Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Computer Vision
Digital Filters – Frequency Domain

WS 2019/2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Content

What is the frequency domain? Motivation and intuition

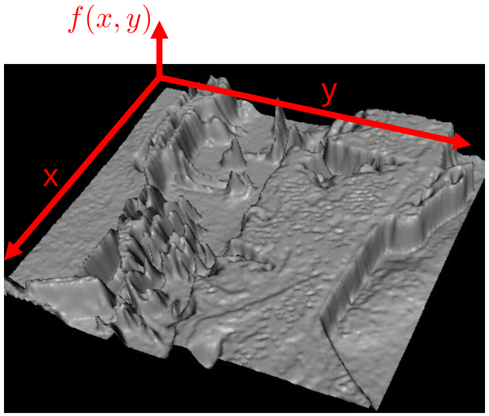
- 1D signals:
  - Discrete Fourier Transformation (DFT) and Inverse DFT
  - Convolution in the frequency domain
  - Filtering in the frequency domain
- 2D signals (images):
  - 2D Discrete Fourier Transformation (2D DFT) and Inverse
  - 2D Convolution and filtering in the frequency domain
  - Understanding a 2D spectrum
  - Lowpass, Highpass, Bandpass and Notch filters

Remember

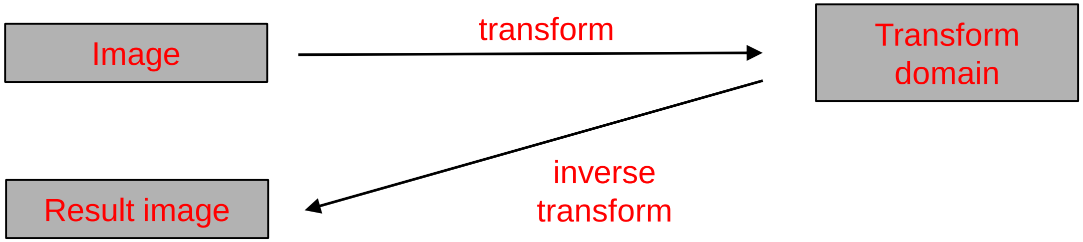
Image Processing Domains

Images can be processed in the

- Spatial domain: the image plane itself



- Transform domain: e.g. the frequency domain



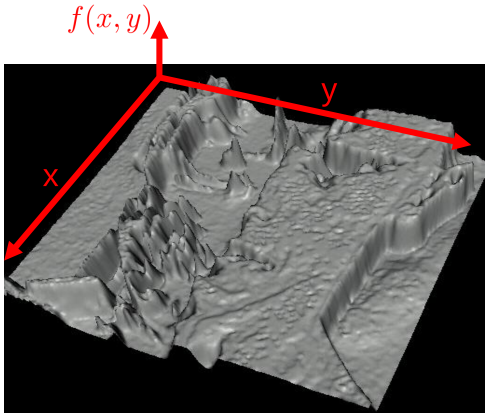
[Image: Steve Seitz]

Remember

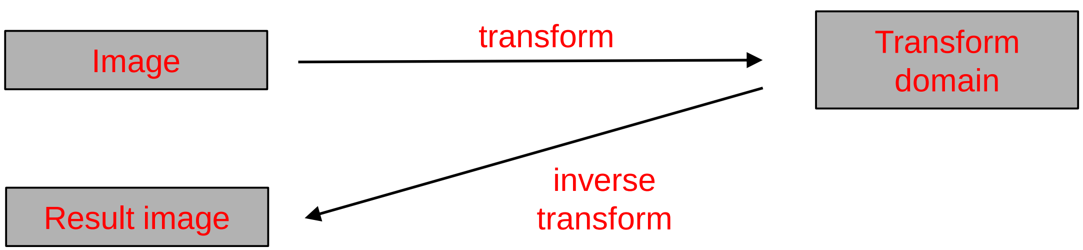
Image Processing Domains

Images can be processed in the

- Spatial domain: the image plane itself



- Transform domain: e.g. the frequency domain



Let's have a look into image processing in the frequency domain

[Image: Steve Seitz]

Why Frequency Domain?

Gonzales/Woods:

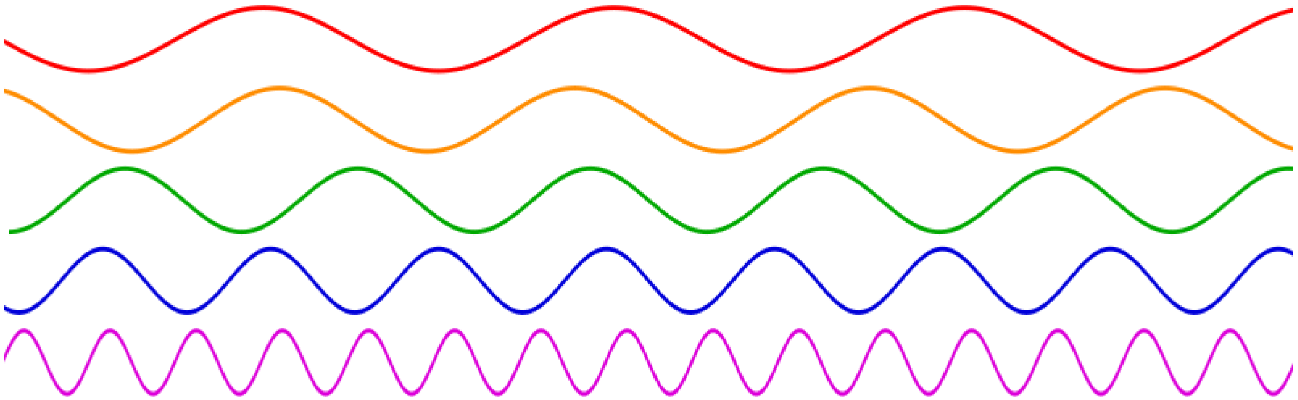
“the frequency domain can be viewed as a laboratory in which we take advantage of the correspondence between frequency content and image appearance. Some tasks that would be exceptionally difficult or impossible to formulate directly in the spatial domain, become almost trivial in the frequency domain. Once we have selected a specific filter via experimentation in the frequency domain, the actual implementation of the method usually is done in the spatial domain.”

Remember

Frequencies

- What are frequencies?
- Frequency: The number of occurrences of a repeating event per unit time (temporal frequency)

Low frequency



High frequency

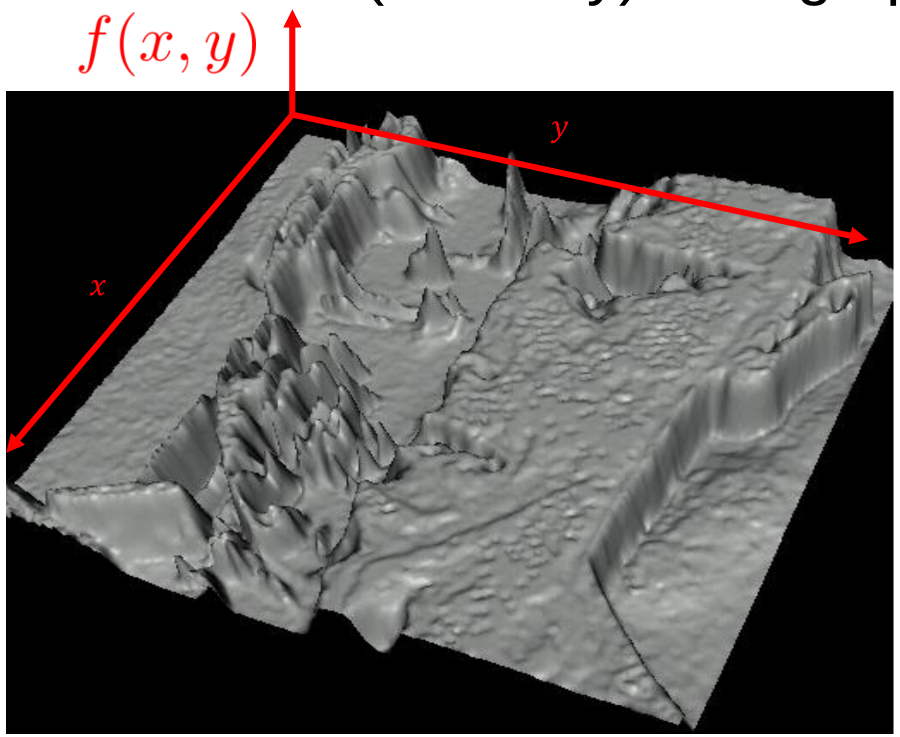
But: there is no time in images! ???

[Wikipedia: Frequency]

Remember

Frequencies

- **Spatial frequencies**: replace time axis by spatial displacement axes
- In images: the amount of (intensity) change per unit



[Image: Steve Seitz]

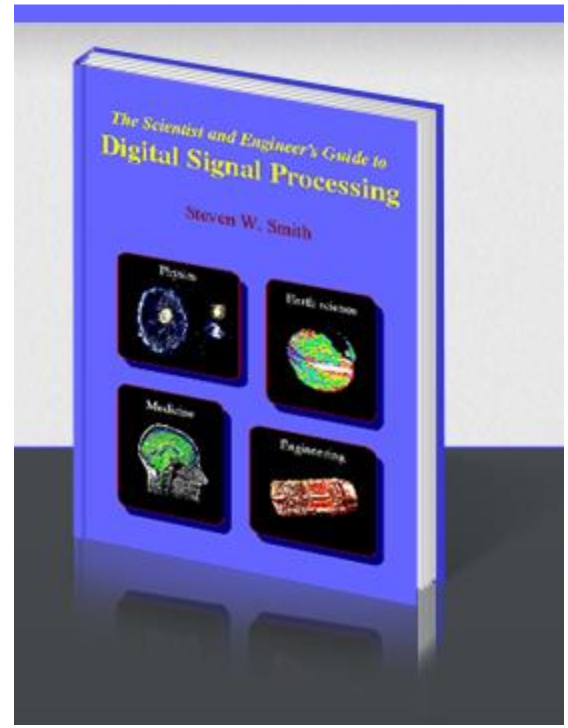
Literature

- Chapter 4 from Gonzalez/Woods

If the DFT and frequency domain are completely new to you, read:

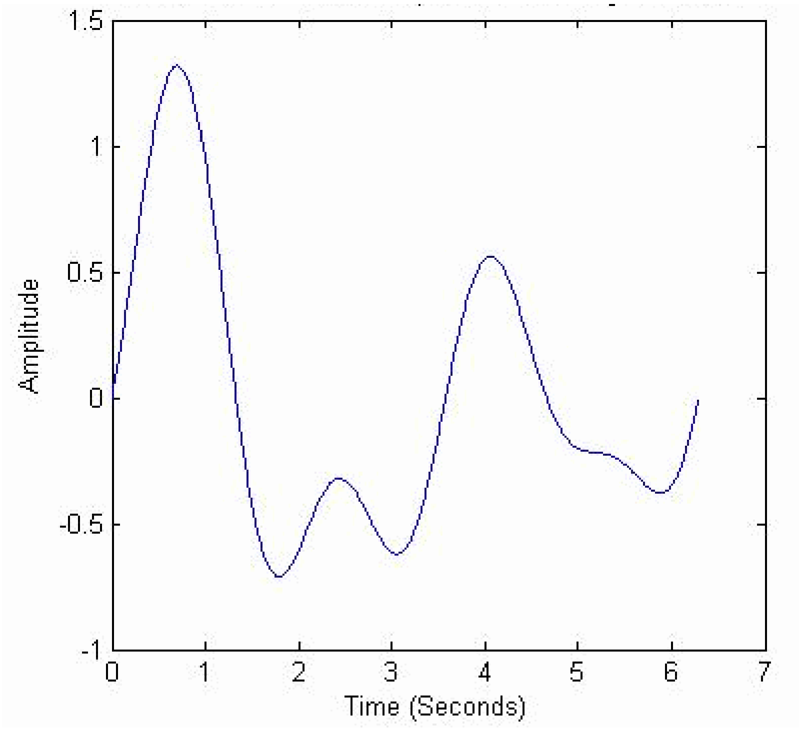
- “The Scientist and Engineer's Guide to Digital Signal Processing”, copyright ©1997-1998 by Steven W. Smith.

For more information visit the book's website at: www.DSPguide.com



Fourier Transform

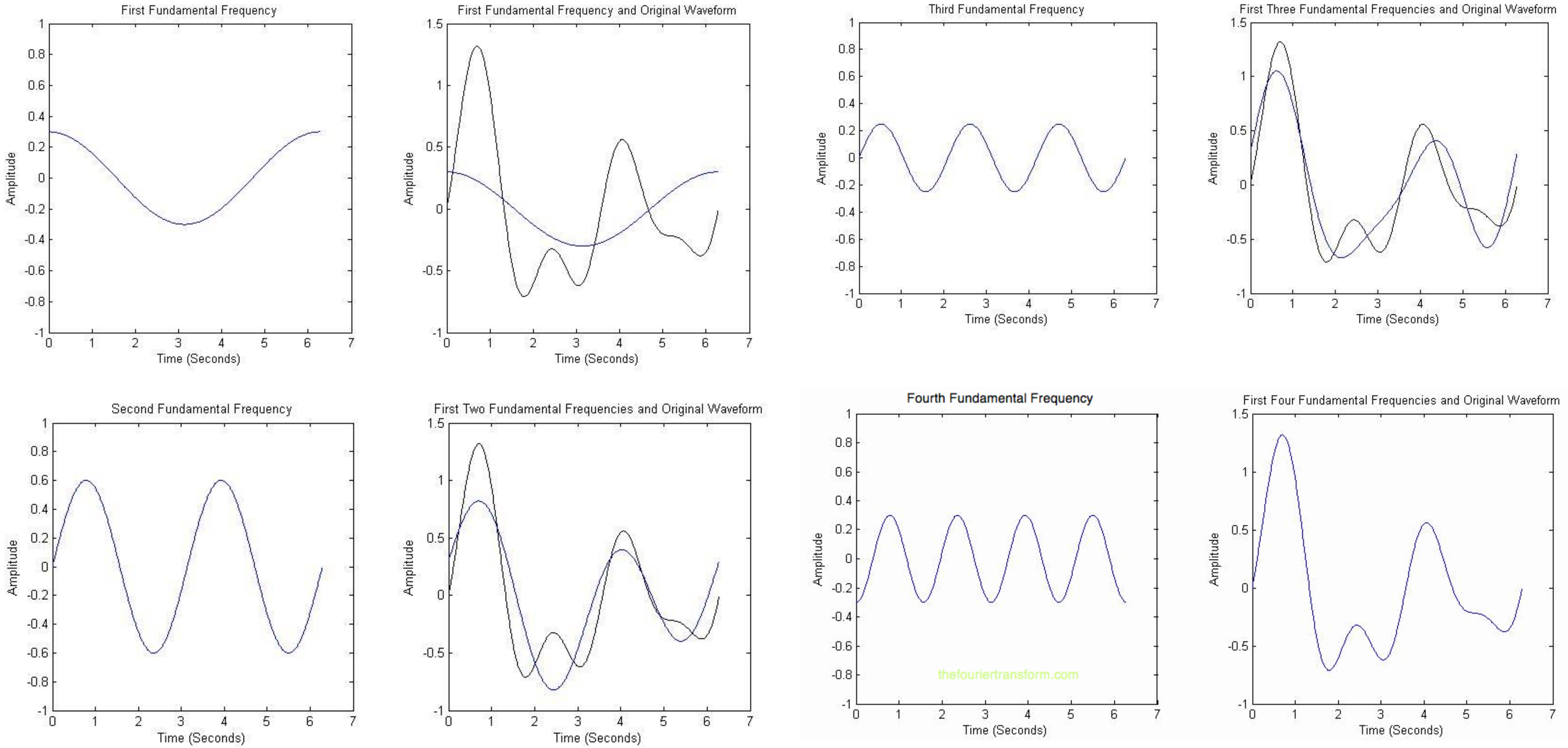
Fourier (1807): any continuous periodic signal can be represented as the sum of sinusoids:



Let’s take this signal and see how we can compose it step by step by sinusoids

[www.thefouriertransform.com]

Fourier Transform



[www.thefouriertransform.com]

Fourier Transform

$$f(t) = \underbrace{5}_{\text{dc}} + \underbrace{2\cos(2\pi t - 90^\circ)}_{\text{1Hz}} + \underbrace{3\cos 4\pi t}_{\text{2Hz}}$$

