\, dt$$

$$= \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_{-W/2}^{W/2} = \frac{-A}{j2\pi\mu}\left[e^{-j\pi\mu W} - e^{j\pi\mu W}\right]$$

$$= \frac{A}{j2\pi\mu}\left[e^{j\pi\mu W} - e^{-j\pi\mu W}\right]$$

$$= AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}$$



Università Ca' Foscari Venezia

DFT

In practice we work with finite functions (assumed to be periodic) composed by a finite number of M discrete samples

Discrete Fourier transform:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \qquad u = 0, 1, \ldots, M-1$$

Discrete Inverse Fourier transform:

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \qquad x = 0, 1, \ldots, M-1$$

Università Ca' Foscari Venezia

DFT

In terms of sines and cosines, the DFT can be expressed as:

$$F(u) = \sum_{x=0}^{M-1} f(x)\big(\cos(-2\pi ux/M) + j\sin(-2\pi ux/M)\big)$$

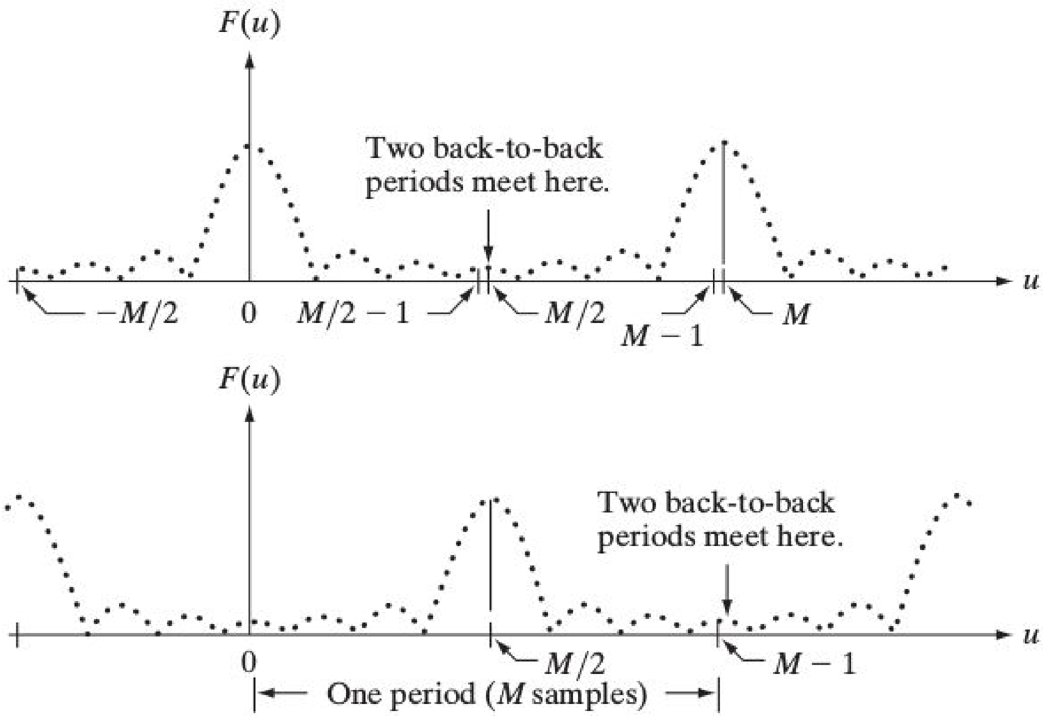
$$u = 0, 1, \ldots, M-1$$

Università Ca' Foscari Venezia

Periodicity

When we compute the DFT of a real function, the Fourier transform is periodic over the interval.

The transform data in the interval from 0 to M-1 consists of two back-to-back half periods meeting at point M/2.



2D DFT

DFT can be computed for any-dimensional input function. In particular, the 2D DFT is useful when working with images

2D Discrete Fourier transform:

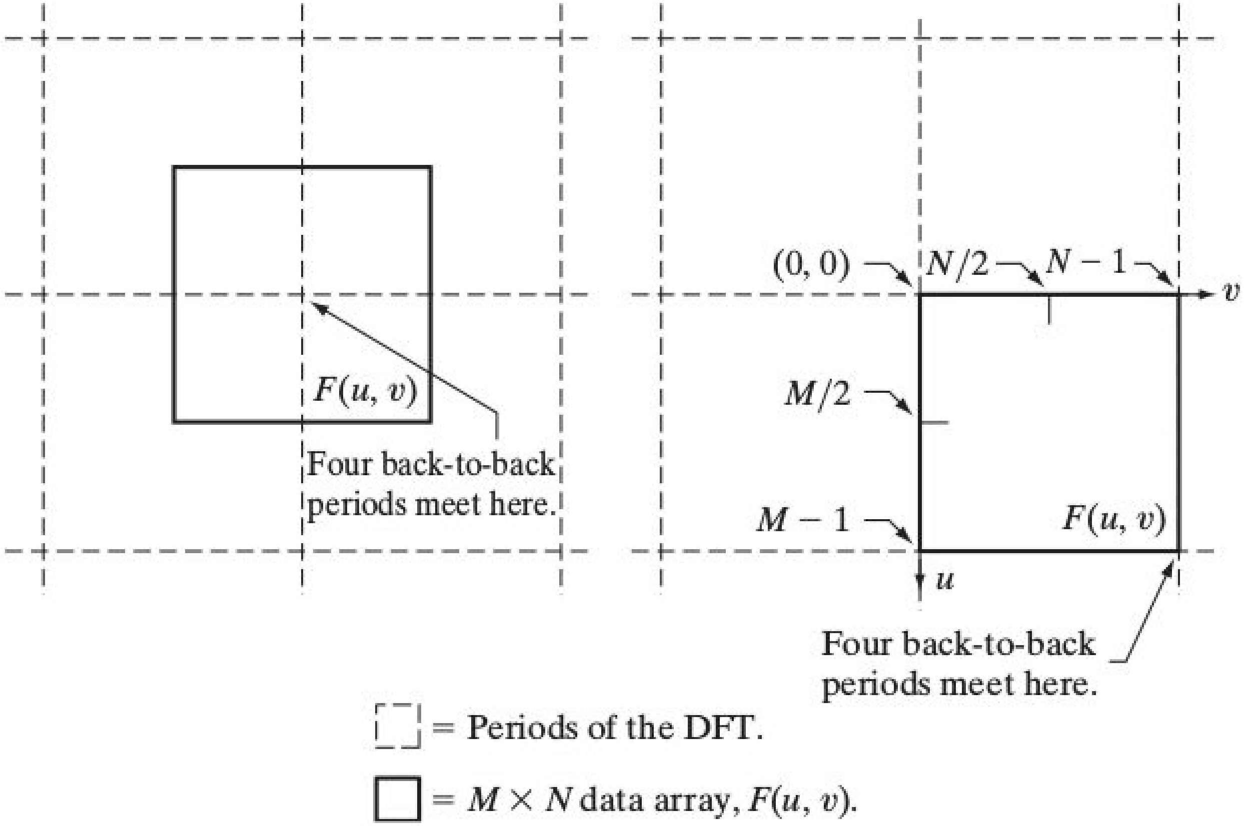
$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

Discrete Inverse Fourier transform:

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{-j2\pi(ux/M + vy/N)}$$

Periodicity

Also in the 2D case we have periodicity both in u and v direction



Spectrum and phase angle

Università Ca' Foscari Venezia

Because the 2-D DFT is complex in general, it can be expressed in polar form:

$$F(u,v) = |F(u,v)|e^{j\,\theta(u,v)}$$

Where the magnitude

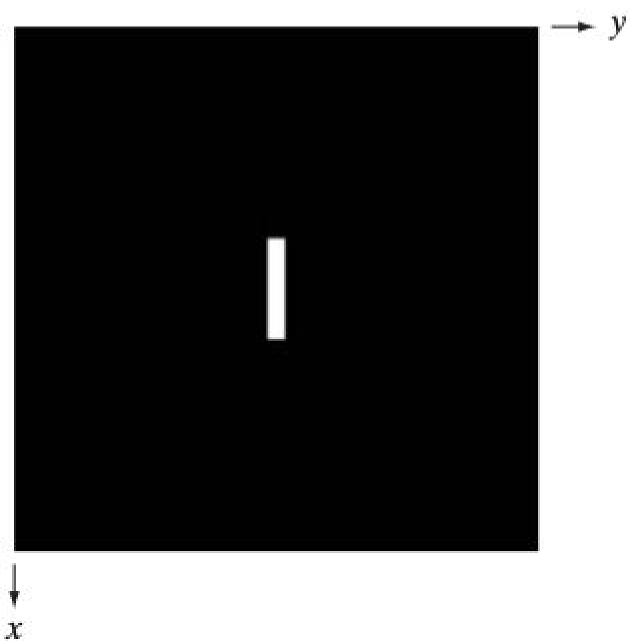
$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