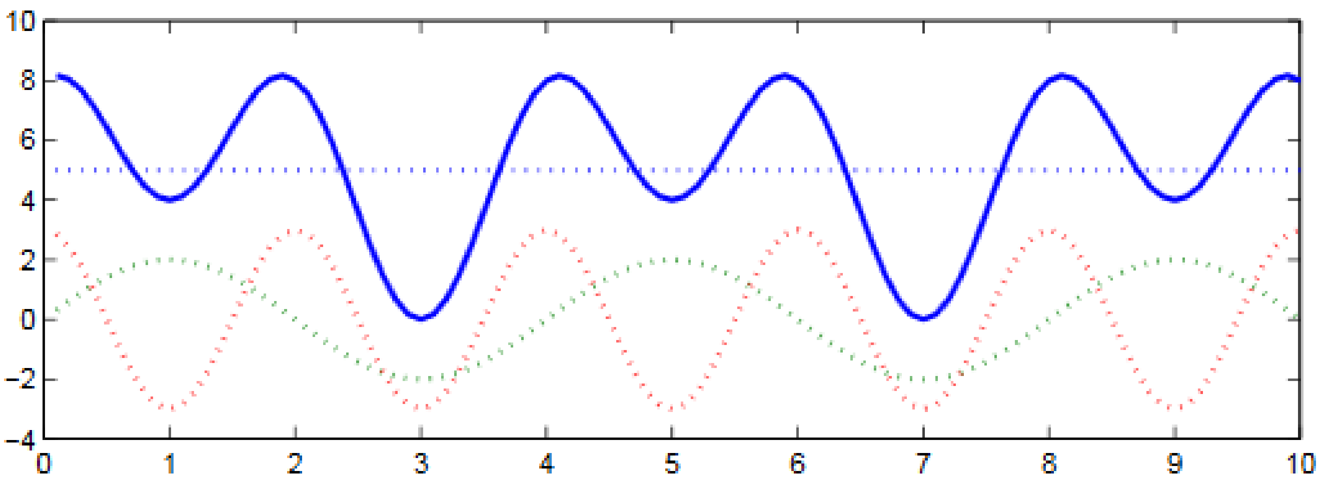


Figure 7.2: *Example signal for DFT.*

[Roberts]

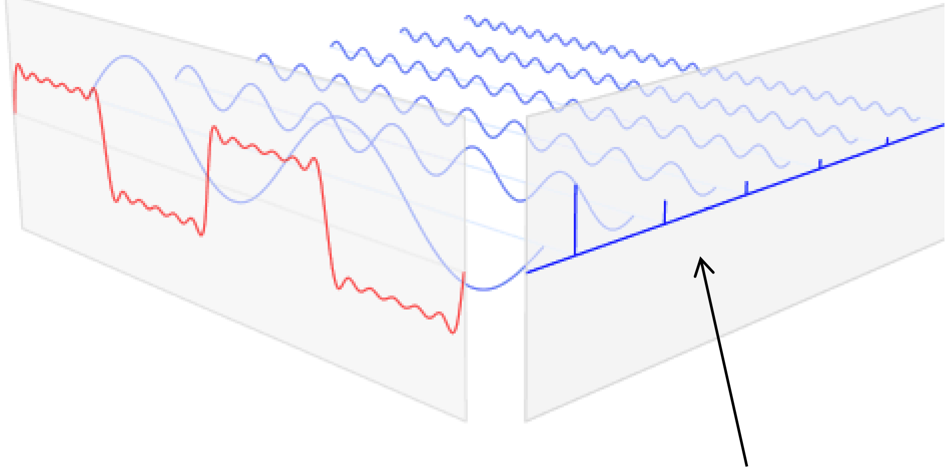
Fourier Transform

- Fourier transform:
  decompose a signal into its frequencies

[Wikipedia: Fourier Transform]

Fourier Transform

- Fourier transform:
  decompose a signal into its frequencies
- Input:
  signal
- Output:
  frequency spectrum



Sinusoids (basis functions) scaled with different amplitudes

Fourier Transforms

- Fourier analysis: a family of mathematical techniques, based on decomposing signals into sinusoids

Require infinite number of sinusoids ☹

DFT

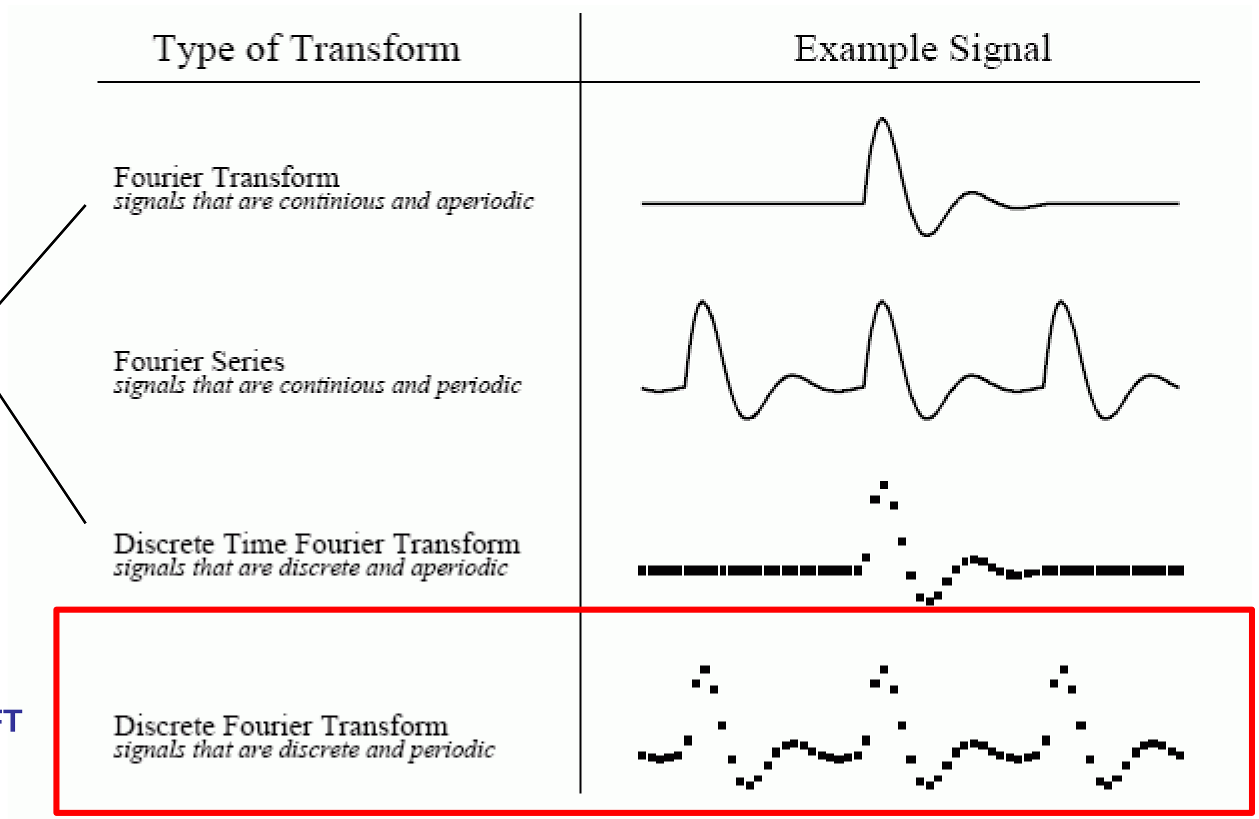
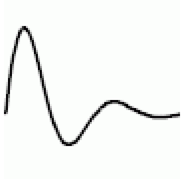


FIGURE 8-2
Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

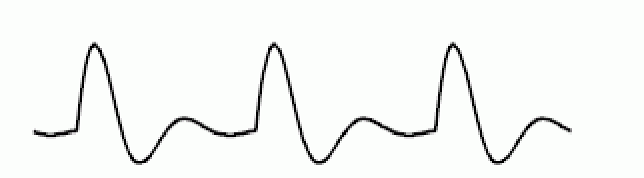
This is the one we are interested in

Periodic signals?

- Most real-world signals are not periodic, are they?



- Just imagine the signal is periodic by repeating an infinite number of samples on left and right



- Now, we have a periodic signal! ☺
- And images?

Periodic images?

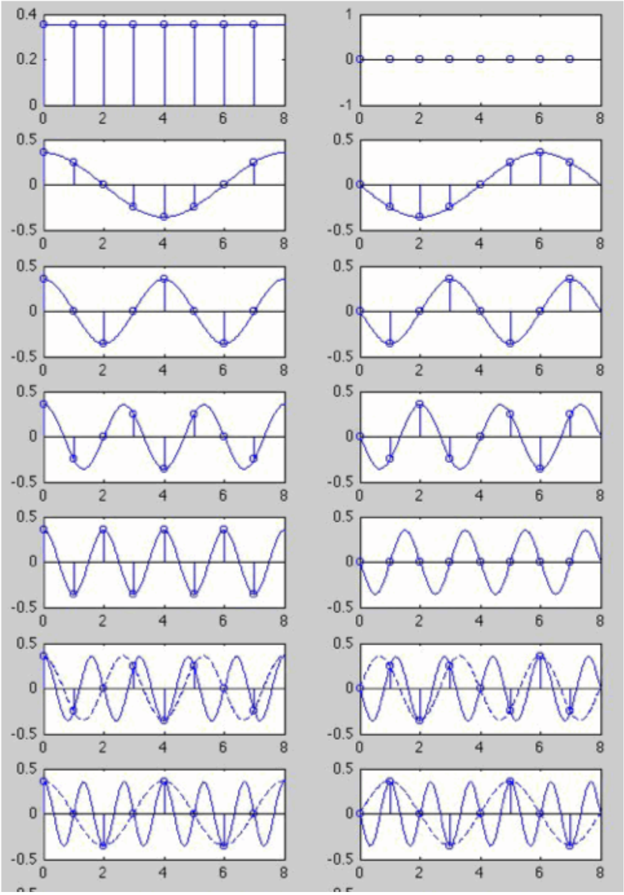
Basis Functions

Basis functions: Sinusoids

- For the real DFT: sine + cosine waves
- For the Discrete Cosine Transform: only cosine waves
- For the Complex DFT: complex sinusoids
- For 1D signals: 1D Sinusoids
- For 2D signals (images): 2D Sinusoids

Basis Functions

- The first 7 cosine and sine basis functions (solid lines):

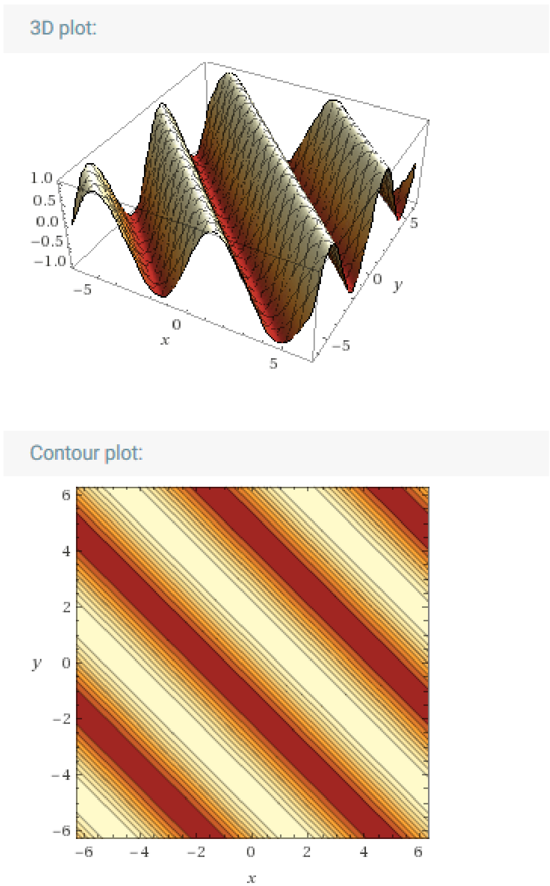


http://fourier.eng.hmc.edu/e59/lectures/e59/node21.html

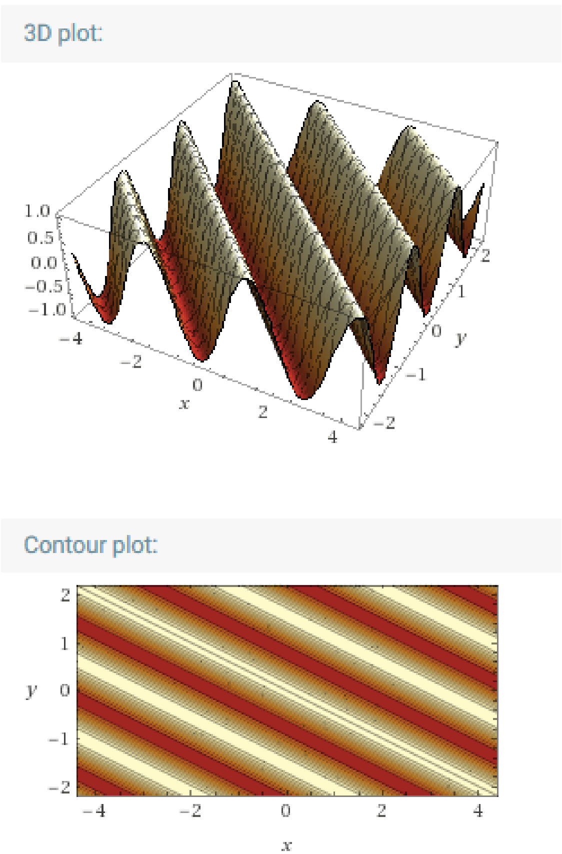
Basis Functions

2D sine and cosine functions:

sin(x+y):

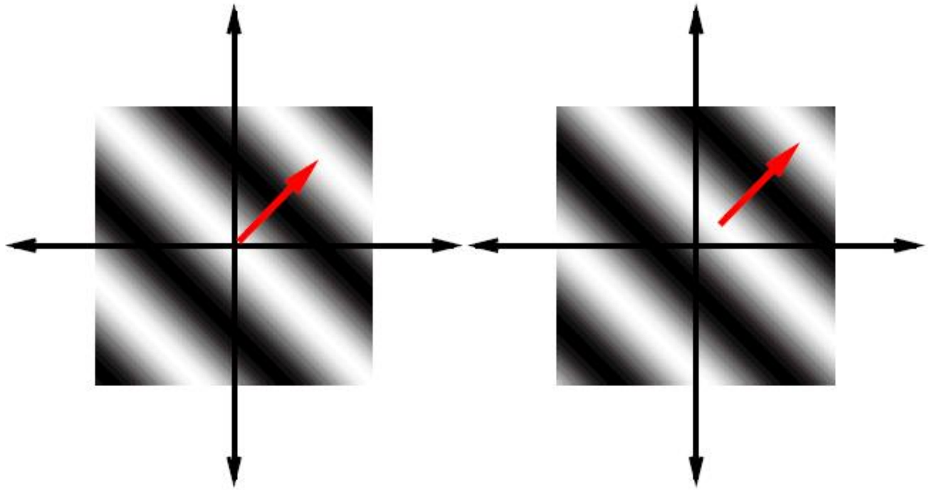


cos(2x+4y)



Basis Functions

2D sine and cosine functions (from top):

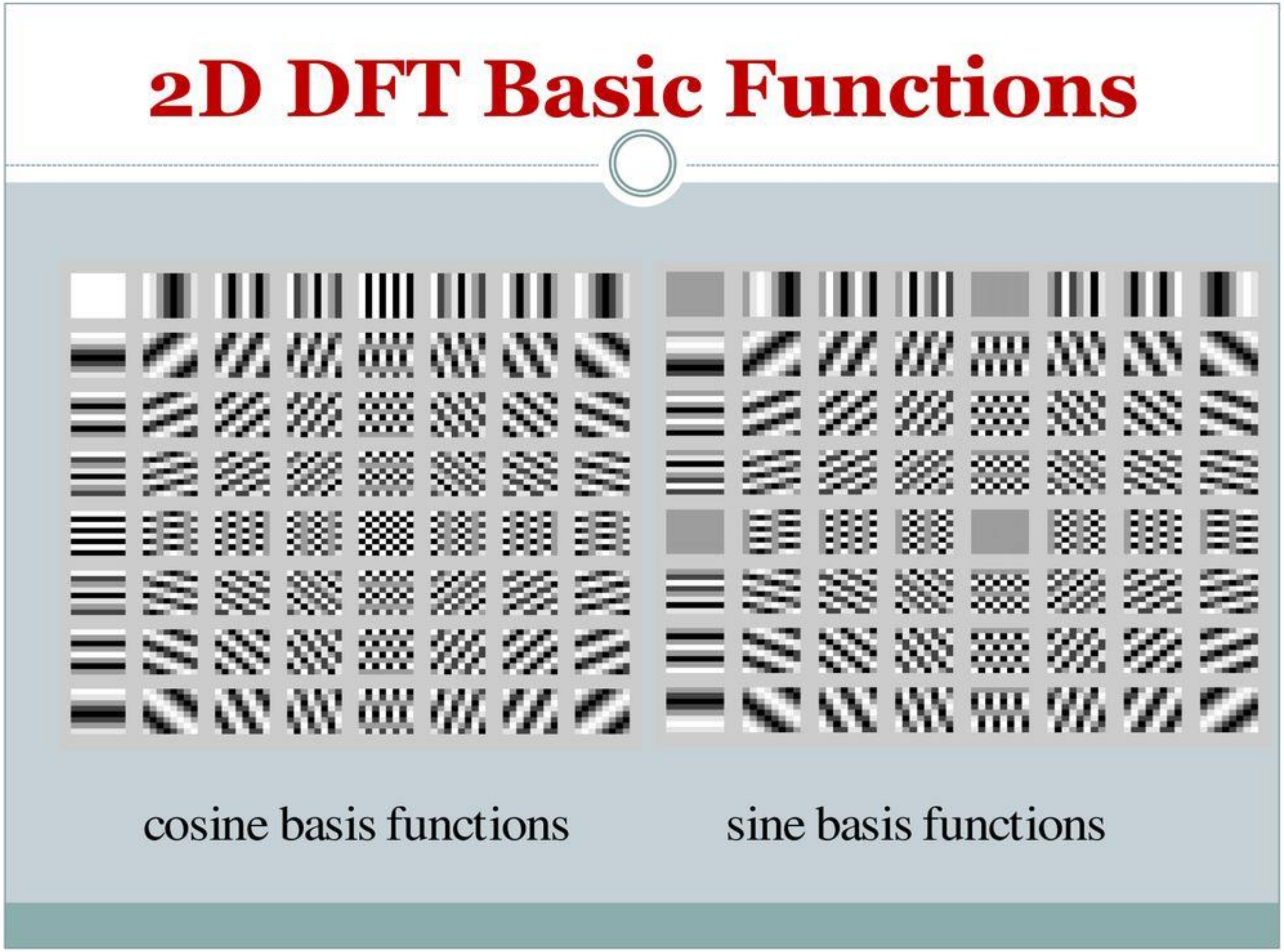


even (cosine)

odd (sine)

[Images: Movellan, 2008]

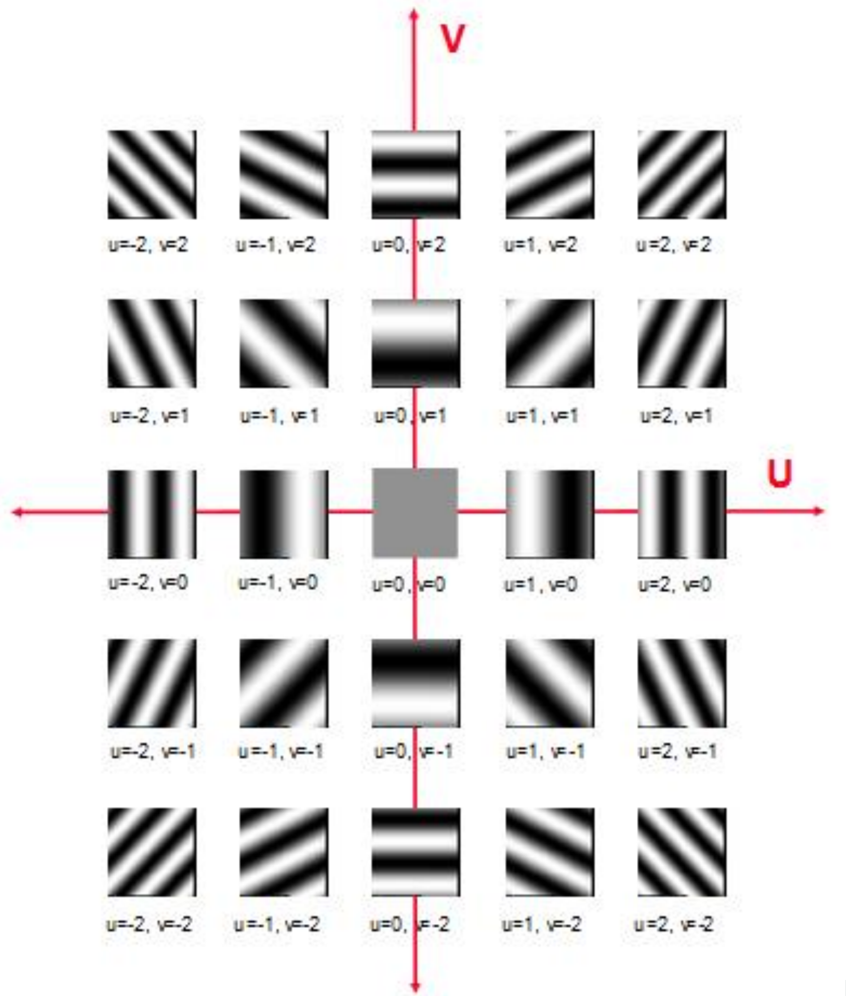
2D Basis Functions



https://slideplayer.com/slide/12911903/78/images/17/2D+DFT+Basic+Functions.jpg

2D Basis Functions

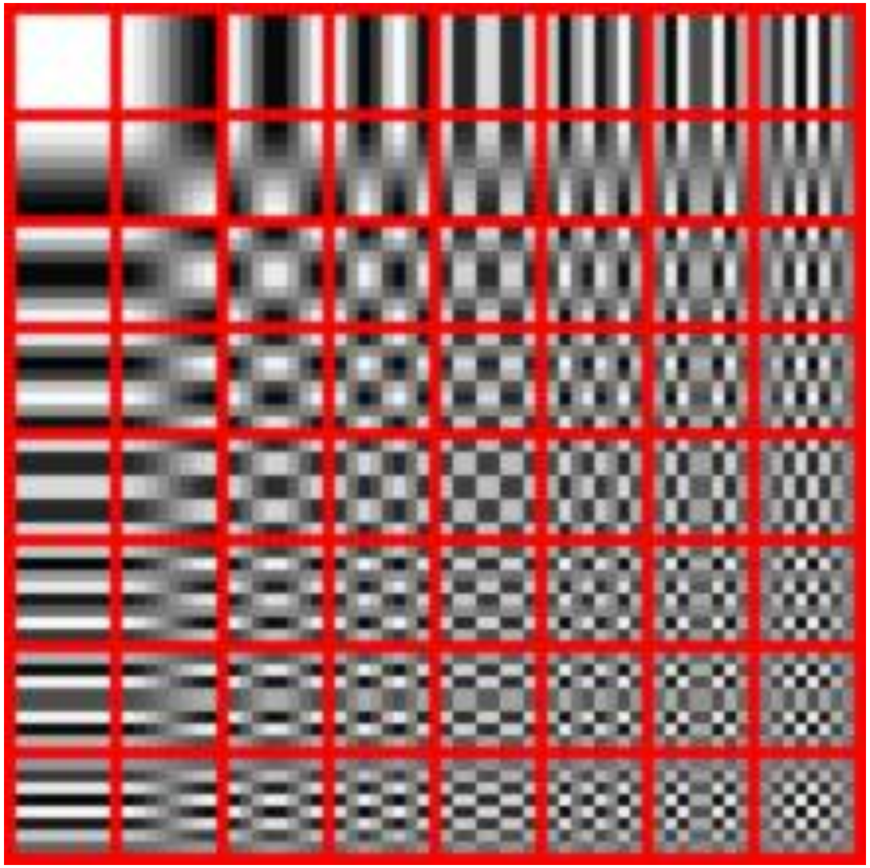
After shifting the coordinate system:



Discrete Cosine Transform

The discrete cosine transform has only cosines as basis functions:

Lowest frequency



Highest frequency

[Wikipedia: Discrete Cosine Transform]

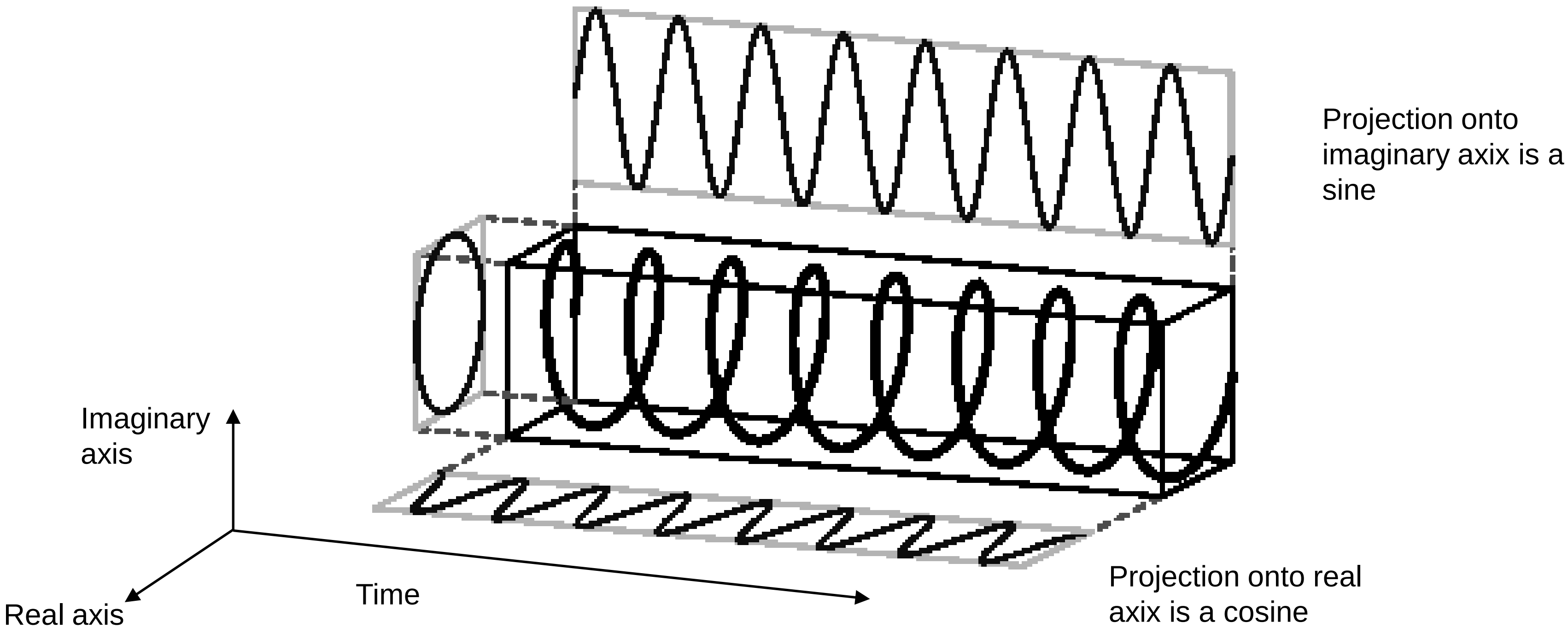
Complex Sinusoid

- The sine and cosine waves can be combined into **complex sinusoids**, using the Euler identity:

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

- Complex sinusoids are the basis functions in the complex DFT
- More details: https://www.dsprelated.com/freebooks/mdft/Complex_Sinusoids.html

A complex sinusoid



[http://www.technick.net/public/code/cp_dpage.php?aiocp_dp=guide_dft_projection_circular_motion]

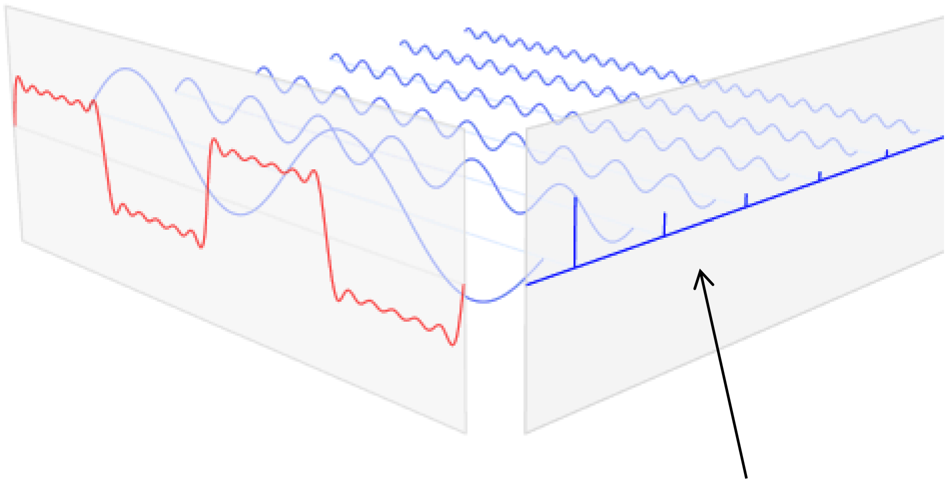
Fourier Transform

Remember

- Fourier transform:
  decompose a signal into its frequencies

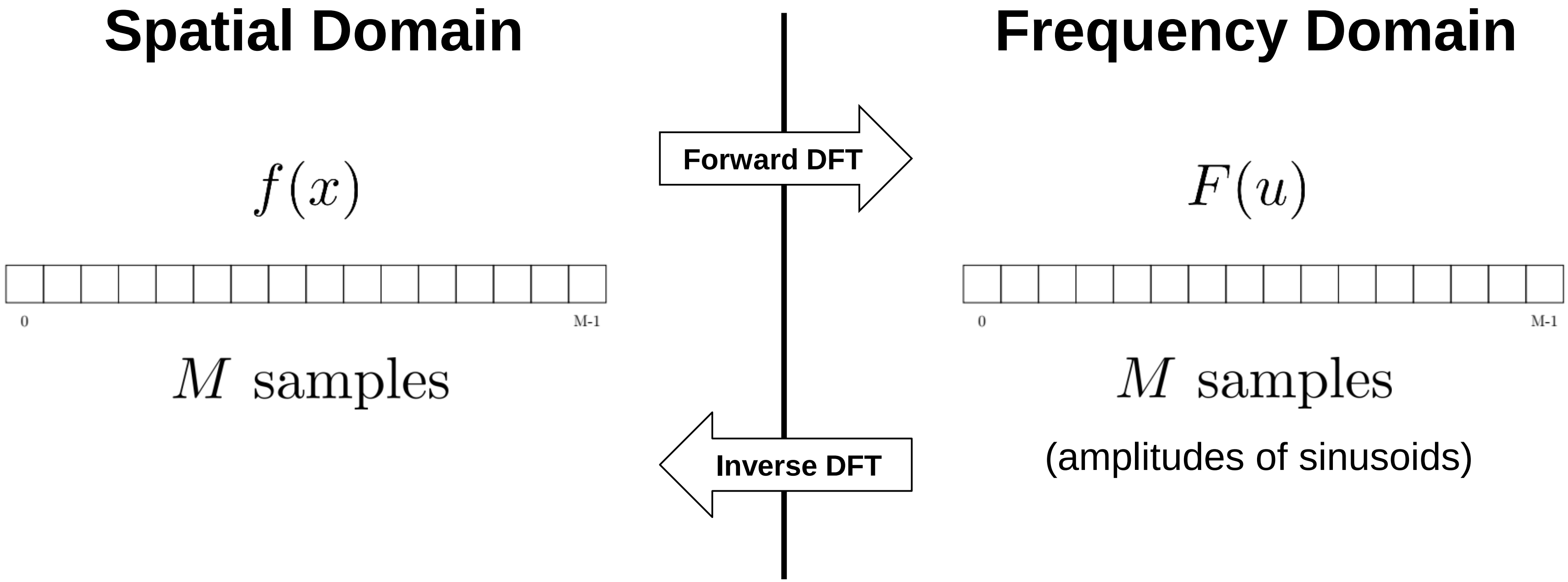
- Input:
  signal

- Output:
  frequency spectrum



Sinusoids (basis functions) scaled with different amplitudes

The Discrete Fourier Transform



In the “frequency domain” a signal is represented by its frequencies, given by the amplitudes of the basis functions (sinusoids)

The Discrete Fourier Transform

Spatial Domain

$f(x)$

M samples

Forward DFT

Inverse DFT

Frequency Domain

$F(u)$

M samples

(amplitudes of sinusoids)

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

DFT:
We have: a signal f of length M
We want: the amplitudes of the basis functions (F)

The Discrete Fourier Transform

The DFT:

signal

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

Basis functions

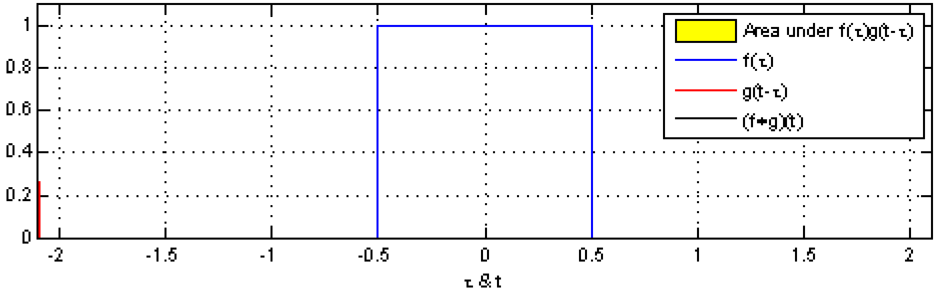
- Which operation is this?
- Hint: it is a sum of products
- It is correlation of the signal with its basis functions

DFT:
We have: a signal f of length M
We want: the amplitudes of the basis functions (F)
Use (Cross-)Correlation or Fast Fourier Transform (FFT)

Correlation

Remember

- More general:
- Correlation of two signals
  - *multiplies its elements* and
  - *sums up the results* (sliding dot product).
- The result is a measure of how similar the two signals are. It is commonly used to search patterns in signals (e.g. images).
- This will be important for the Fourier analysis of signals, and for template matching of image patches.