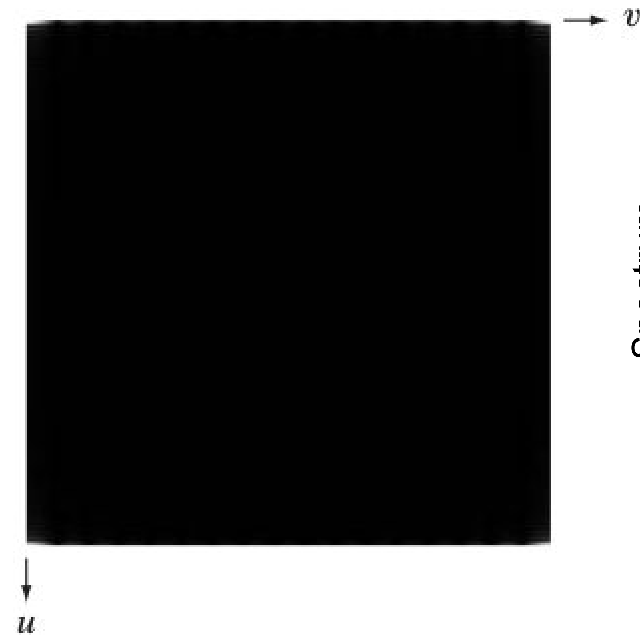
Is called the **Fourier spectrum** and

$$\theta(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$$

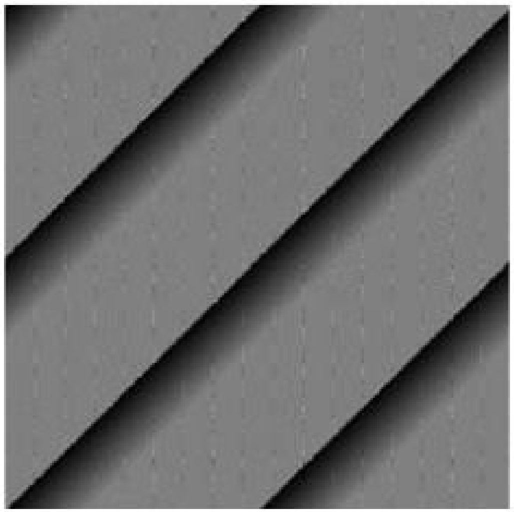
Is called the **phase angle**

Spectrum and phase angle



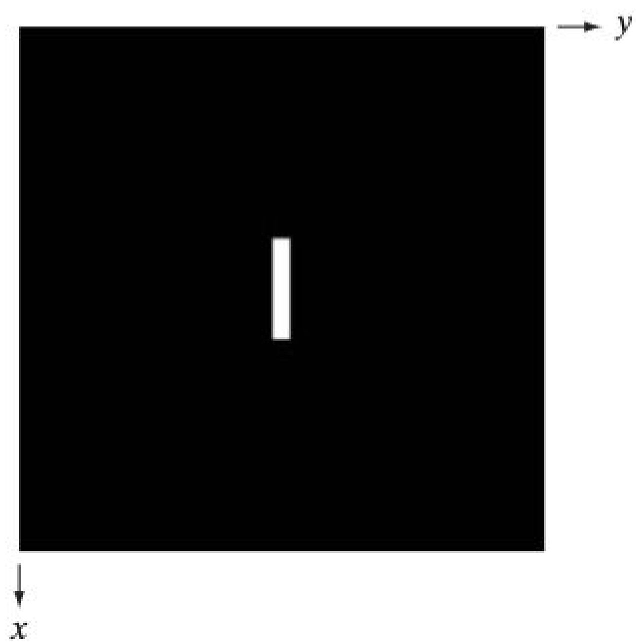


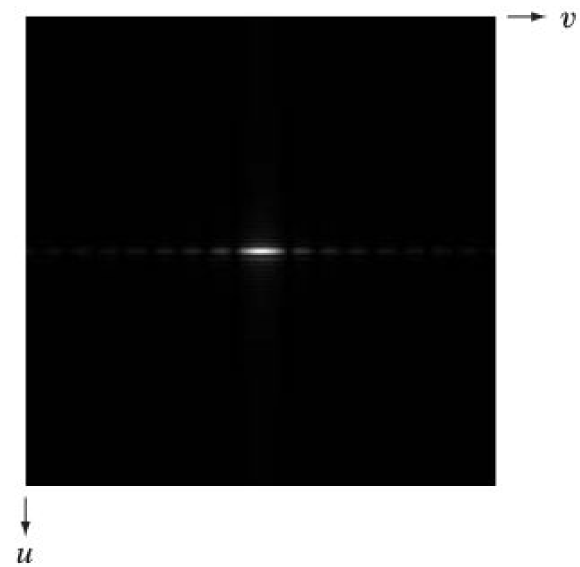
Spectrum



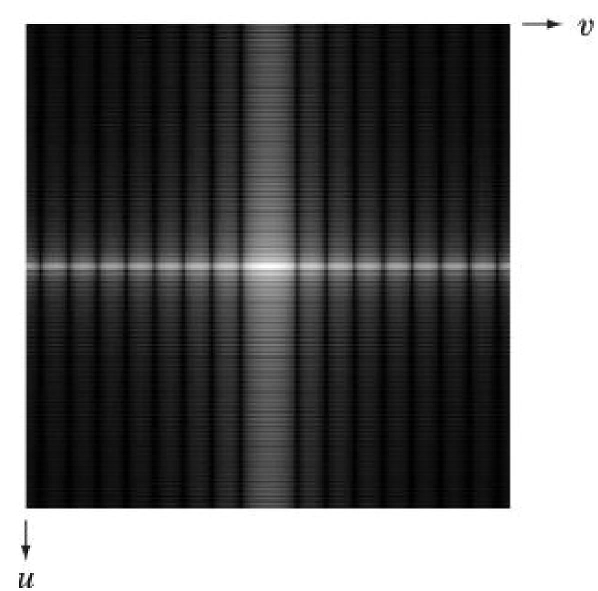
Phase angle

Spectrum and phase angle



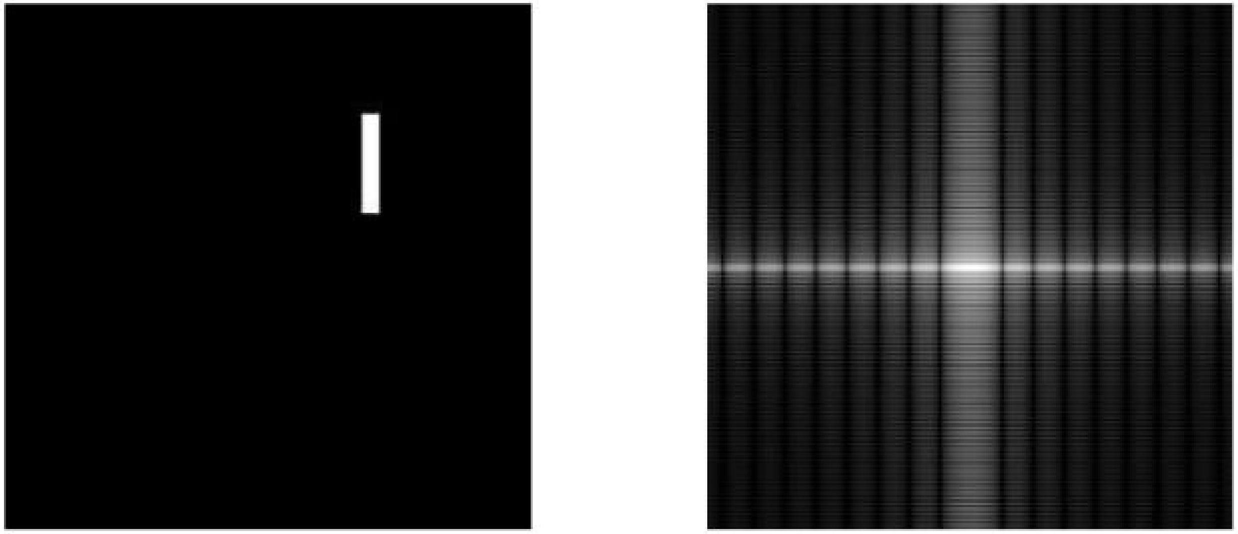


Centered Spectrum



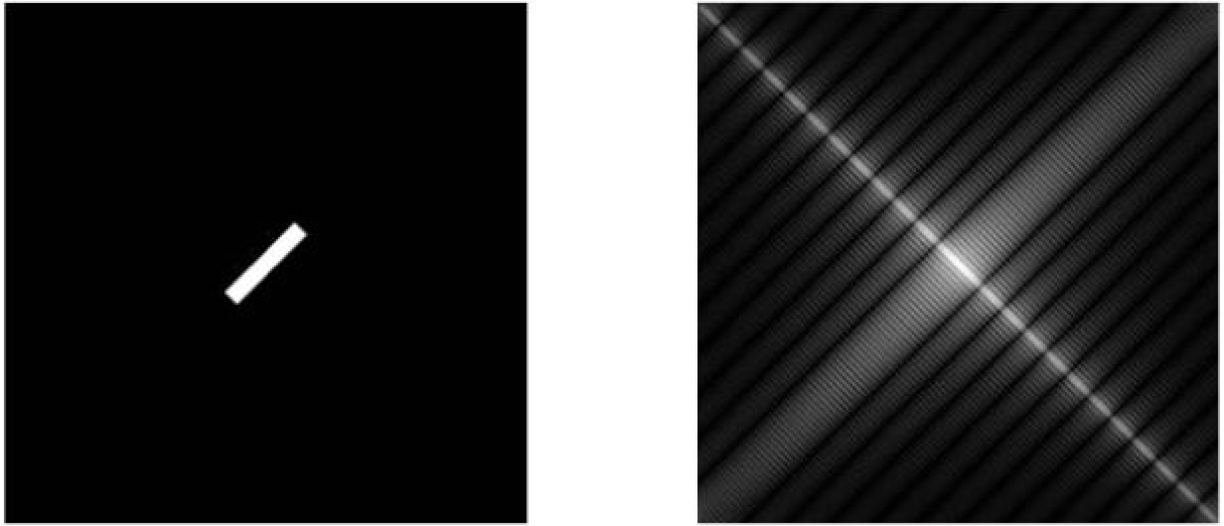
Log Centered Spectrum

DFT Spectrum



Translating f(x, y) do not change the Fourier spectrum but only the phase angle.