[Wikipedia: Convolution]

The Discrete Fourier Transform

Spatial Domain

$f(x)$

0 M-1

M samples

Forward DFT

Inverse DFT

Frequency Domain

$F(u)$

0 M-1

M samples

(amplitudes of sinusoids)

Inverse DFT:
We have: the basis functions and their amplitudes (F)
We want: the original signal f

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

Inverse DFT

The inverse DFT:

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

Basis functions

Amplitudes

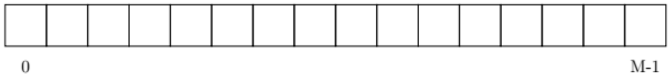
The inverse DFT simply adds all scaled basis functions

Inverse DFT:
We have: the basis functions and their amplitudes (F)
We want: the original signal f

The Discrete Fourier Transform

Spatial Domain

$f(x)$



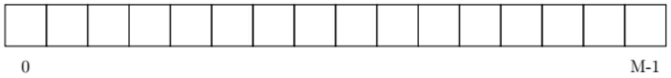
M samples

Forward DFT

Inverse DFT

Frequency Domain

$F(u)$



M samples

(amplitudes of sinusoids)

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

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

Fast Convolution

Convolution theorem:

Convolution in the time/spatial domain equals point-wise multiplication in the frequency domain:

$$f(t) * g(t) \longleftrightarrow F(\omega) \cdot G(\omega)$$

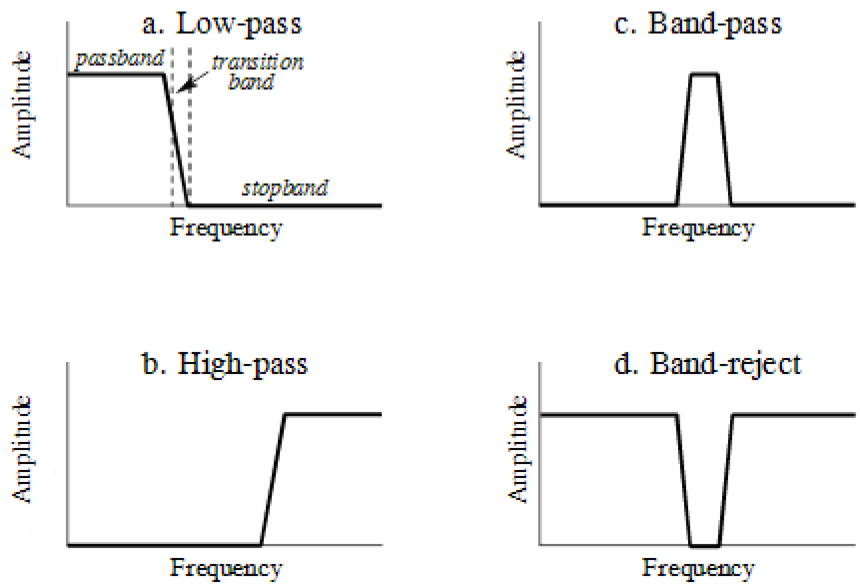
Time/spatial domain Frequency domain

$\Rightarrow$ Convolution is much faster in the frequency domain.

$\Rightarrow$ But: FFT takes time itself! So, this only pays off for large filters.

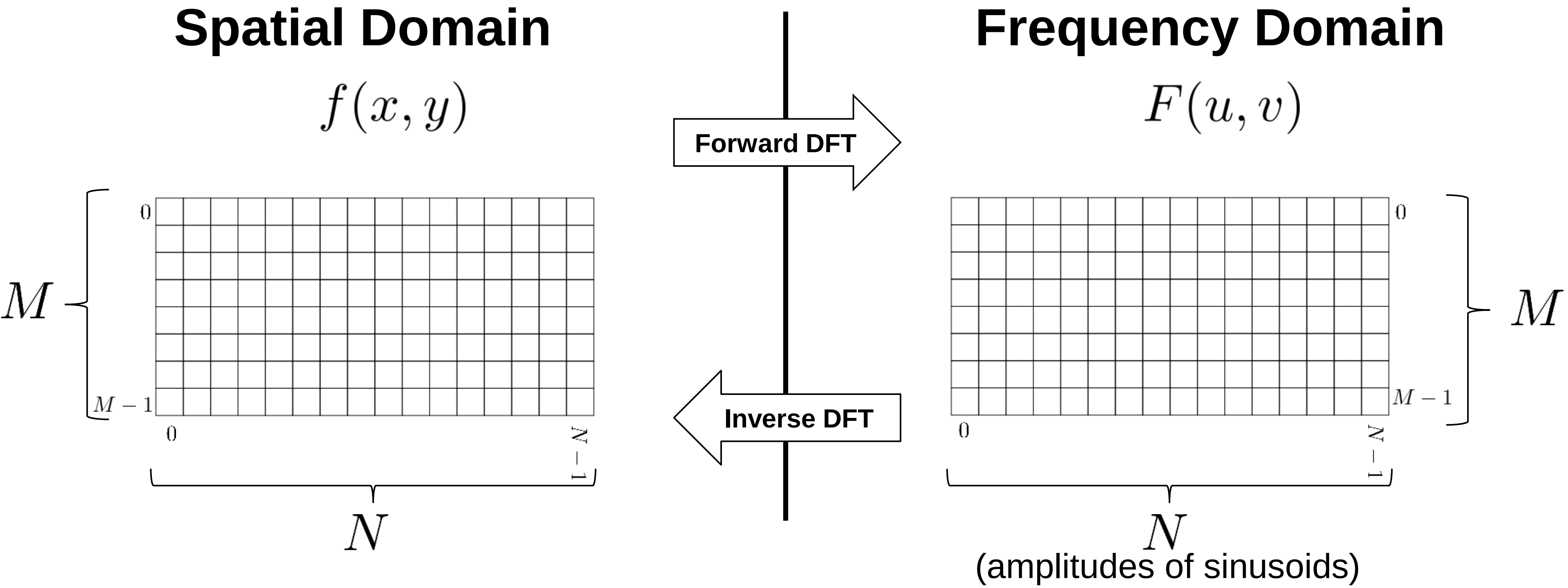
Digital Filters in Frequency Domain

- Some operations are easier and more intuitive in the frequency domain (e.g. if you want to design filters that deal with special frequencies)
- Examples:



[Smith 1997]

2D DFT/IDFT



A 2D signal (image f(x,y)) is transformed into a 2D spectrum F(u,v)

2D DFT

DFT:
We have: a signal f
We want: the amplitudes of the basis functions (F)
Use (Cross-)Correlation or Fast Fourier Transform (FFT)

- 1D DFT:

signal

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

Basis functions

- 2D DFT:

signal

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

Basis functions

2D Inverse DFT

Inverse DFT:
We have: the basis functions and their amplitudes (F)
We want: the original signal f

1D inverse DFT:

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

Basis functions

Amplitudes

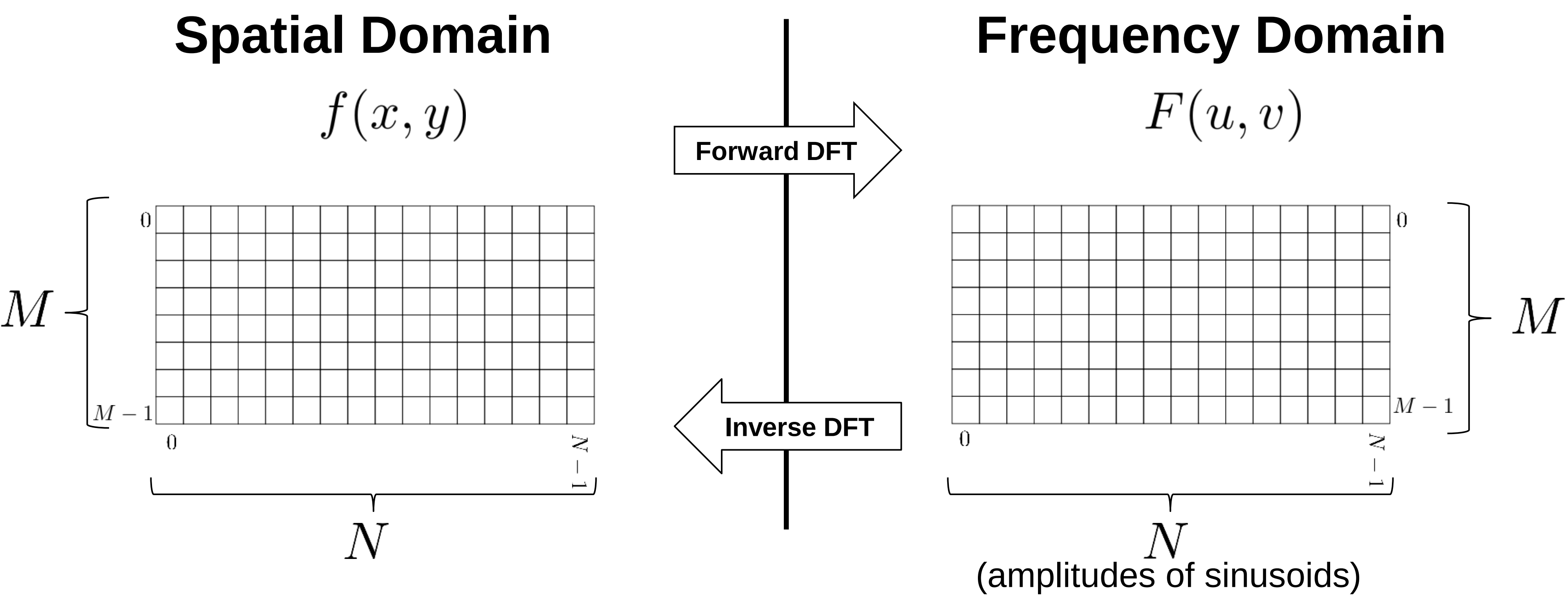
2D inverse DFT:

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

Basis functions

Amplitudes

2D DFT/IDFT



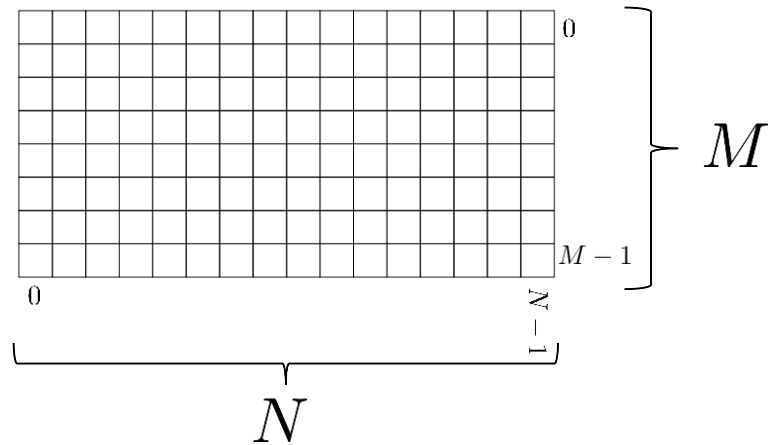
A 2D signal (image f(x,y)) is transformed into a 2D spectrum F(u,v)

2D DFT/IDFT

Let us now look closer at the representation F(u,v) in the frequency domain

Frequency Domain

$$F(u,v)$$



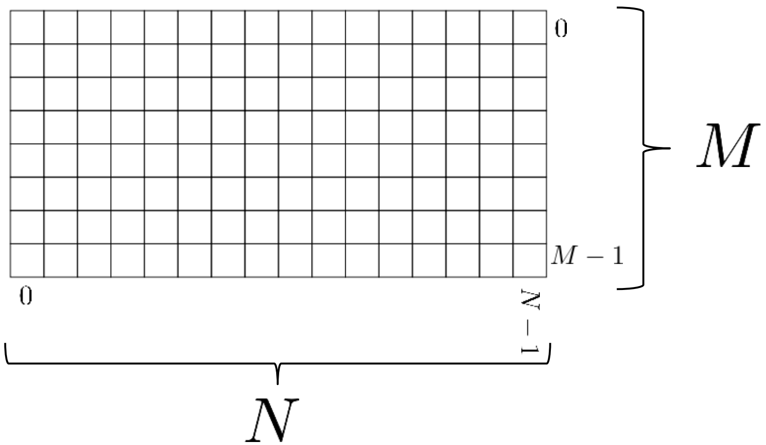
(amplitudes of sinusoids)

2D DFT/IDFT

- What does a value F(u,v) express?
- How prominent the corresponding basis function is in the image f(x,y)

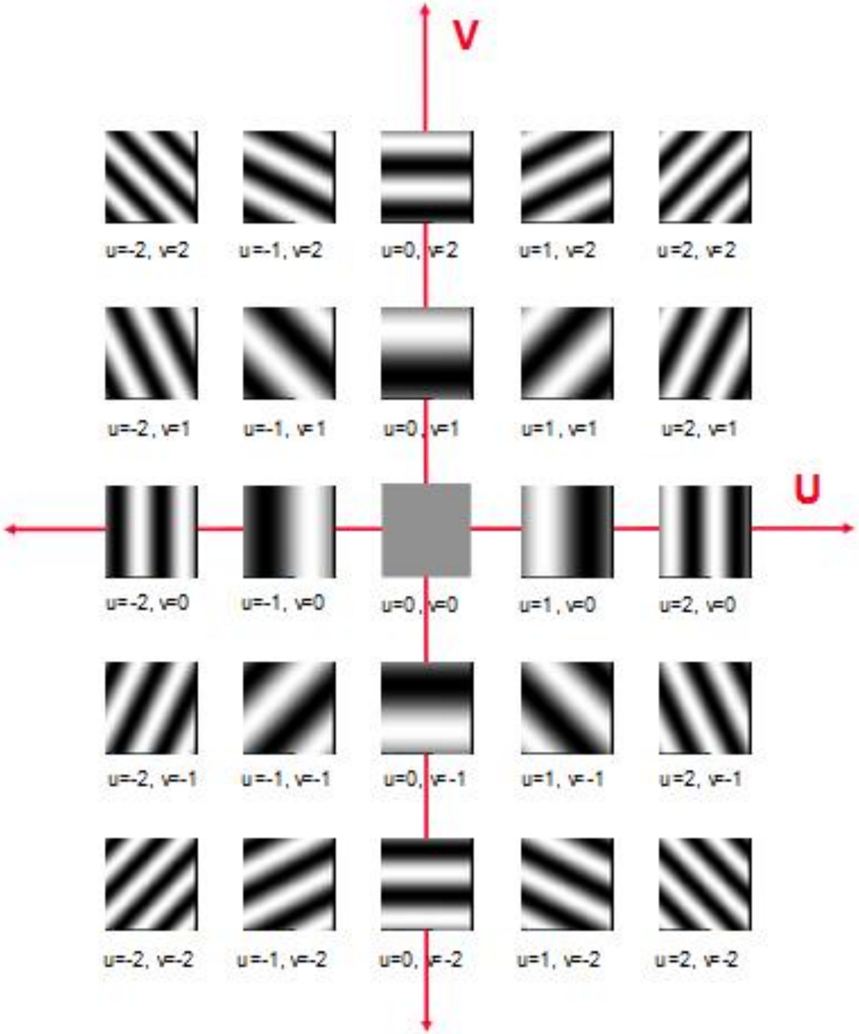
Frequency Domain

$F(u,v)$



(amplitudes of sinusoids)

2D Basis Functions



We can reproduce any image by overlaying (adding) these scaled basis functions

Spectrum and Phase

- Note: input and output of the DFT are complex numbers, i.e. the Fourier transform of f(x) can be expressed as:

$$F(u,v) = R(u,v) + jI(u,v)$$

- Remember: complex numbers can be expressed in polar form:

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

- with the magnitude (frequency spectrum/Fourier spectrum):

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

- and the phase (phase angle/phase spectrum):

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

[Gonzalez/Woods]

Spectrum and Phase

- Magnitude (frequency spectrum/Fourier spectrum):

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

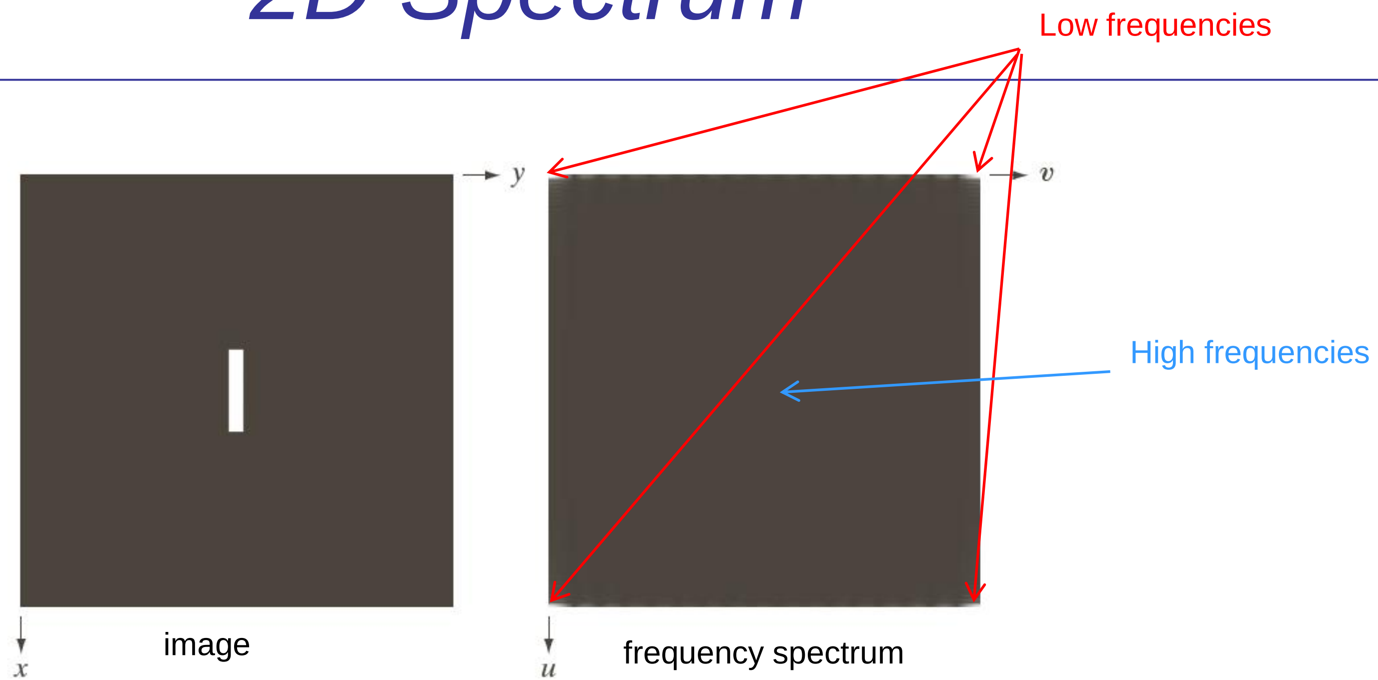
- Phase (phase angle/phase spectrum):

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

- Both, frequency spectrum and phase spectrum, can be visualized as images
- Let us start with the frequency spectrum…

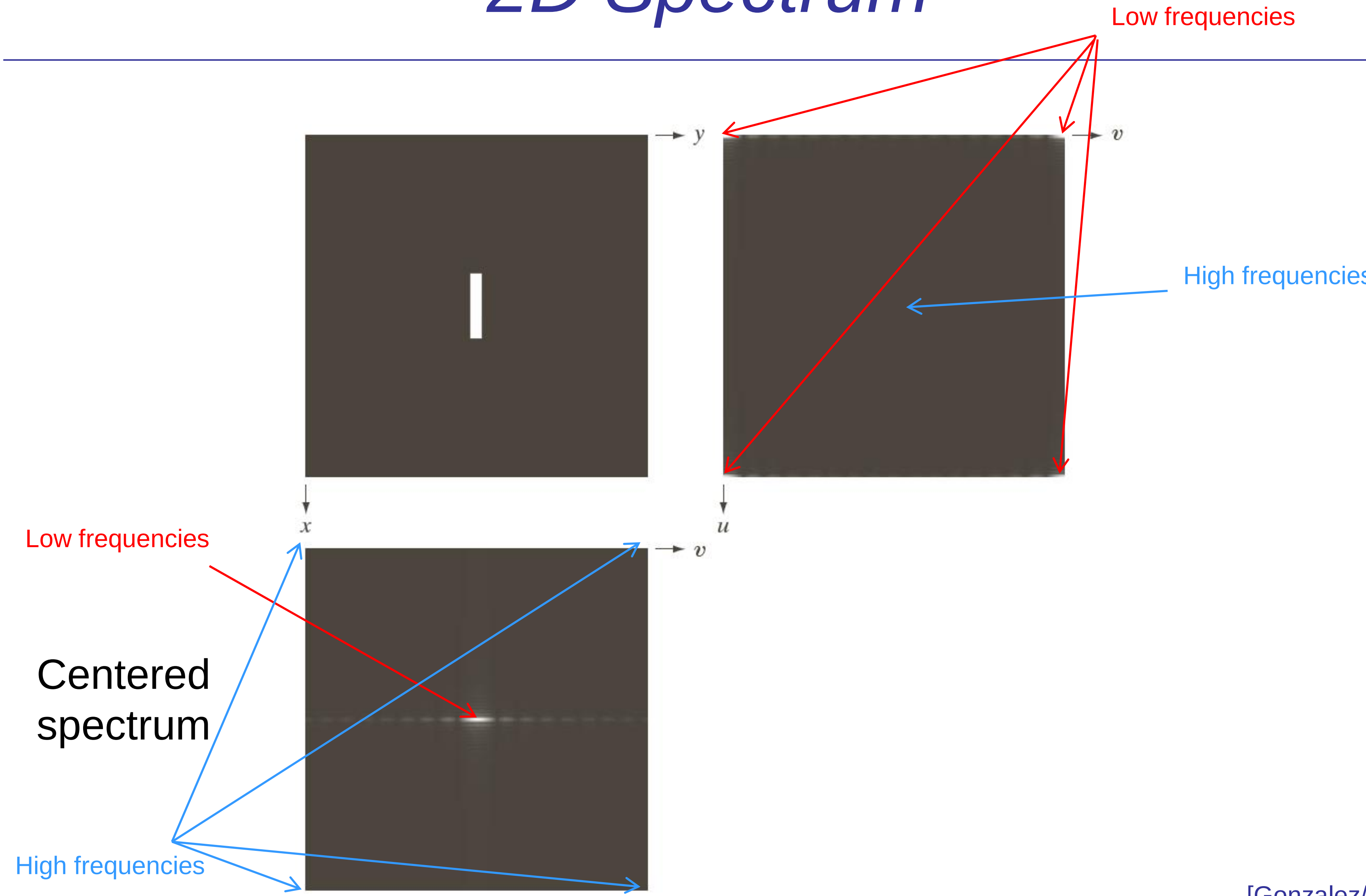
[Gonzalez/Woods]

2D Spectrum



[Gonzalez/Woods]

2D Spectrum

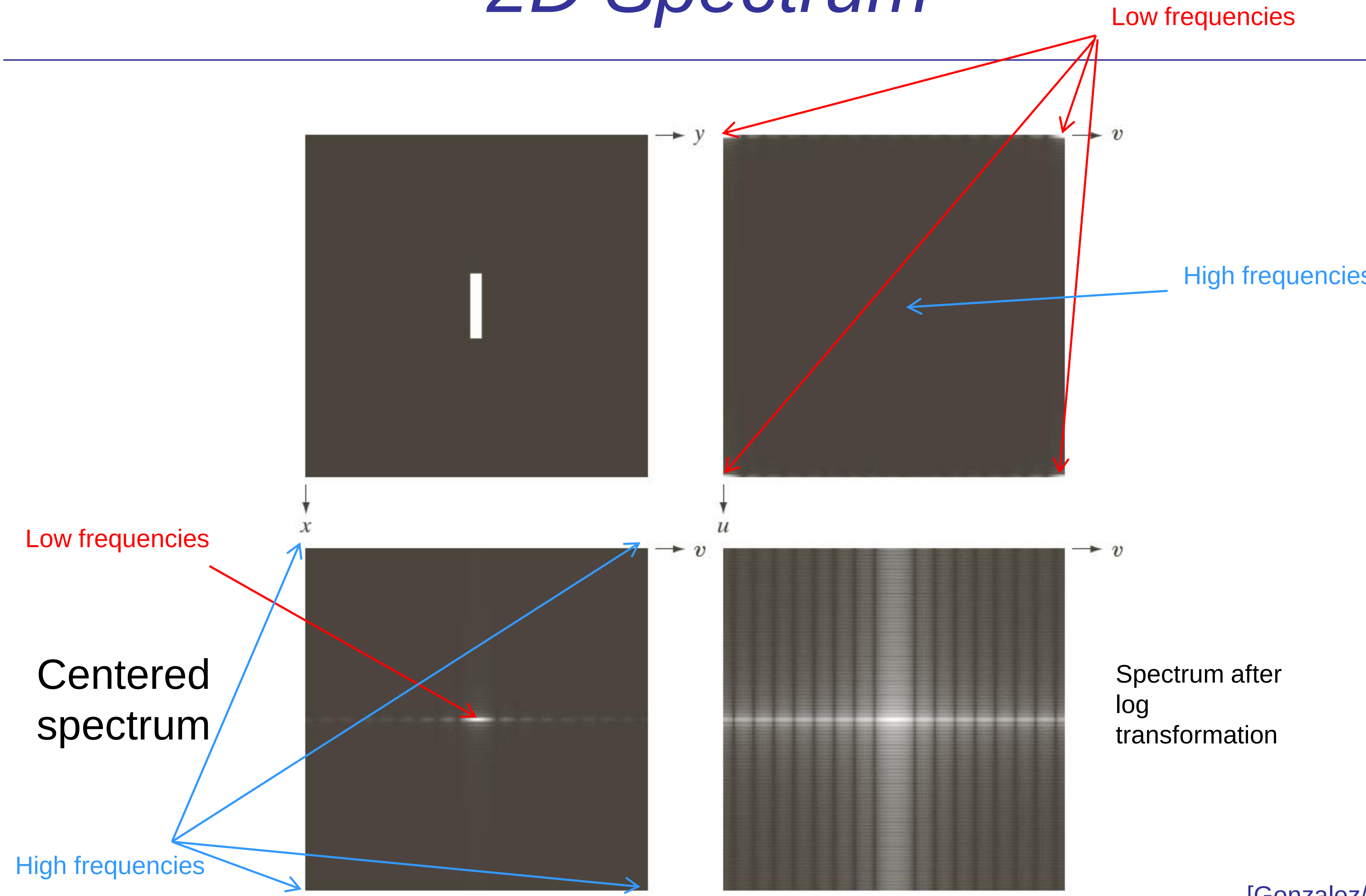


Centered spectrum

[Gonzalez/Woods]

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

2D Spectrum



Centered spectrum

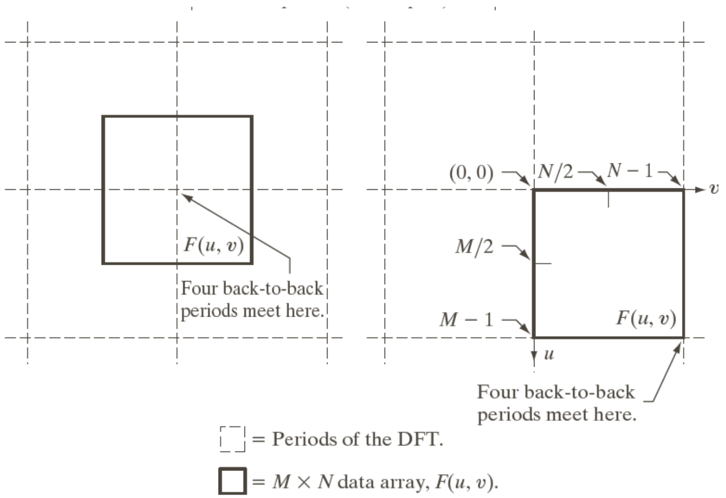
Spectrum after log transformation

[Gonzalez/Woods]

Result of DFT

We always look at the centered spectrum for better visibility:

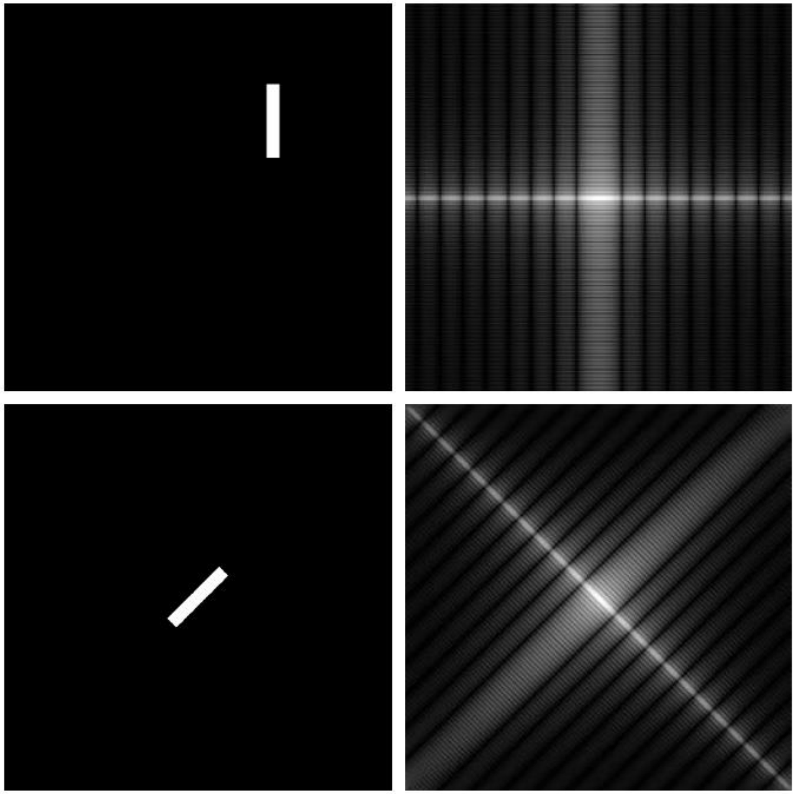
Result of DFT: Centered (shifted) DFT:



The spectrum is even symmetric about the origin: $|F(u,v)| = |F(-u,-v)|$

Translations/Rotations

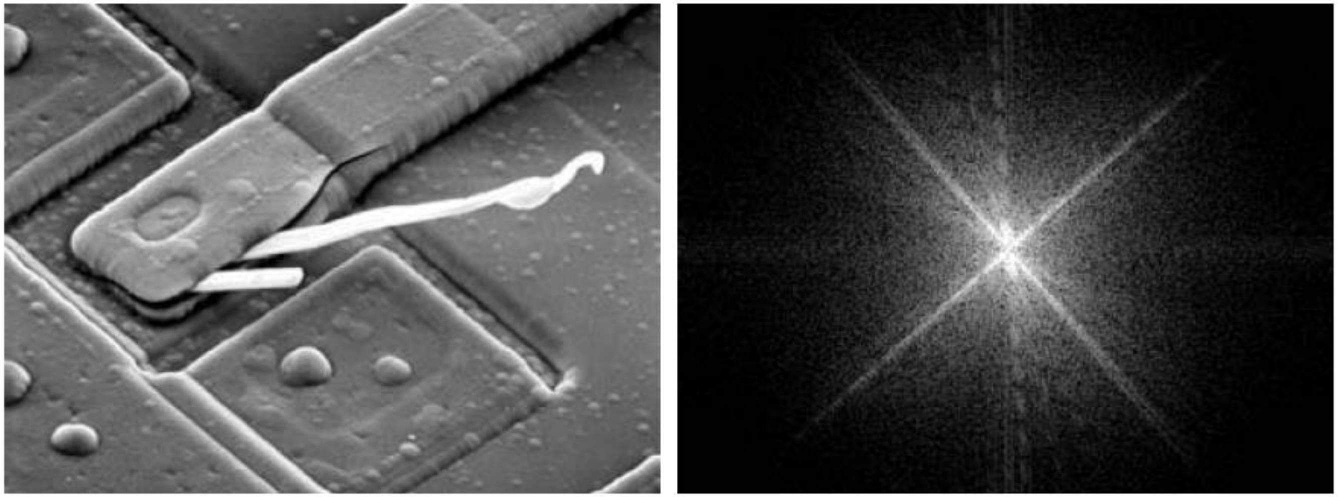
The frequency spectrum is insensitive to translation, but sensitive to rotations:



[Gonzalez/Woods]

Spectrum

- Another example of a spectrum



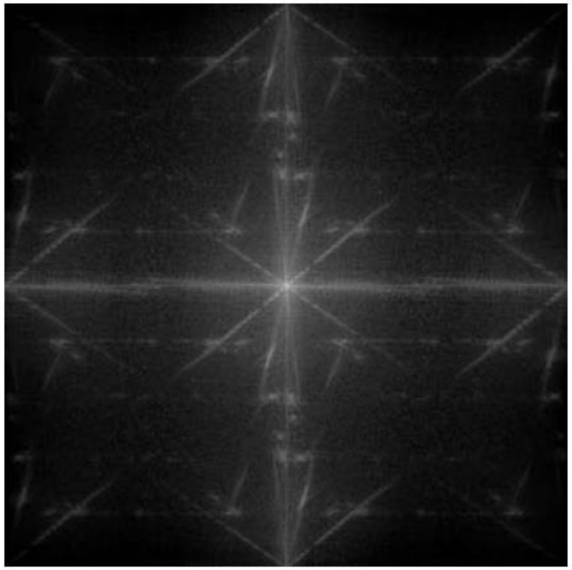
a b

FIGURE 4.29 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

[Gonzalez/Woods]

Spectrum

- Another example of a spectrum

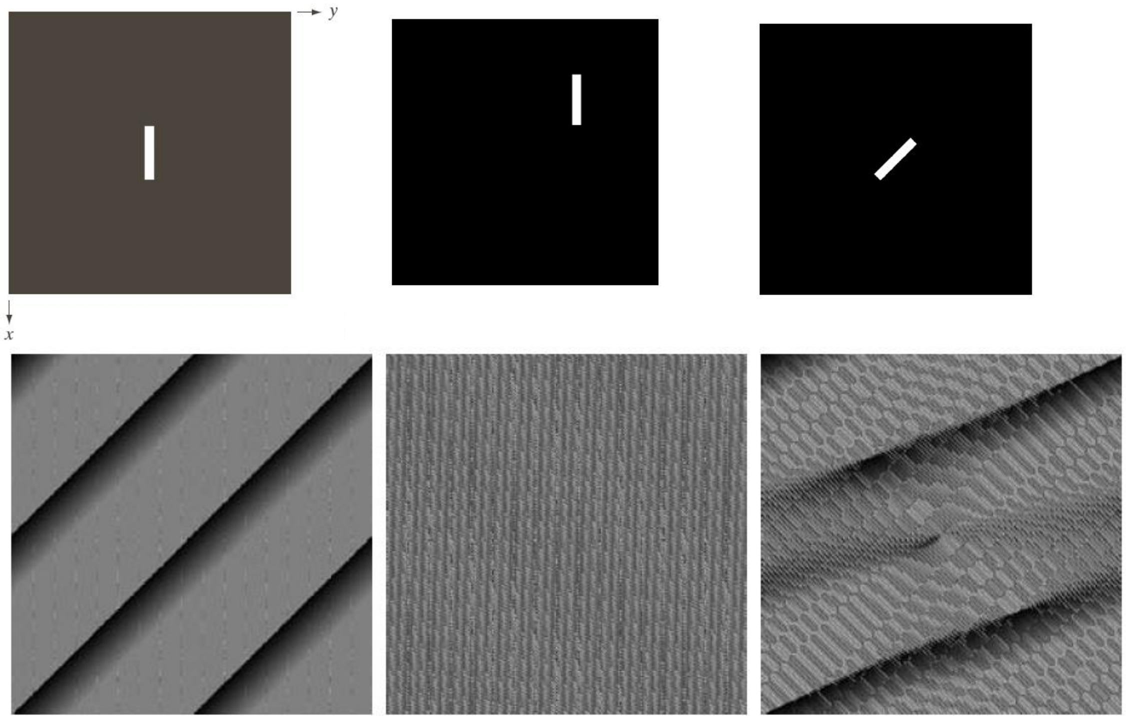
[Gonzalez/Woods]

Phase

The phase is sensitive to translation and rotation:

images

phase



Note that the phase angles are completely different (and not especially intuitive)

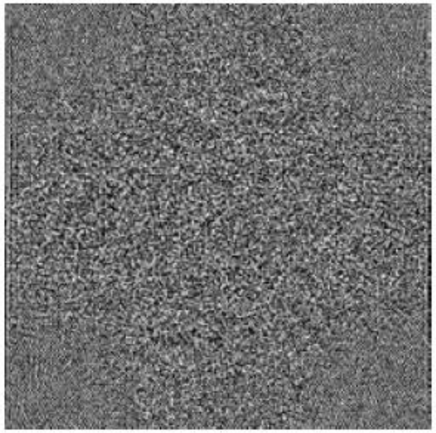
[Gonzalez/Woods]

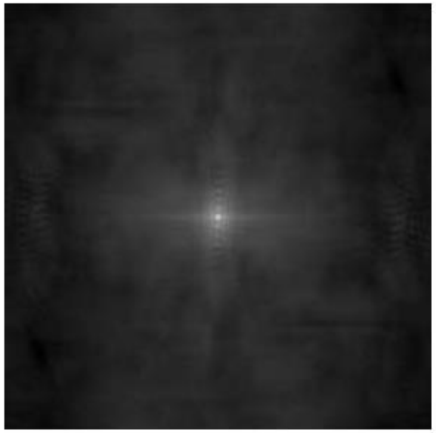
Spectrum and Phase

Phase angle

IDFT of only
phase angle

IDFT of only
spectrum

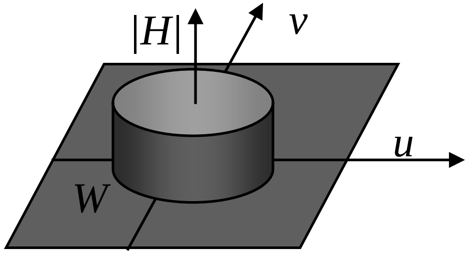


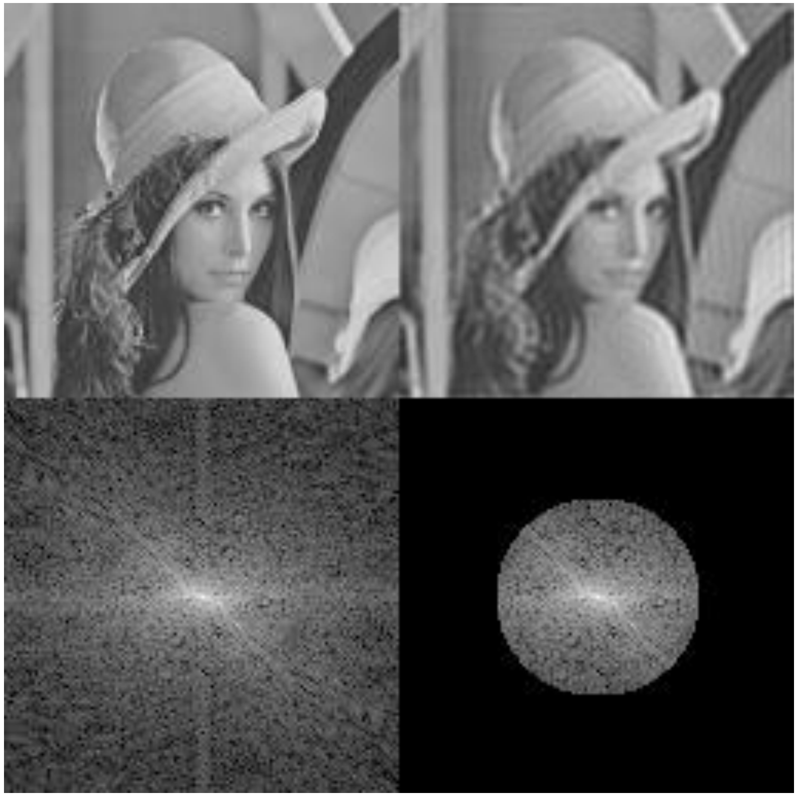


Note that the reconstruction using the phase shows the shape of the face, while using only the spectrum there is no shape information left

Low-pass Filtering

- Filtering by manipulating the frequency spectrum
- The “ideal” low-pass filter:





- This low pass filter gives us ringing artifacts

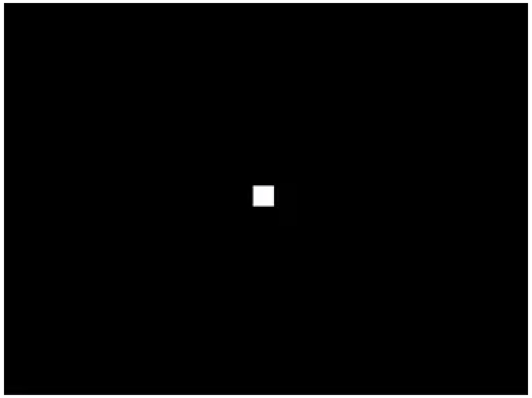
[www.cs.unm.edu/~brayer/vision/fourier.html]

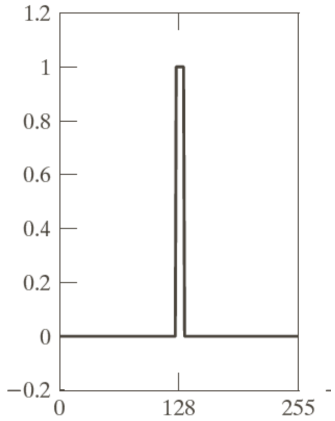
Low-pass Filtering

[www.cs.unm.edu/~brayer/vision/fourier.html]

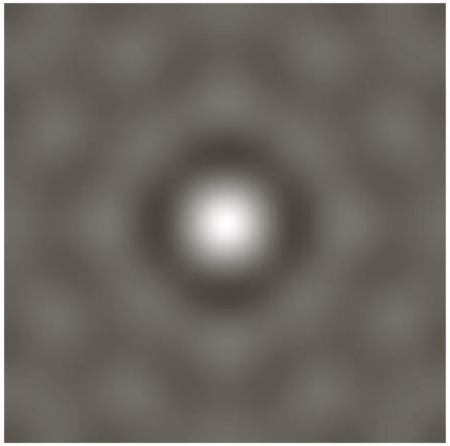
Ideal Low-Pass Filter

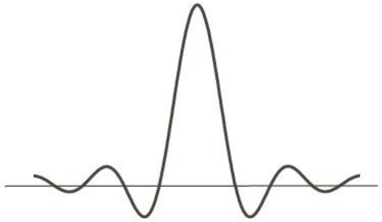
Filter in Frequency Domain





Filter in Spatial Domain





Sinc function

Here, we see why we have ringing artifacts from the ideal low-pass filter: the filter corresponds to a sinc function in the spatial domain.

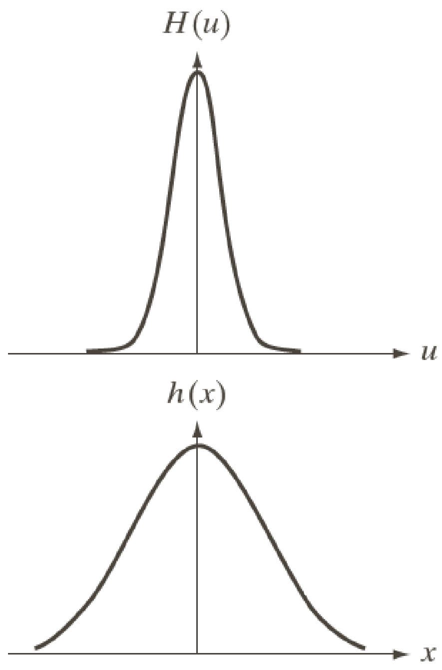
[Gonzalez/Woods]

Gaussian

A Gaussian in frequency domain transforms to a Gaussian in the spatial domain (and vice versa)

Frequency domain:

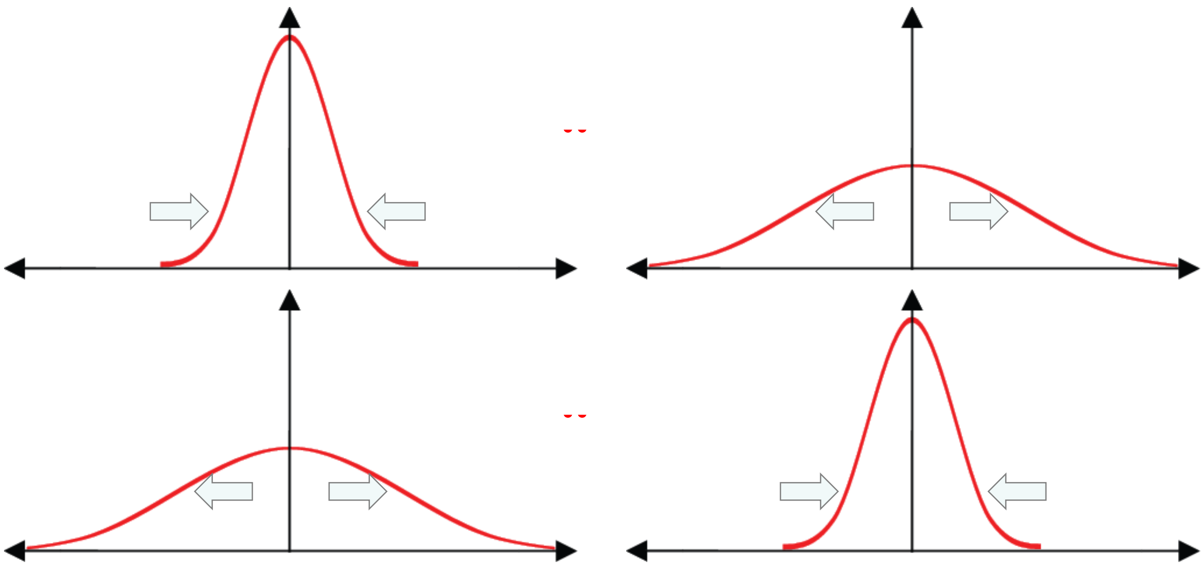
Spatial domain:



[Gonzalez/Woods]

Duality

The better a function is localized in one domain, the worse it is localized in the other.