DFT Spectrum



Rotating f(x, y) by an angle theta rotates F(u, v) by the same angle, vice-versa

DFT Spectrum and Phases

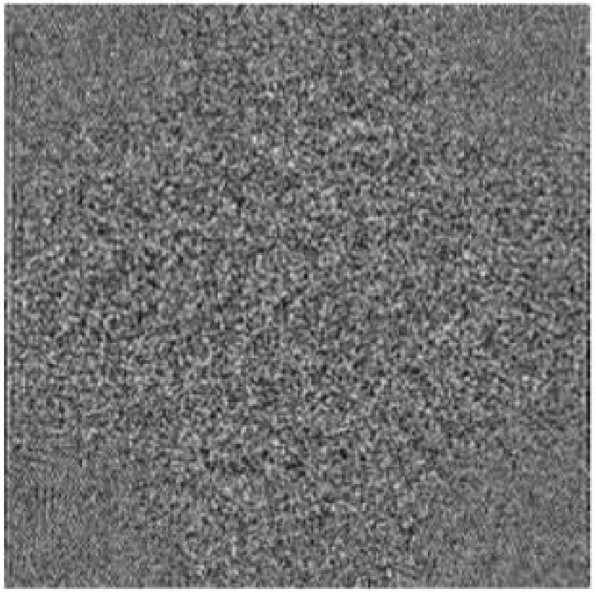
The components of the DFT spectrum determine the **amplitudes of the sinusoids** that combine to form the resulting image

> determine the intensities in the image

The phase is a measure of **displacement of the various sinusoids** with respect to their origin.

> carry much of the information about where discernable objects are located

DFT Spectrum and Phases

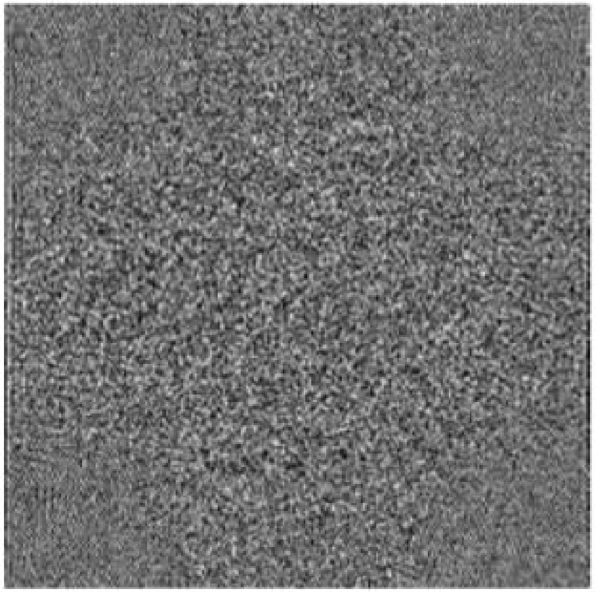


Phase angle

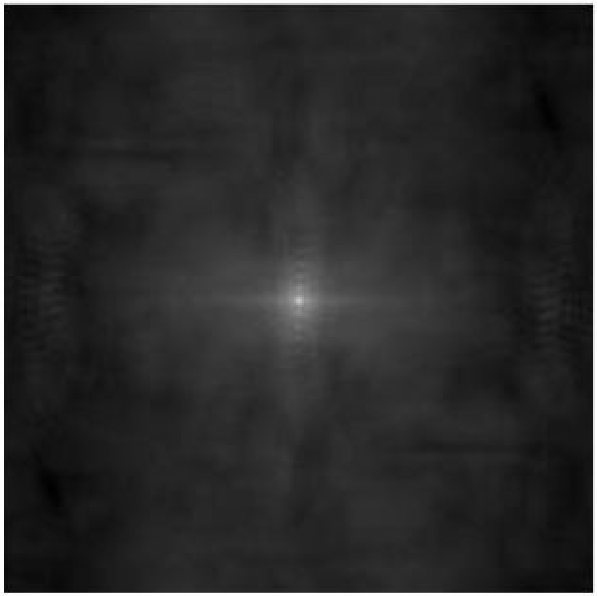
Reconstruction with phase angle only

Università Ca' Foscari Venezia

DFT Spectrum and Phases

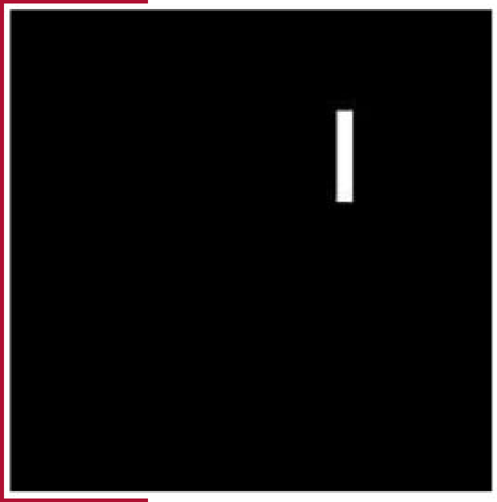


Phase angle



Reconstruction with spectrum only

DFT Spectrum and Phases



Reconstruction with woman phase and rectangle spectrum

Università
Ca' Foscari
Venezia

2D Convolution theorem

$$f(x,y) \star h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v) \star H(u,v)$$

Università Ca' Foscari Venezia

Frequency domain filtering

Filtering in the frequency domain consists of **modifying the Fourier transform** of an image and then computing the inverse transform to obtain the processed result.

$$g(x,y) = \Im^{-1}[H(u,v)F(u,v)]$$

Inverse Fourier transform

Filter function

DFT of the input image

Low-pass, high-pass

low frequencies in the transform are related to slowly varying intensity components in an image

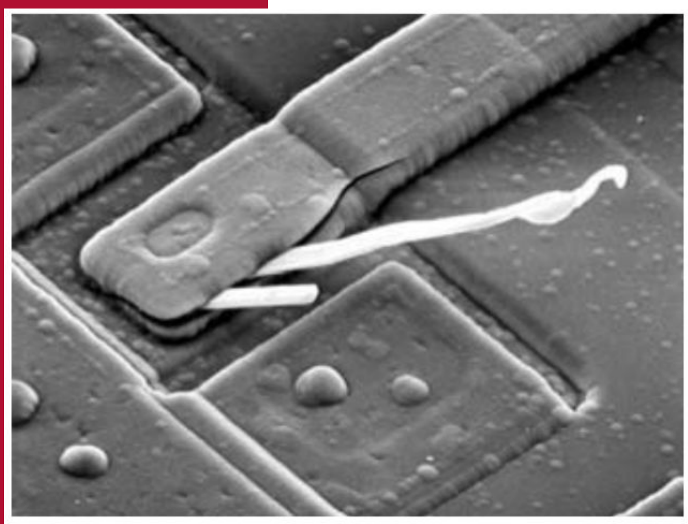
high frequencies are caused by sharp transitions in intensity, such as edges and noise

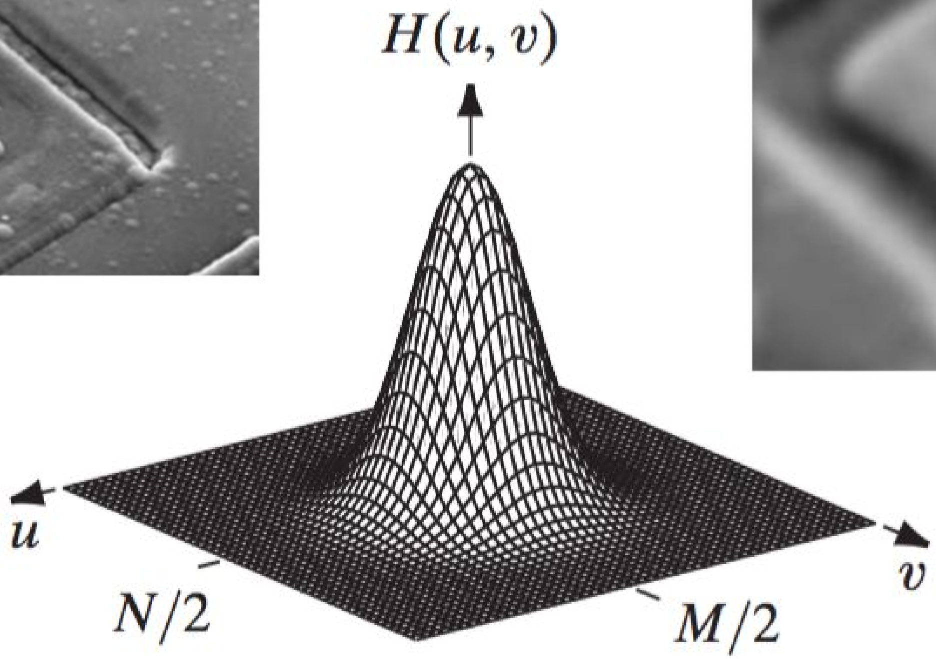
A filter H(u, v) that attenuates high frequencies while passing low frequencies (**low-pass filter**) blurs an image

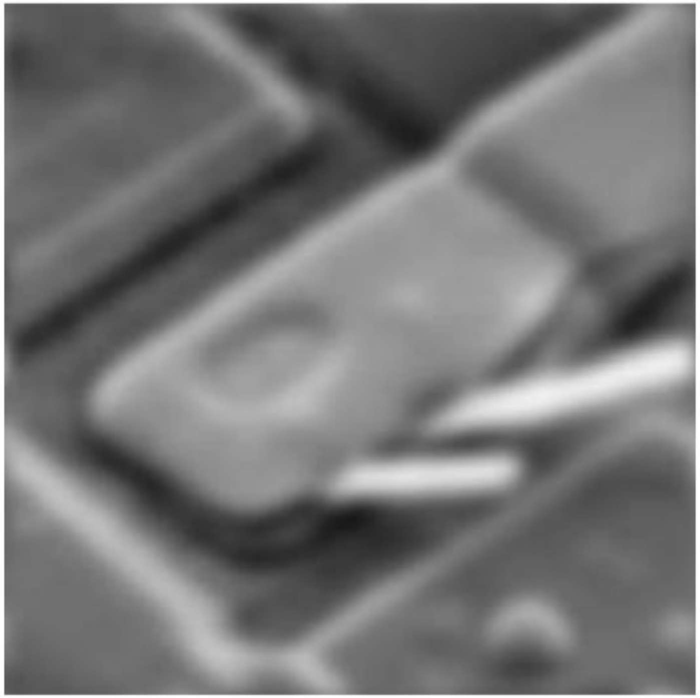
A **high-pass filter** (which attenuates low frequencies) enhances sharp detail, but cause a reduction in contrast in the image.

Università
Ca' Foscari
Venezia

Low-pass







Ideal Low-pass filter

Università Ca' Foscari Venezia

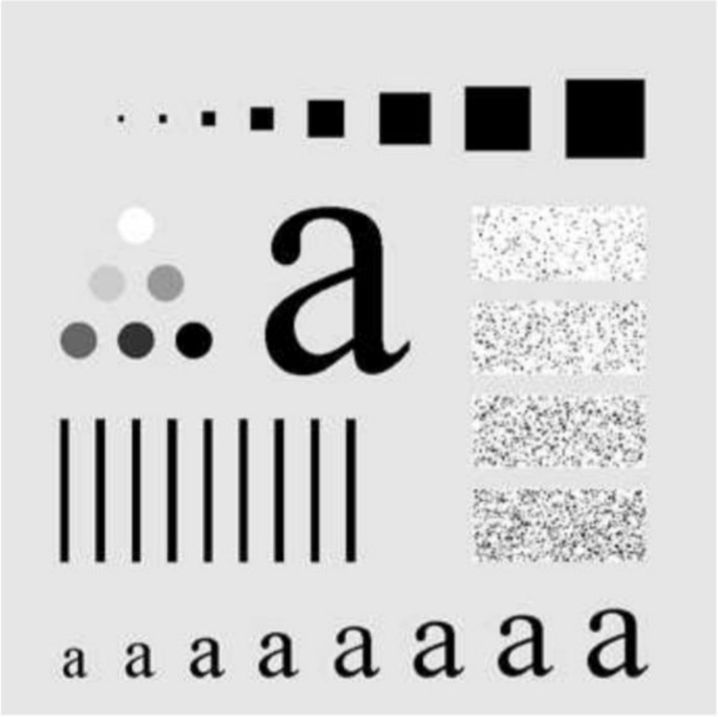
An ideal low-pass filter ILPF is defined by:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

The point of transition between H(u, v) = 1 and H(u, v)= 0 is called the **cutoff frequency**

Università
Ca' Foscari
Venezia

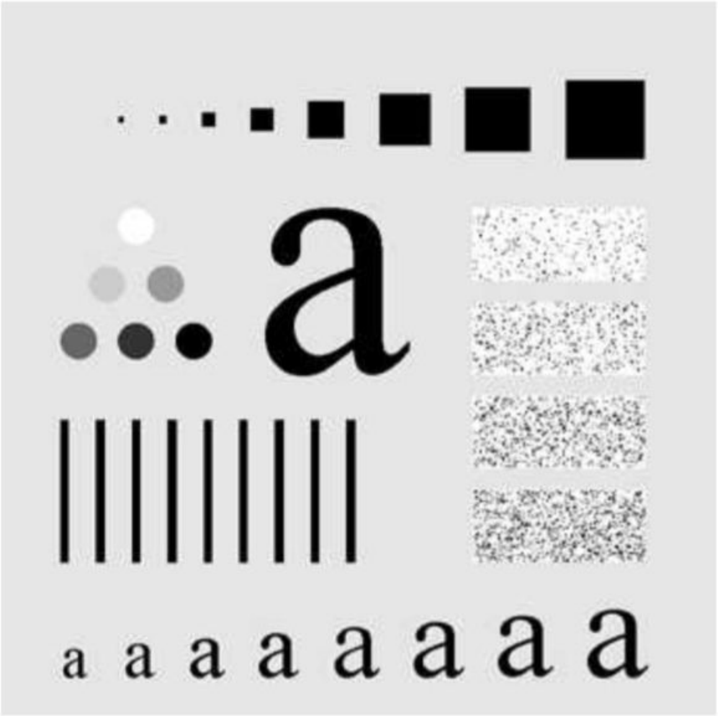
ILPF



ILPF with cutoff frequency =60

Università
Ca' Foscari
Venezia

ILPF



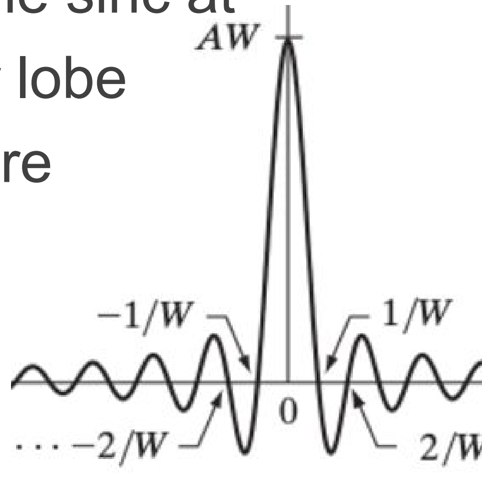
ILPF with cutoff frequency =30

ILPF

The blurring and ringing properties of ILPFs can be explained using the convolution theorem:

Because a cross section of the ILPF in the frequency domain looks like a box filter, a cross section of the corresponding spatial filter has the shape of a sinc.

Convolving a sinc with an impulse copies the sinc at the location of the impulse. The sinc center lobe causes the blurring, while the outer lobes are responsible for ringing.

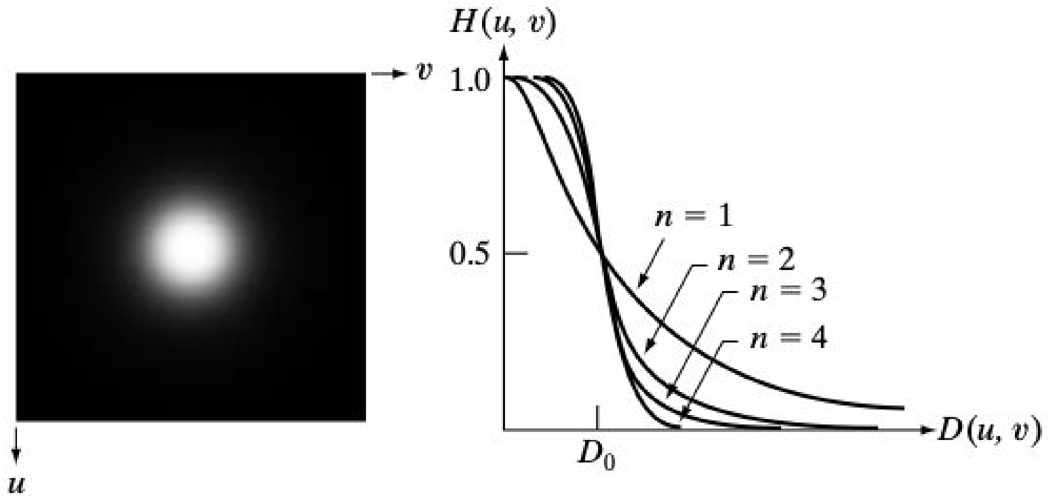


Butterworth Low-pass filter

Università Ca' Foscari Venezia

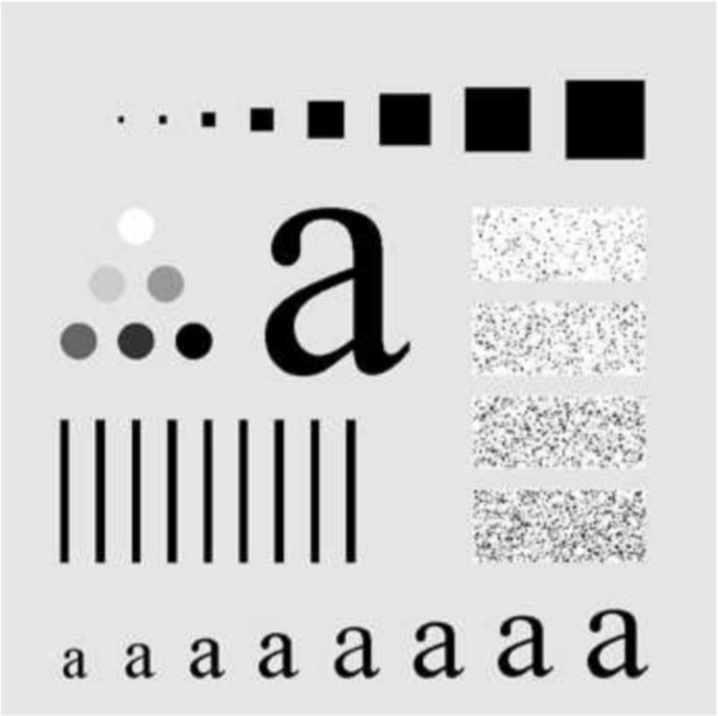
A Butterworth low-pass filter (BLPF) of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$



Università
Ca' Foscari
Venezia

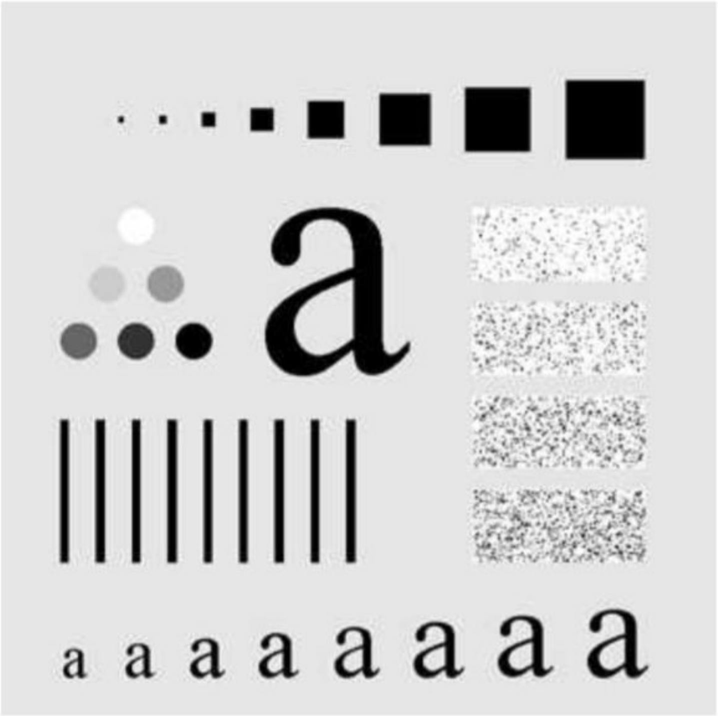
BLPF



BLPF of order 2 and cutoff frequency =60

Università
Ca' Foscari
Venezia

BLPF

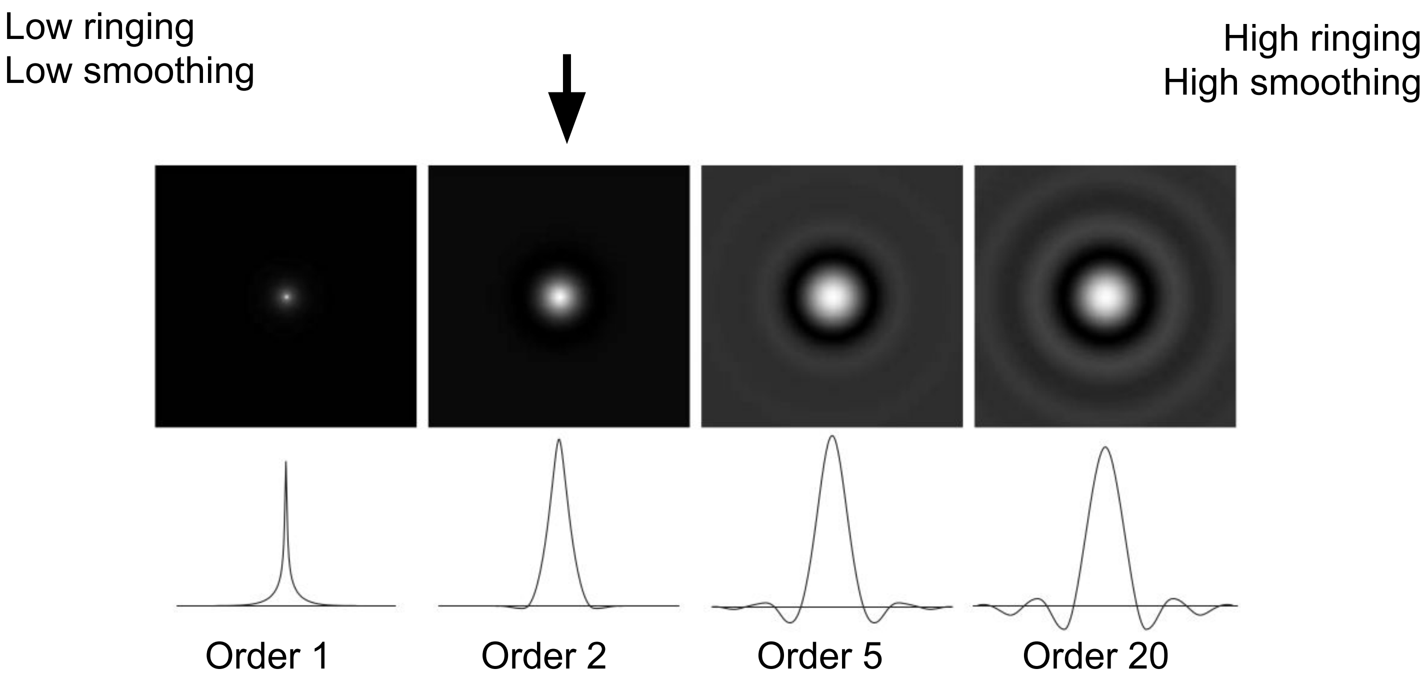


BLPF of order 2 and
cutoff frequency =30

Università
Ca' Foscari
Venezia

BLPF

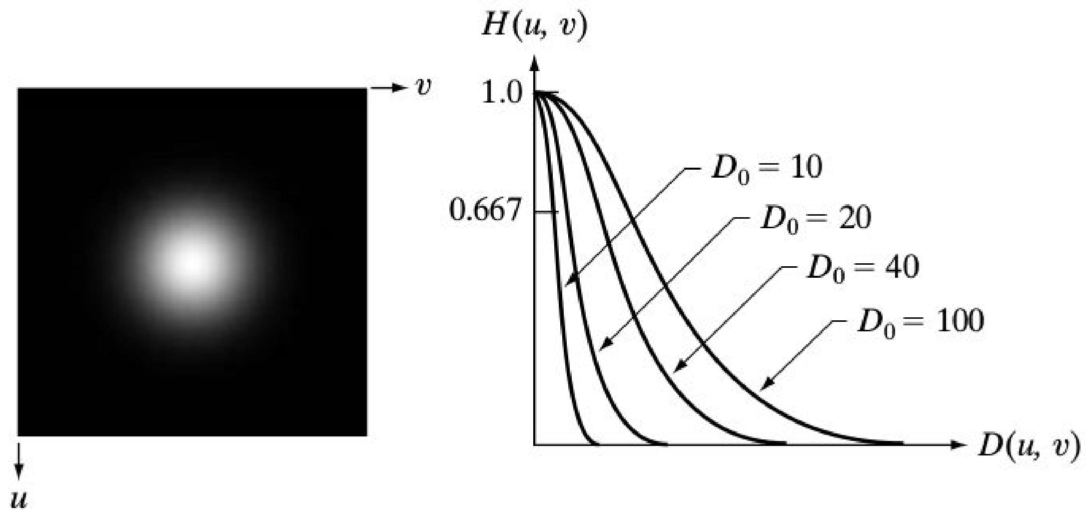
Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that gives a clear cutoff between passed and filtered frequencies.



Gaussian Low-pass filter

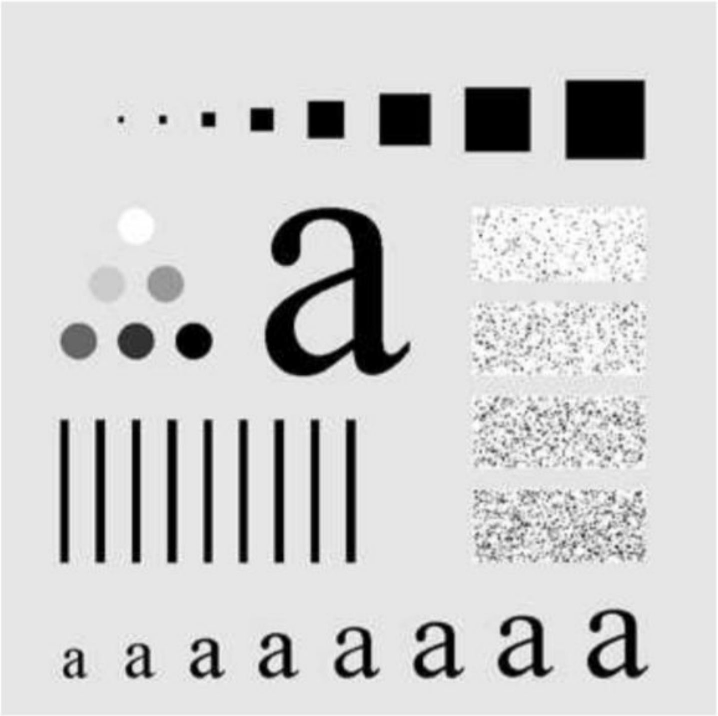
A Gaussian low-pass filter (GLPF) is defined as

$$H(u, v) = e^{-D^2(u,v)/2D_0^2}$$



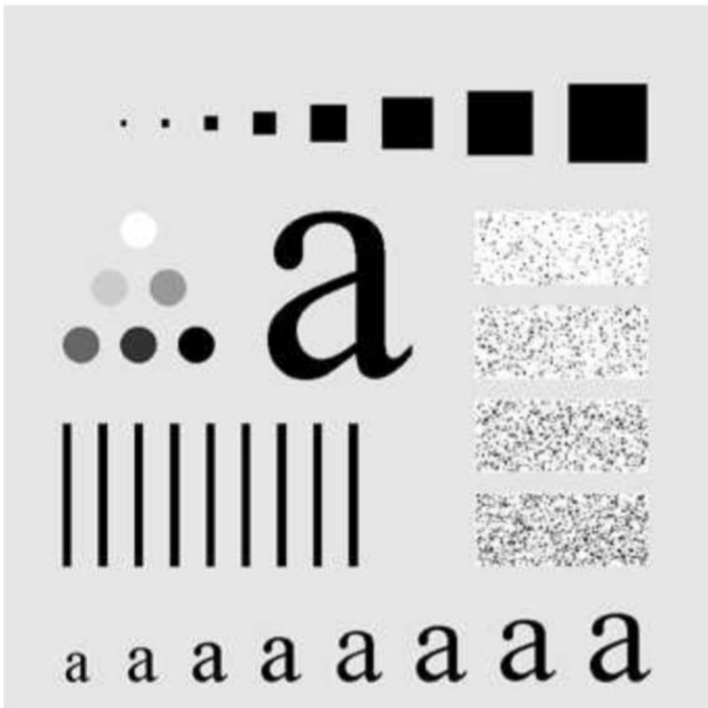
Università
Ca' Foscari
Venezia

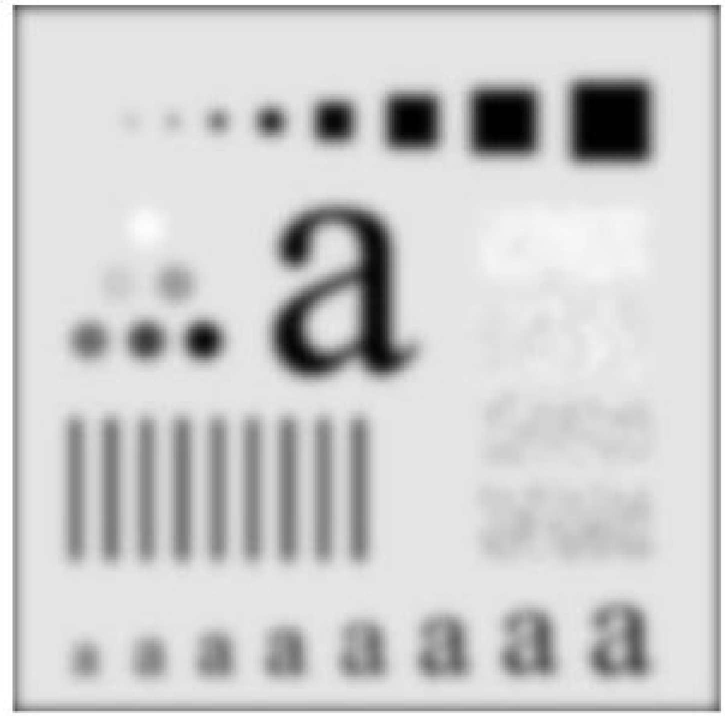
GLPF



GLPF cutoff
frequency =60

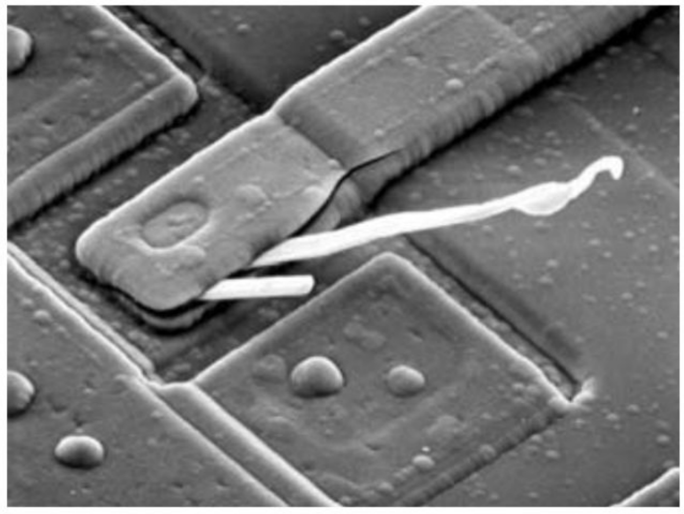
GLPF

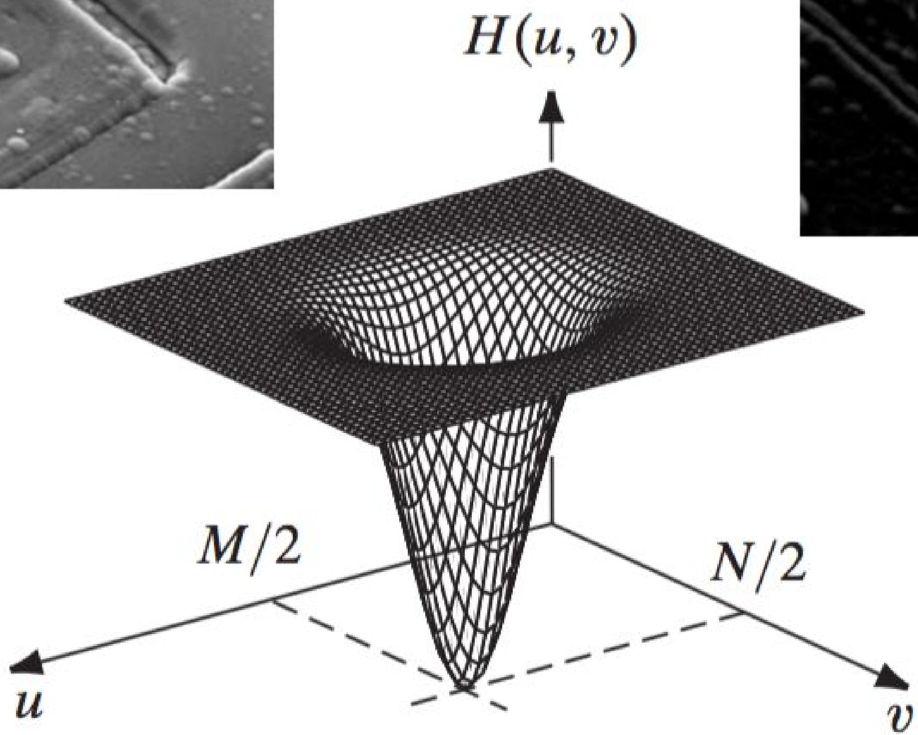




GLPF cutoff frequency =30

High-pass



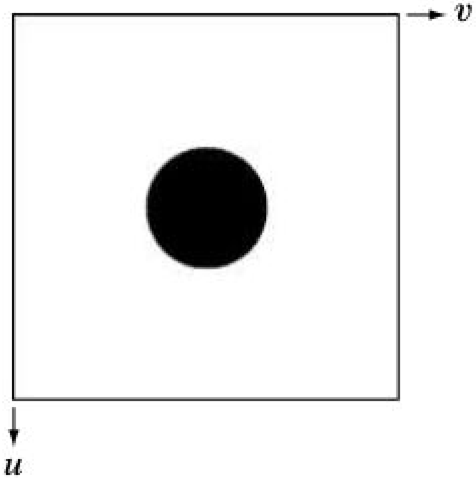


Ideal High-pass filter

An ideal high-pass filter IHPF is defined by:

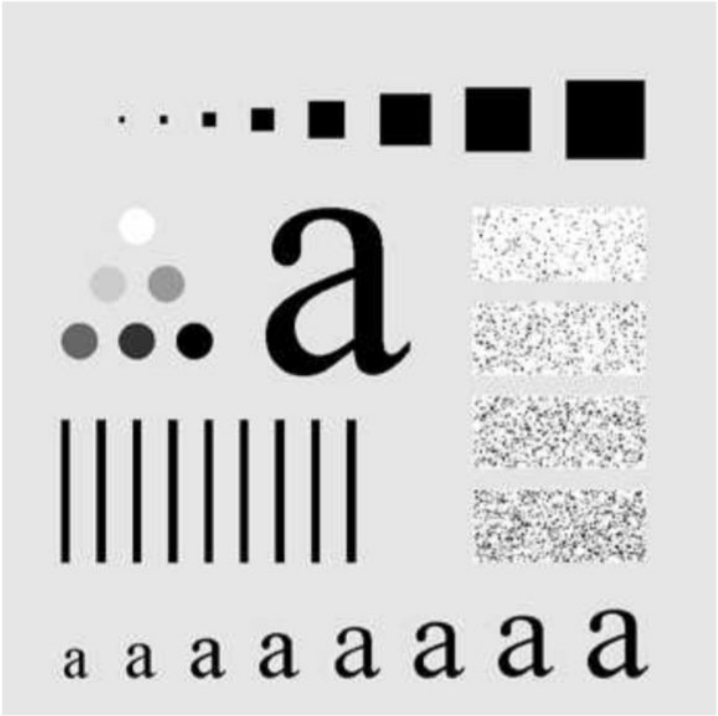
$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

IHPF is the opposite of an ILPF



Università Ca' Foscari Venezia

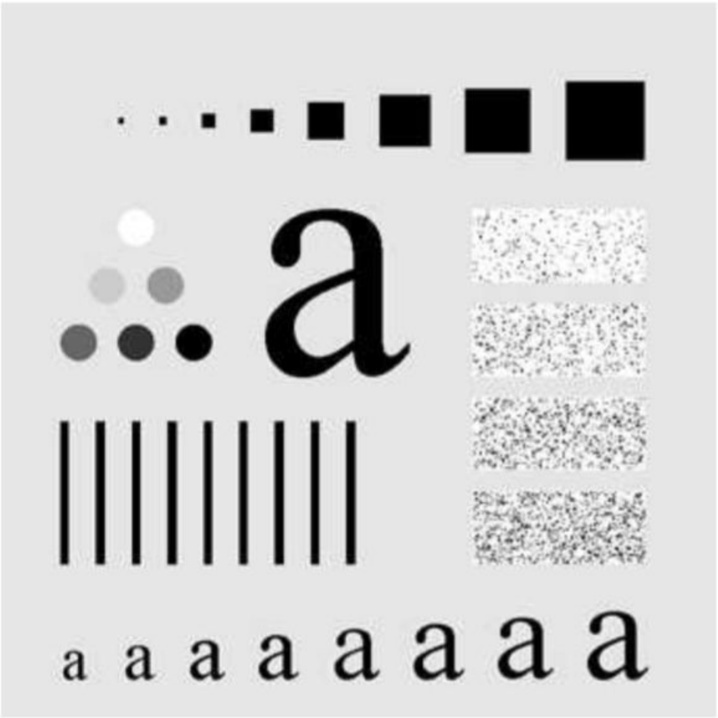
IHPF



IHPF with cutoff frequency =60

Università
Ca' Foscari
Venezia

IHPF



IHPF with cutoff frequency =30

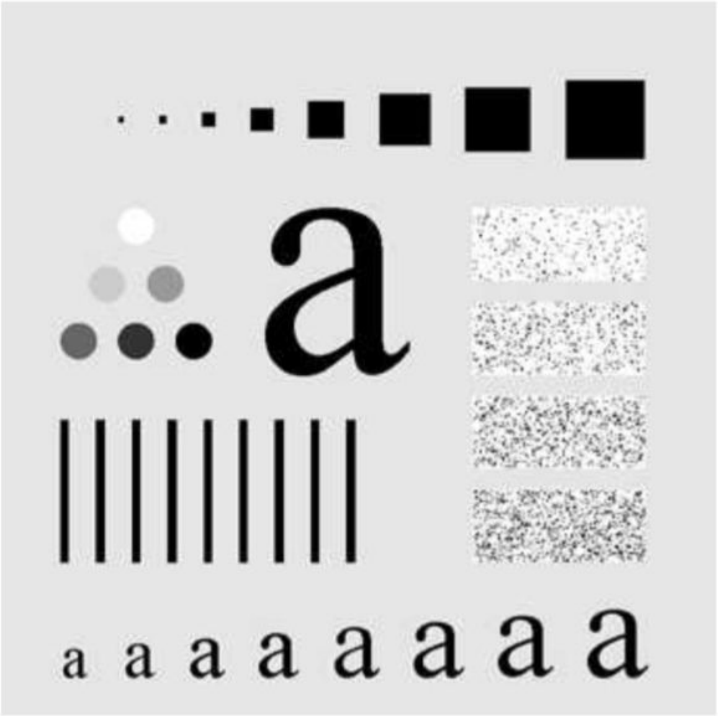
Butterworth High-pass filter

A Butterworth high-pass filter (BHPF) of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

Università
Ca' Foscari
Venezia

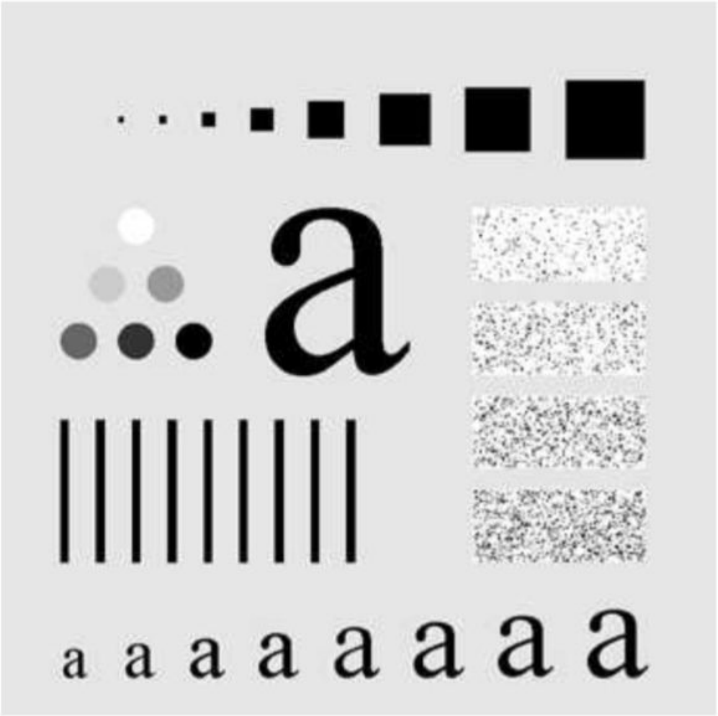
BHPF



BLPF of order 2 and
cutoff frequency =60

Università
Ca' Foscari
Venezia

BHPF



BHPF of order 2 and
cutoff frequency =30

Laplacian

It can be shown that the Laplacian can be implemented in the frequency domain using the filter

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

The laplacian of an image can then be computed as:

$$\nabla^2 f(x, y) = \Im^{-1}\{H(u, v)F(u, v)\}$$

Laplacian filtering:

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

Notch Filters

A notch filter rejects (or passes) frequencies in a predefined neighborhood about the center of the frequency rectangle.

Must be symmetric about the origin, so a notch with center at (u0, v0) must have a corresponding notch at location (-u0, -v0)

> Otherwise the filter is not zero-phase-shift

Products of high-pass filters whose centers have been translated to the centers of the notches.

Università Ca' Foscari Venezia

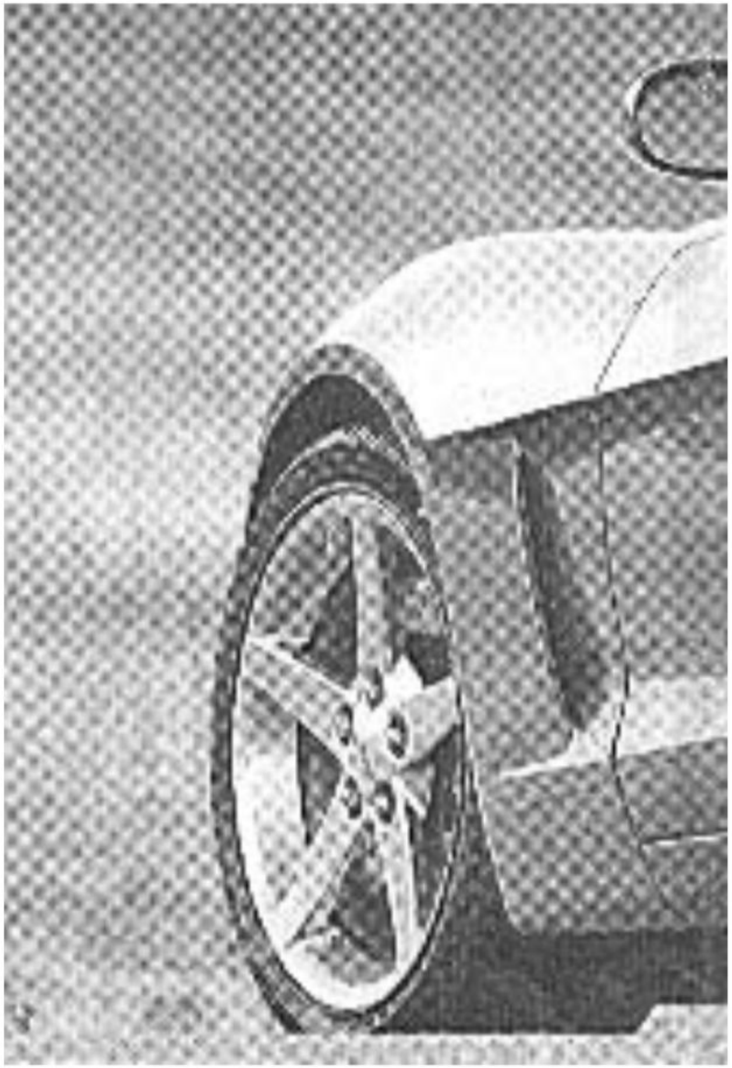
Notch Filters

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

High-pass filter centered at (u_k,v_k)

High-pass filter centered at $(-u_k,-v_k)$

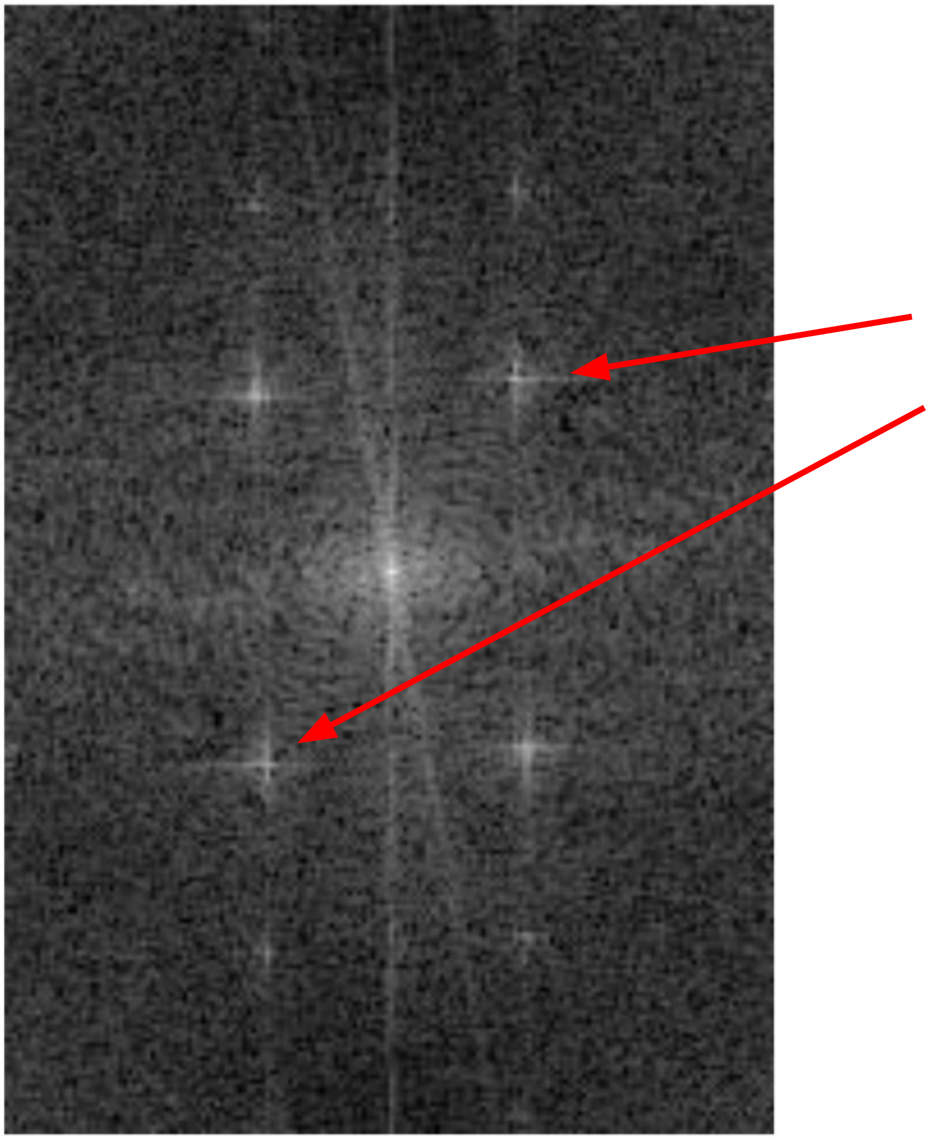
Notch Filters



Newspaper image showing moiré pattern composed by the combination of different sinusoids

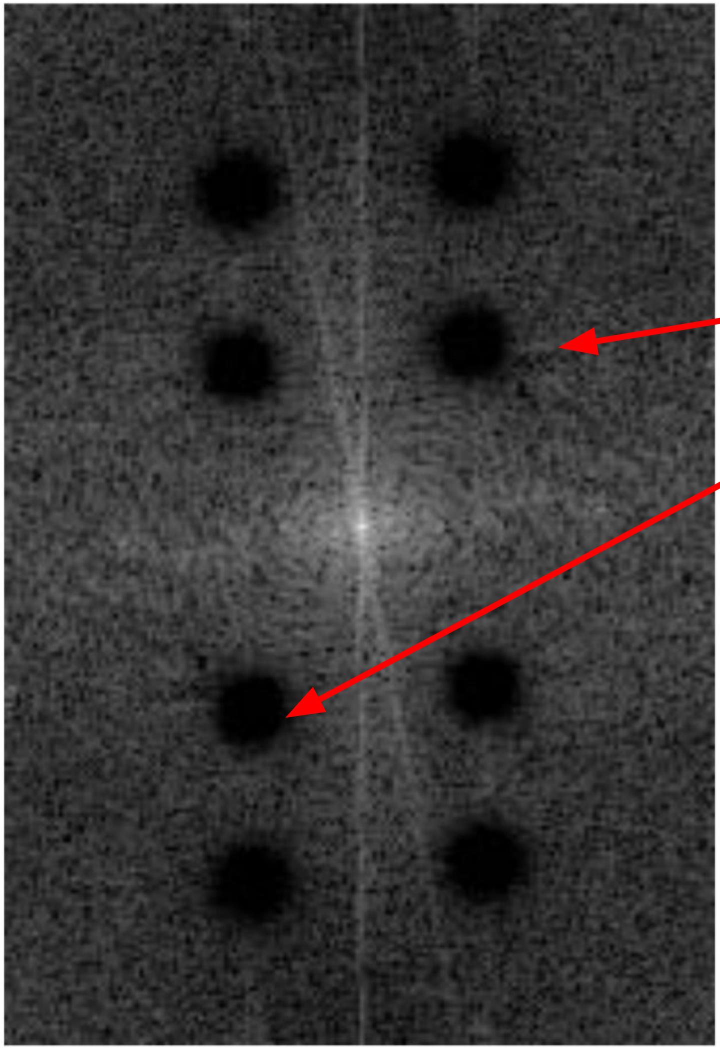
The Fourier transform of a pure sine is a pair of conjugate symmetric impulses.

Notch Filters



Fourier spectrum showing clear symmetric impulses bursts as a result of the near periodicity of the moiré pattern.

Notch Filters

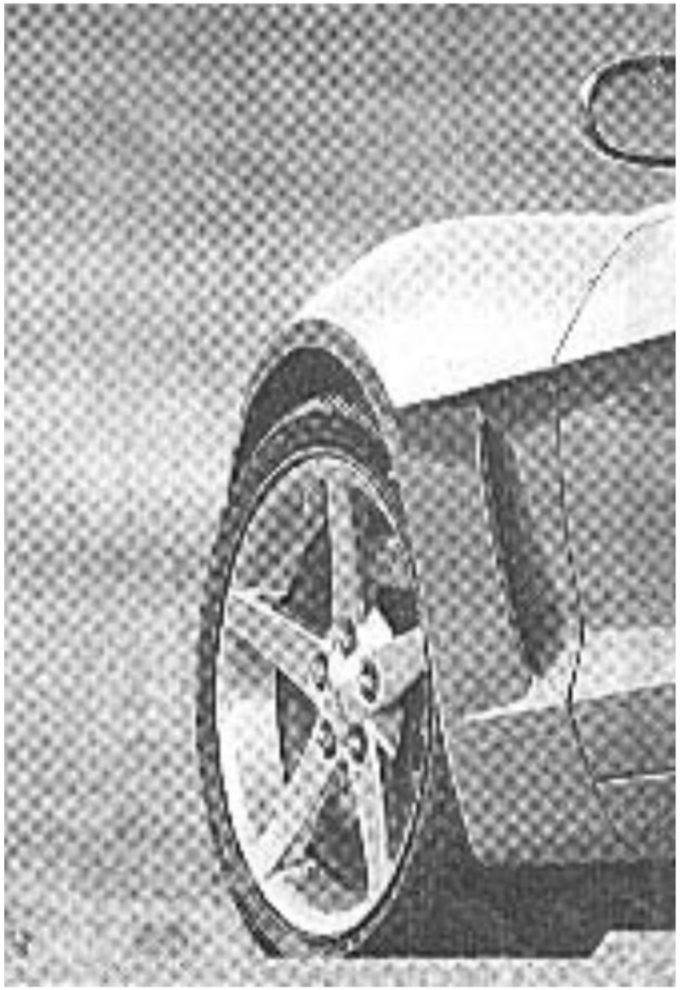


A Butterworth notch reject is applied to each burst

(and acting simultaneously on the conjugate symmetric impulses)

Università Ca' Foscari Venezia

Notch Filters



Original image

After notch filter

Università Ca' Foscari Venezia

Image restoration

The degradation process is often modeled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y).

If H is a linear, position-invariant process, then the degraded image is given in the spatial domain by:

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y)$$

Or, in frequency domain:

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

Image restoration

In the trivial case in which the noise is absent and we know perfectly the degradation function, an estimate of the original image can be obtained by inverse filtering:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

With noise the process is very unstable, especially when H(u,v) has zero or very small values

Università Ca' Foscari Venezia

Wiener filtering

The method is based on considering images and noise as **random variables**.

The objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized.

Assumptions:

- noise and the image are uncorrelated
- noise or the image has zero mean
- the intensity levels in the estimate are a linear function of the levels in the degraded image.

Wiener filtering

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

Power spectrum of the degradation function

Noise power-spectrum

Undegraded image power-spectrum

Noise to signal ratio