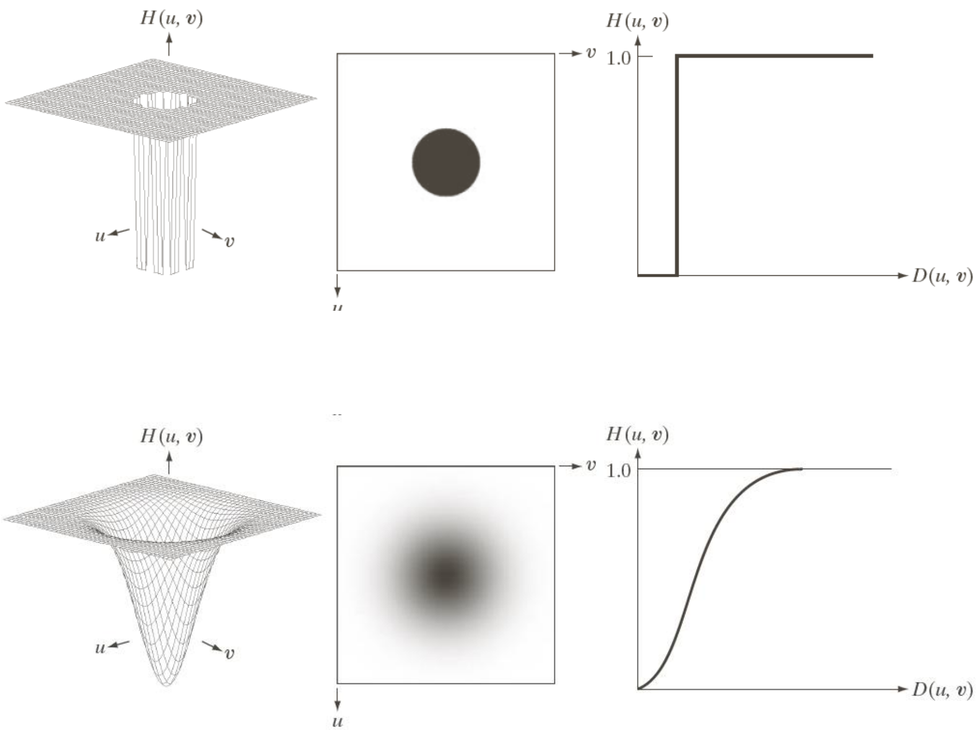
Gaussian Filter

Gaussian filtering with two different filter sizes

[Gonzalez/Woods]

High-pass Filters

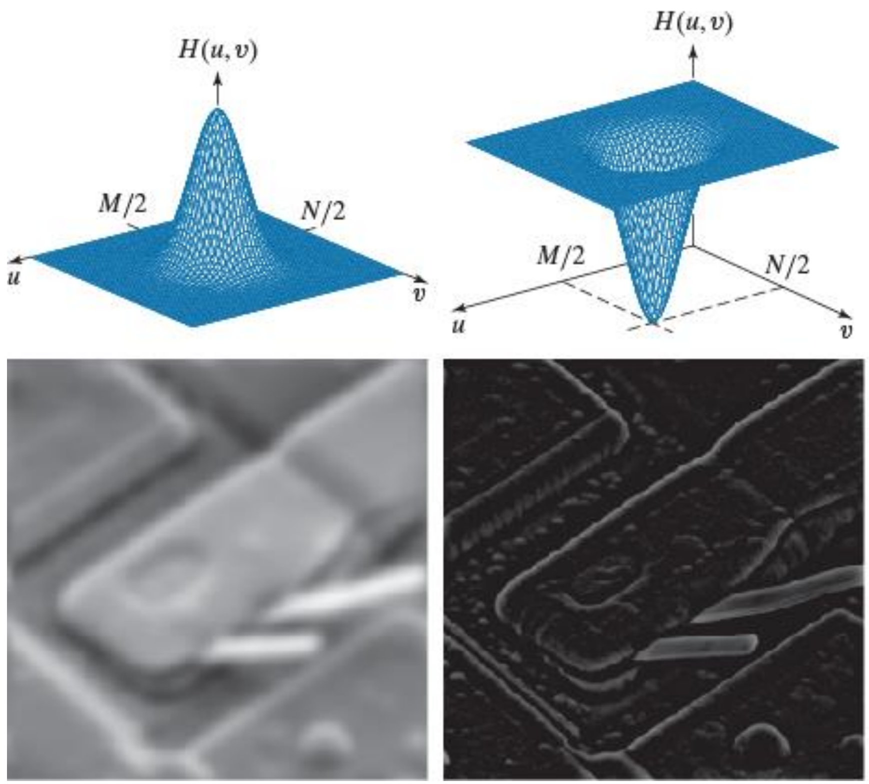
High-pass filters let high frequencies pass and suppress low frequencies



[Gonzalez/Woods]

Digital Filters in Frequency Domain

Low-pass filter High-pass filter



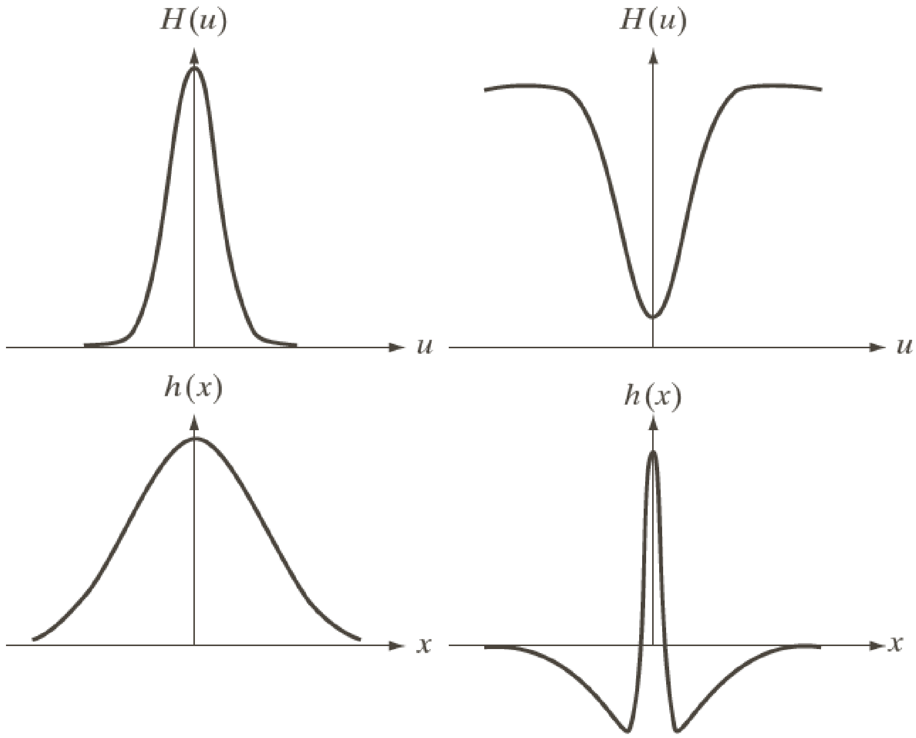
[Gonzales/Woods]

Low-pass/High-pass

Low-pass filter High-pass filter

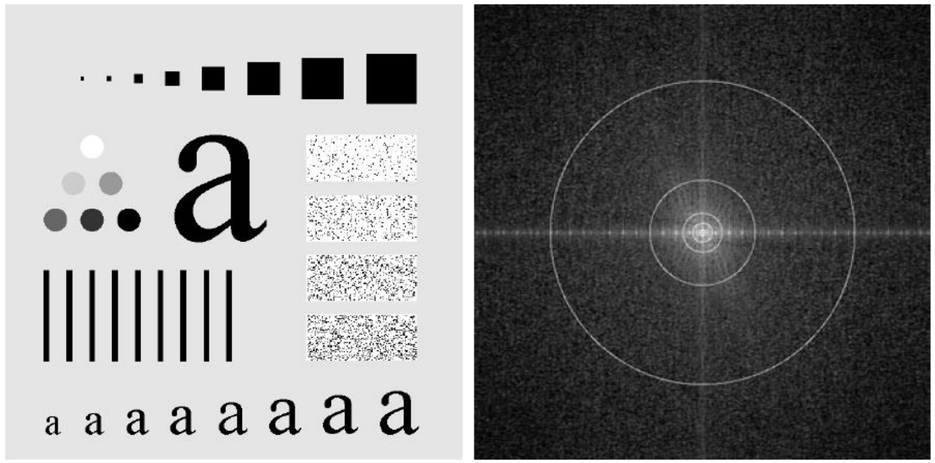
Frequency domain

Spatial domain



[Gonzalez/Woods]

High-pass Filtering



Spectrum with superimposed circles at radii 10, 30, 60, 160, 460

a b c

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

[Gonzalez/Woods]

Fast Convolution in 2D

Convolution theorem:
Convolution in the time/spatial domain equals point-wise multiplication in the frequency domain (also in 2D):

$$f(x,y) * g(x,y) \leftrightarrow F(u,v) \cdot G(u,v)$$

Time/spatial domain   Frequency domain

⇒ Convolution is much faster in the frequency domain.

⇒ But: FFT takes time itself! So, this only pays off for large filters (> 7x7, for separable filters 27). In computer vision we usually have small filters.

High-pass Filtering

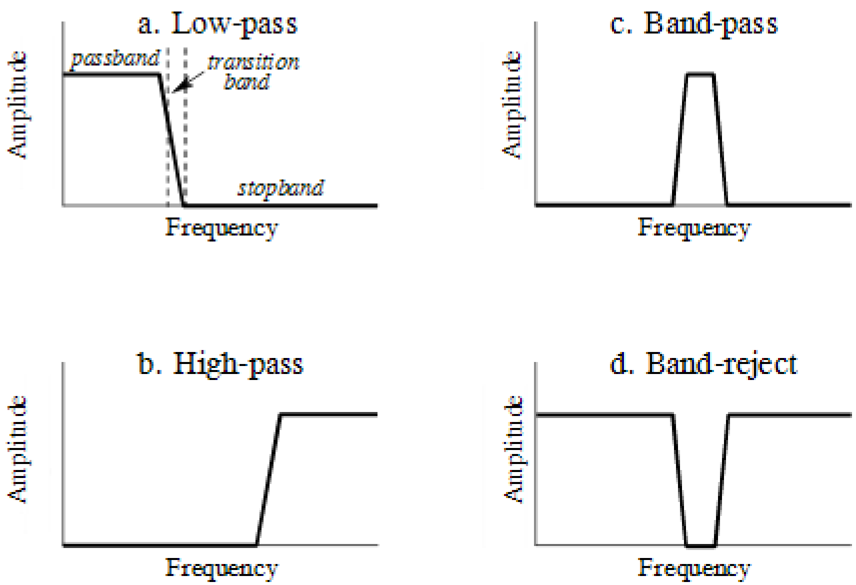
a b c

FIGURE 4.57 (a) Thumb print. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

[Gonzalez/Woods]

Digital Filters in Frequency Domain

- Some operations are easier and more intuitive in the frequency domain (e.g. if you want to design filters that manipulate only some frequencies)
- Examples:



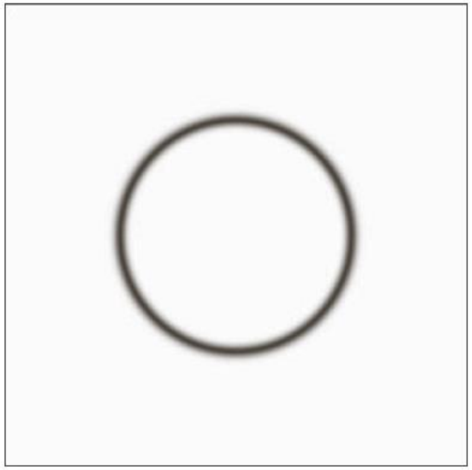
[Smith 1997]

Selective Filters

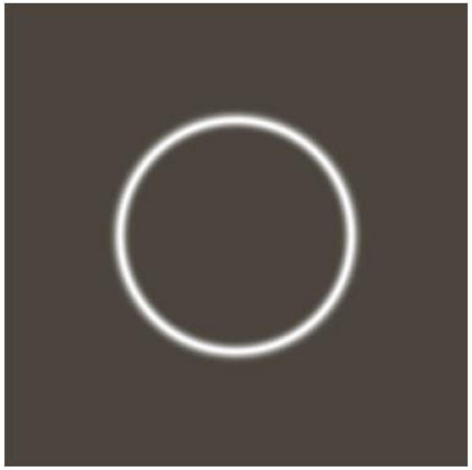
- Selective filters process specific parts of the spectrum (or: ignore specific parts)
- Types of selective filters:
  - Band-pass/band-reject filters
  - Notch filters

Band-pass/Band-reject Filter

- Band-pass filter let frequencies within a specific frequency band pass
- Band-reject filter reject frequencies within a specific frequency band pass and keep the remaining frequencies



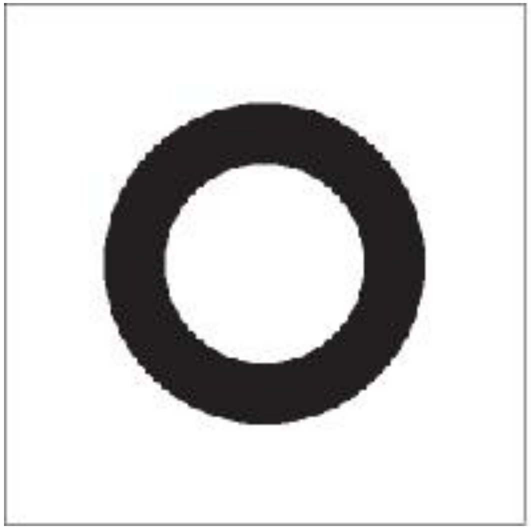
Band-reject filter



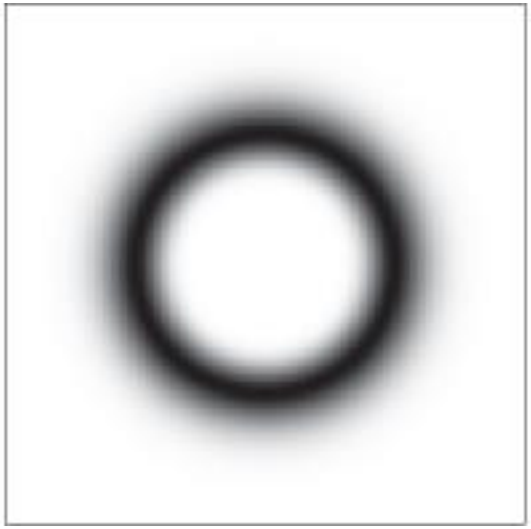
Band-pass filter

[Gonzalez/Woods]

Band-pass/Band-reject Filter



Ideal band-reject filter

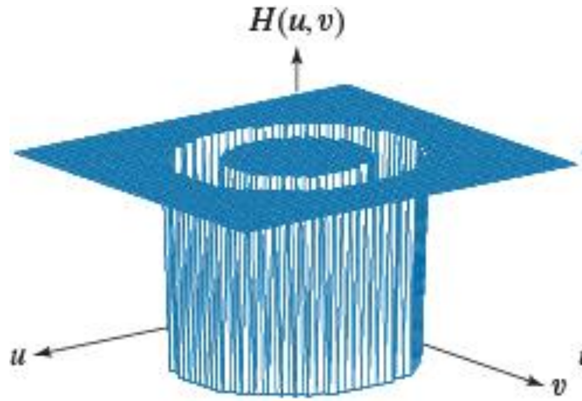


Gaussian band-reject filter

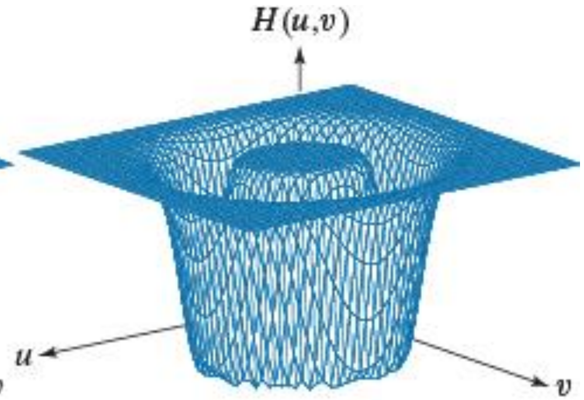
[Gonzalez/Woods]

Band-pass/Band-reject Filter

Perspective plots:



Ideal band-reject filter

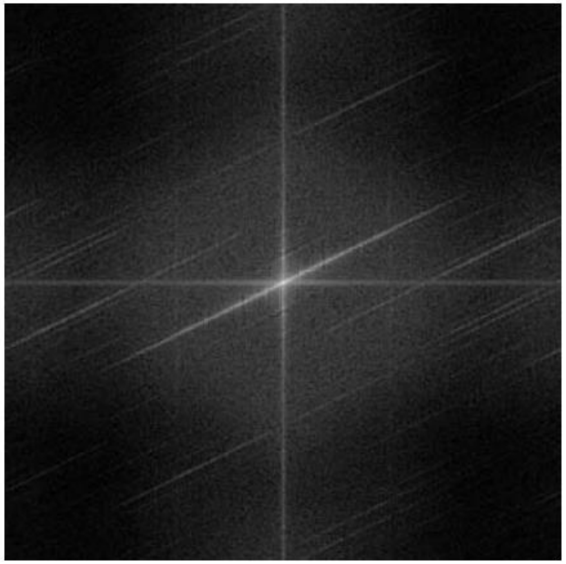


Gaussian band-reject filter

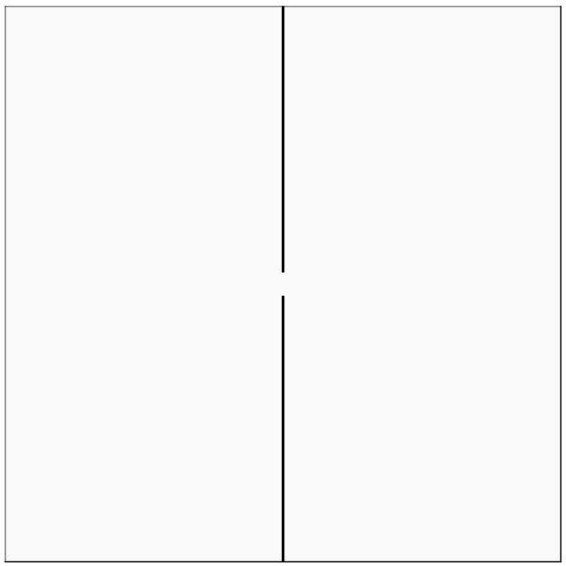
[Gonzalez/Woods]

Notch Filter

- A notch filter is a selective filter
- It rejects or passes frequencies in a pre-defined neighborhood



Spectrum



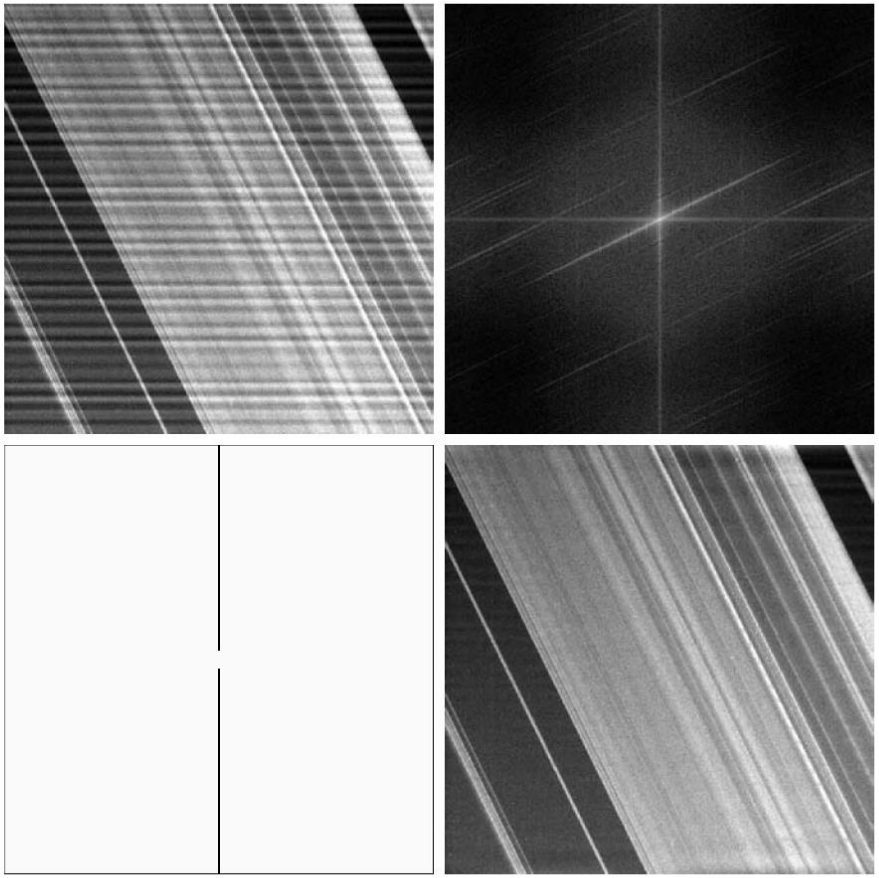
Notch reject filter

[Gonzalez/Woods]

Notch Filter

Original image of
Saturn rings

Spectrum



Notch reject filter

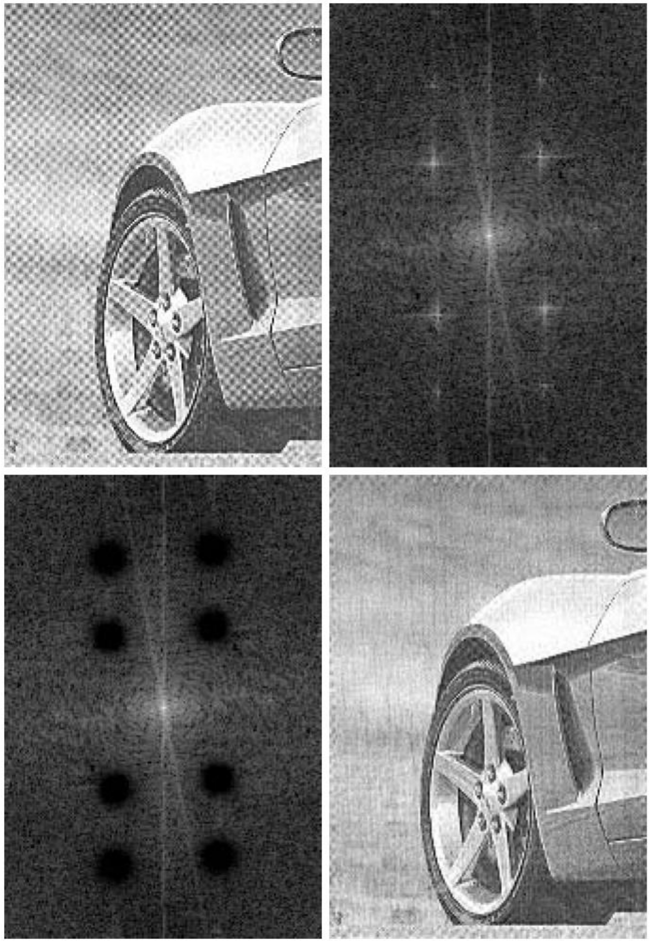
Image after applying
Notch reject filter

[Gonzalez/Woods]

Notch Filter

Original image

Spectrum



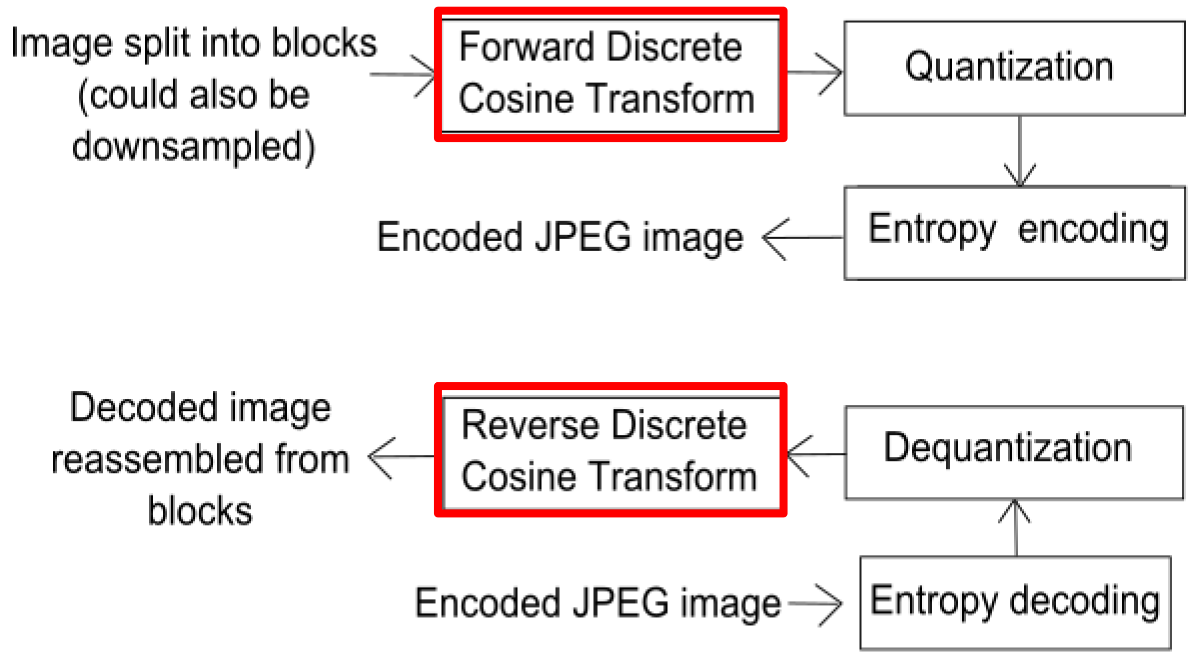
Notch reject filter
applied to spectrum

Image after applying
Notch reject filter

[Gonzalez/Woods]

Compression - JPEG

A variant of the DFT (the Discrete Cosine Transform (DCT)) is used in JPEG compression:



[Wikipedia: JPEG]

Quantization

Idea of quantization:

- humans can distinguish small differences in brightness over a relatively large area well (low frequencies), but not so good at distinguishing the exact strength of a high frequency brightness variation (high frequencies).
- Reduce information in high frequency components: Divide each frequency component by a frequency-specific constant. Use a quantization matrix Q with higher values for higher frequencies.

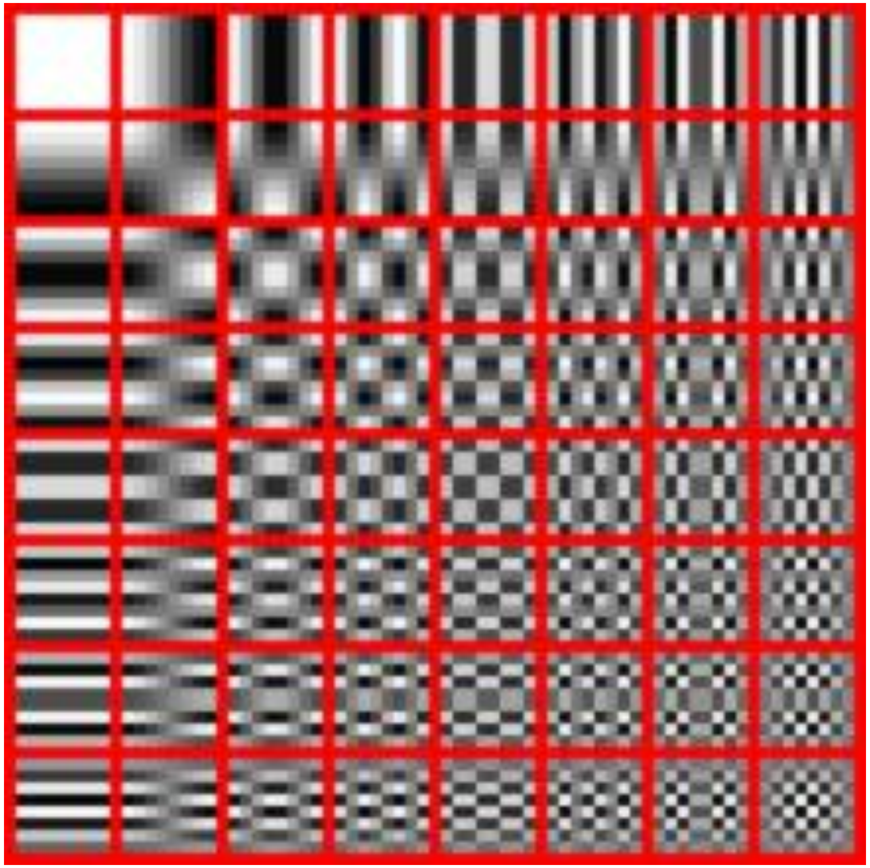
$$F^Q(x,y) = \mathrm{round}\left(\frac{F(x,y)}{Q(x,y)}\right)$$

- This process is lossy (the DCT is not)

Discrete Cosine Transform

64 Basis functions:

Lowest frequency

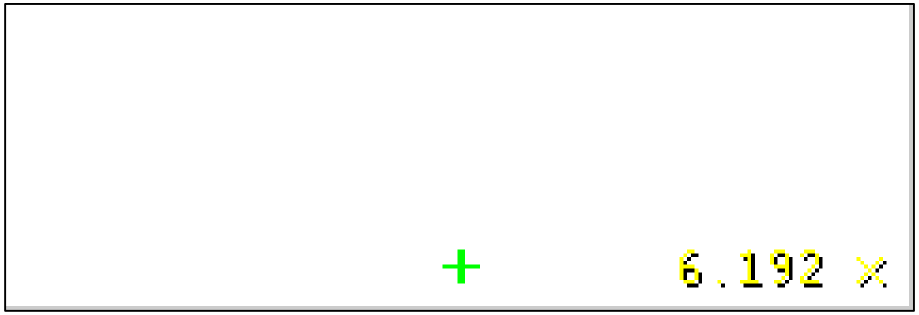


Highest frequency

[Wikipedia: Discrete Cosine Transform]

Discrete Cosine Transform

Step-by-step generating a signal (image) from scaled basis functions:



Current sum

Weighted basis function

Basis function

[Wikipedia: Discrete Cosine Transform]

Effect JPEG-Compression

Low compression

High compression

Notice the block structure!

[Wikipedia: JPEG]

Summary

The Frequency Domain:

- 1D signals:
  - Discrete Fourier Transformation (DFT) and Inverse DFT
  - Convolution in the frequency domain
  - Filtering in the frequency domain
- 2D signals (images):
  - 2D Discrete Fourier Transformation (2D DFT) and Inverse
  - 2D Convolution and filtering in the frequency domain
  - Understanding a 2D spectrum
  - Lowpass, Highpass, Bandpass and Notch filters

Take Home Message

- The DFT decomposes a signal into scaled basis function of different frequencies

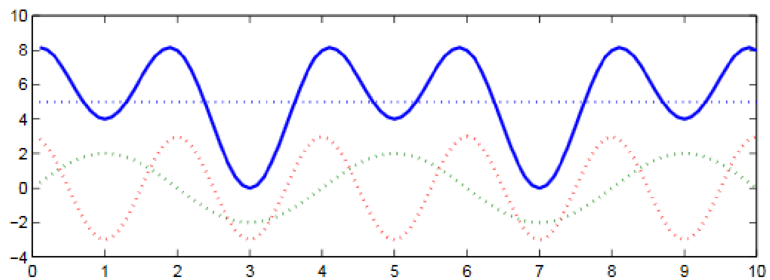
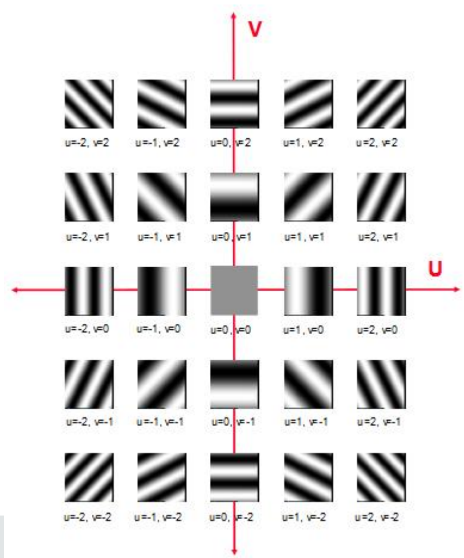


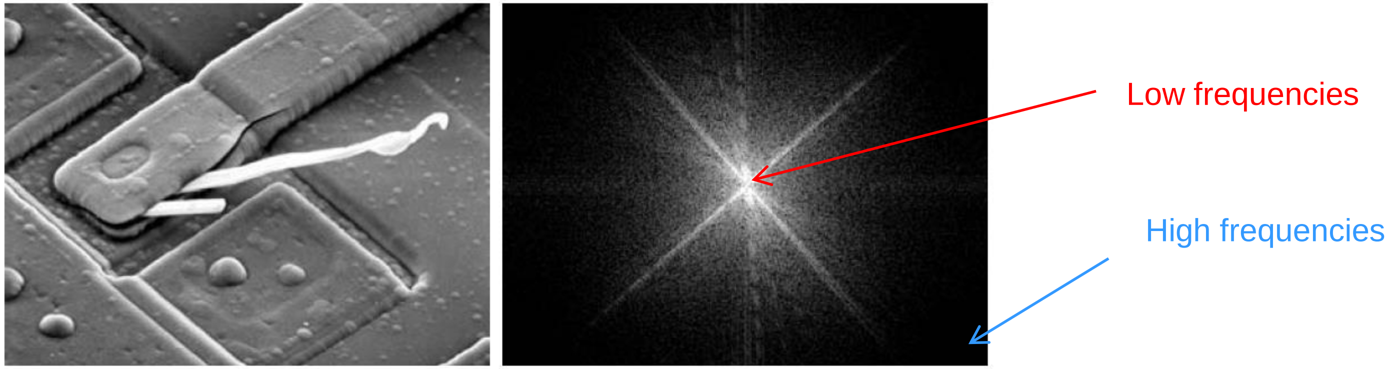
Figure 7.2: *Example signal for DFT.*

- Same in 2D:



Take Home Message

- The DFT decomposes a signal into scaled basis function of different frequencies
- The 2D spectrum shows how strongly each of the 2D basis functions is present in the image



- In the frequency domain we can easily remove undesired frequencies

Primary Literature

1D DFT & IDFT (real and complex):

- “The Scientist and Engineer's Guide to Digital Signal Processing, copyright ©1997-1998 by Steven W. Smith. For more information visit the book's website at: www.DSPguide.com”
(especially chapters 8 and 31)
Look at the pdfs for the individual chapters, the html is not complete!

2D DFT & IDFT:

- Gonzalez/Woods, 2017, 4th edition: relevant parts of chapter 4

Secondary Literature

- Stephen Roberts, Oxford-Man Institute, Lecture Notes on DFT: http://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf