Vision Systems

Lecture 4

Part 1

Image in the Frequency Domain

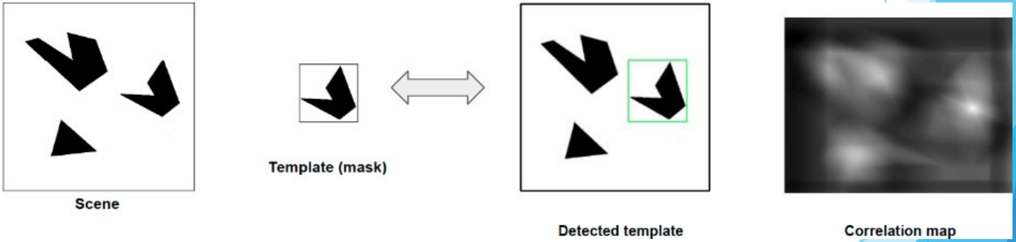
Review: Questions to Think About

- Do we then need (cross)-correlation at all?

- Are all filters always linear?

Is Correlation Still Useful?

- Can be used for Template Matching
- Filters look like objects they are intended to find ==⇒ use Normalized Cross-correlation (to control relative brightness) score to find a given pattern in an image



Credit: K Grauman, Univ of Texas Austin

Even if the template is not identical to some subimage in the scene, match can be meaningful, if scale, orientation and general appearance is right.

Is Correlation Still Useful?

Scene

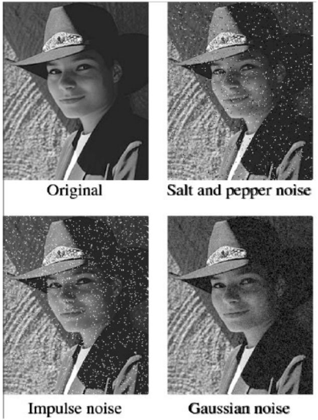
Template

Detected template

Credit: K Grauman, Univ of Texas Austin

Non-Linear Filters

Different types of noise in images



Reducing Salt-and-Pepper noise using Gaussian filters

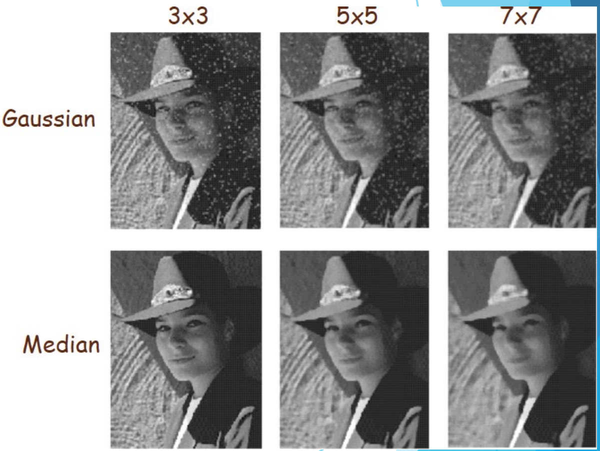
See the problem? What do we do?

Credit: S Seitz, Univ of Washington & J Ko˘seck´a, George Mason University

Non-Linear Filters: Median Filter

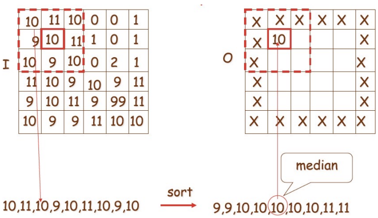
- Replace each pixel with MEDIAN value of all pixels in neighbourhood

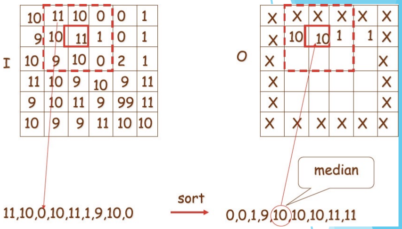
- Properties:
  - Non-linear
  - Does not spread noise
  - Can remove spike noise
  - Robust to outliers, but not good for Gaussian noise

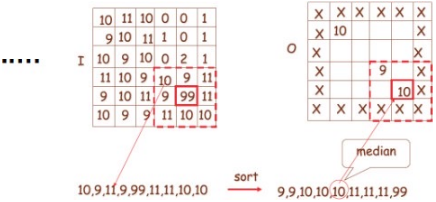


Credit: J Ko˘seck´a, George Mason University

Median Filter: Example







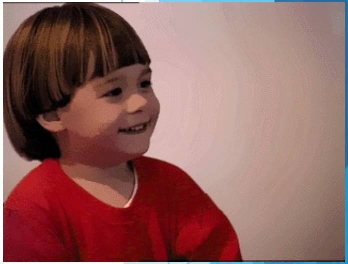
Notice how the outlier pixel value (99) got filtered out

Credit: J Ko˘seck´a, George Mason University

Non-Linear Filters: Bilateral Filtering

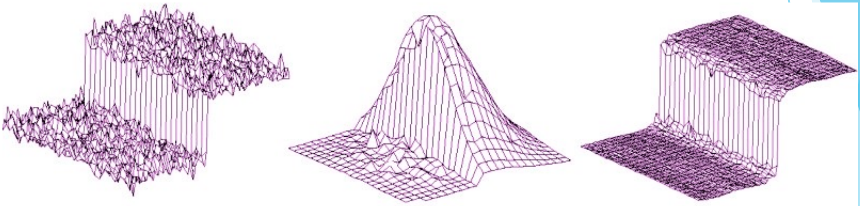
- Noise removal comes at expense of image blurring at edges
- **Bilateral filtering:** Simple, non-linear edge-preserving smoothing
- Reject (in a soft manner) pixels whose values differ too much from the central pixel value.
- Output pixel value is weighted combination of neighbouring pixel values: $g(i,j) = \frac{\sum_{k,l} I(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$
- Data-dependent bilateral weight function composed of domain and range kernel:

$$w(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{||I(i,j) - I(k,l)||^2}{2\sigma_r^2}\right)$$



Credit: Wikipedia; CVOnline

Bilateral Filters: Example



Given region in an image

23 x 23 filter centered two pixels to right of step in previous figure

Filtered output

Credit: CVOnline

What do we lose in a low-resolution image?

Credit: Derek Hoeim, UIUC; James Hays, Gatech

Fourier

Jean Baptist Joseph Fourier (1768-1830) had an idea in (1807).

- **Idea:** Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.

Fourier

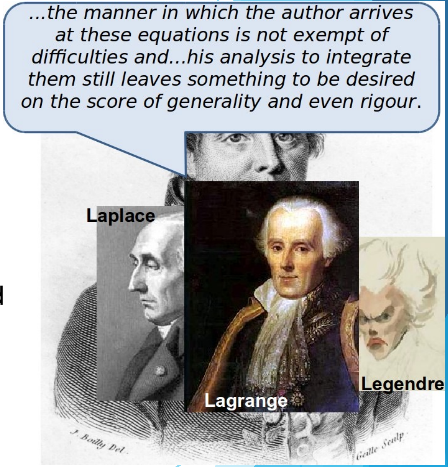
Jean Baptist Joseph Fourier (1768-1830) had an idea in (1807).

- **Idea:** Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.
- **Of course, what's new?**
  - Many including Lagrange, Laplace, Poisson and other big wigs did not believe him
    Not translated into English until 1878!

(Mostly) true!

Called Fourier Series Some subtle restrictions

Credit: James Hays

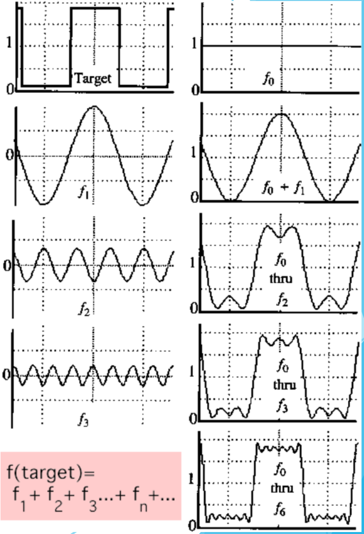


A sum of sines

- Building block:

$$A \sin(\omega x + \varphi)$$

- Add enough of them to get any signal f(x) you want!

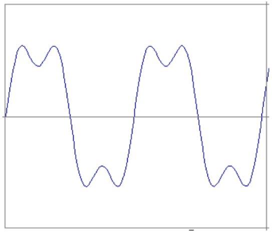


f(target)= $f_1 + f_2 + f_3 ... + f_n + ...$

Credit: James Hays, Gatech

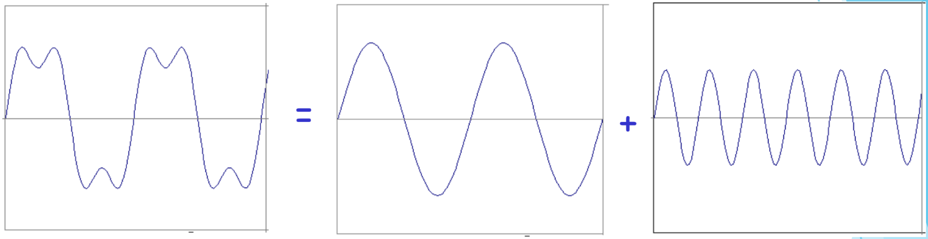
The Fourier Spectrum

Example: $g(t) = \sin(2pft) + (1/3)\sin(2p(3f)t)$



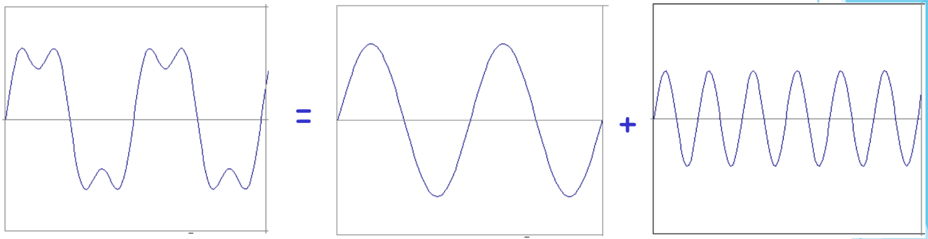
The Fourier Spectrum

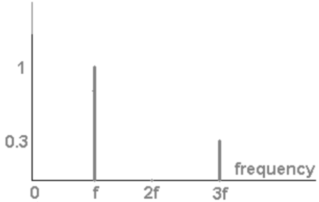
Example: $g(t) = \sin(2pft) + (1/3)\sin(2p(3f)t)$



The Fourier Spectrum

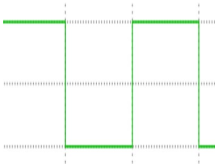
Example: $g(t) = \sin(2pft) + (1/3)\sin(2p(3f)t)$



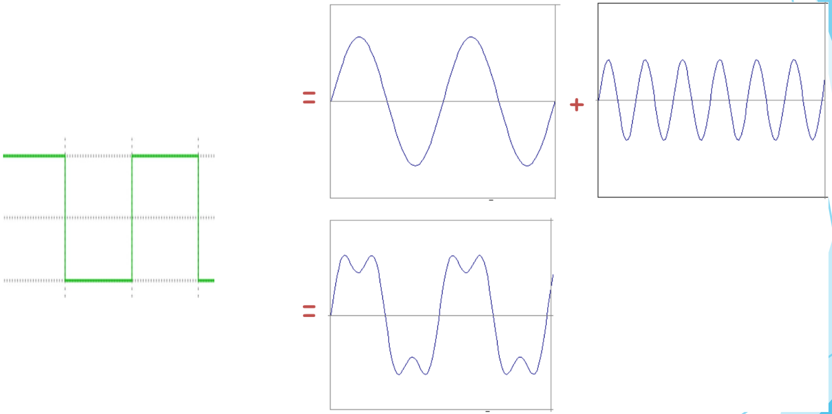


Credit: Alexei Efros, UCBerkeley; James Hays, Gatech

The Fourier Spectrum

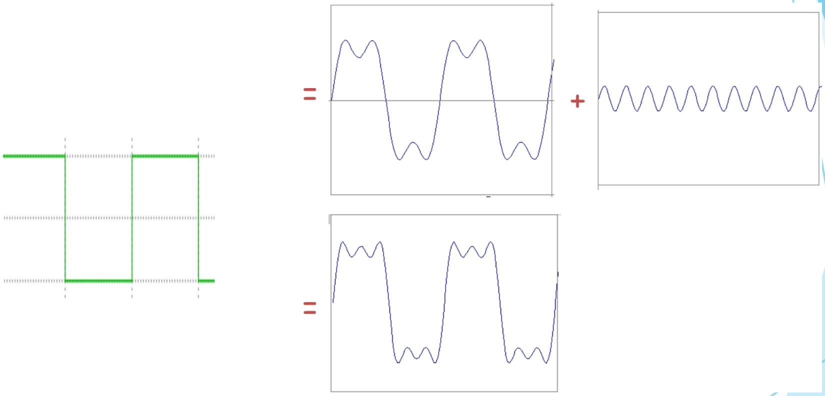


The Fourier Spectrum



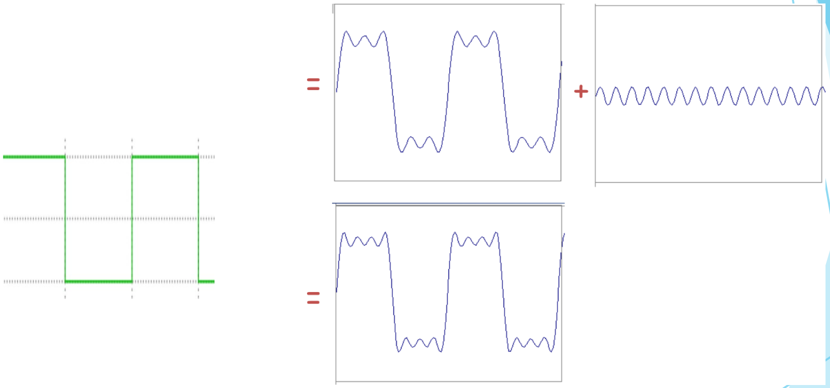
Credit: *James Hays, Gatech*

The Fourier Spectrum



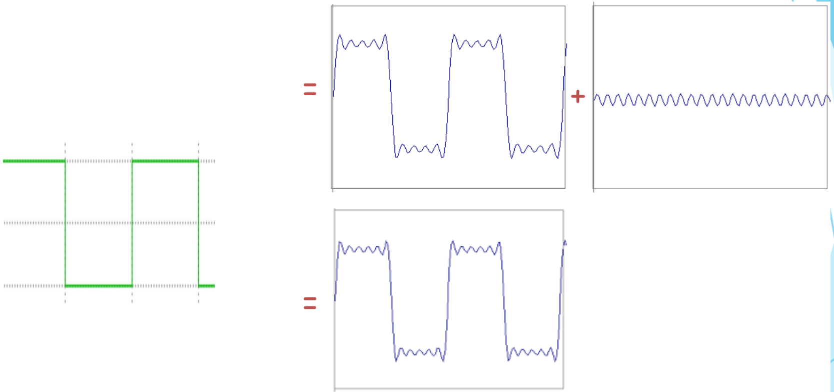
Credit: James Hays, Gatech

The Fourier Spectrum



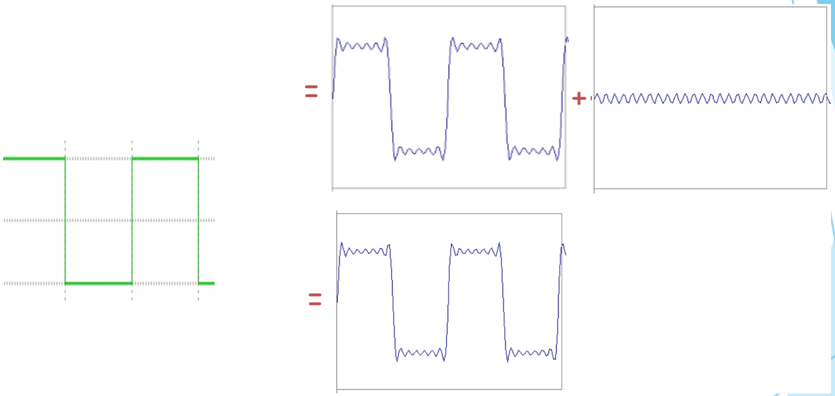
Credit: James Hays, Gatech

The Fourier Spectrum



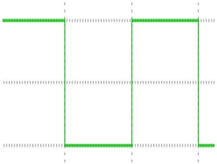
Credit: James Hays, Gatech

The Fourier Spectrum

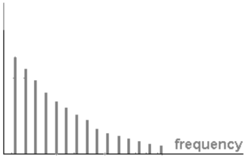


Credit: James Hays, Gatech

The Fourier Spectrum



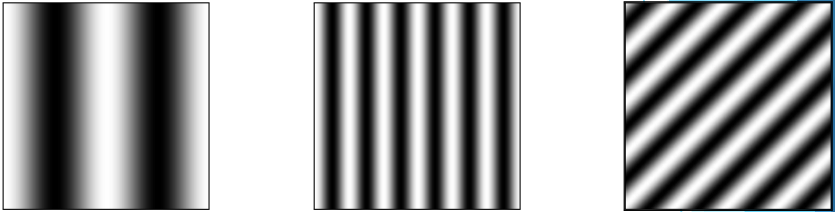
$$= A \sum_{1}^{\infty} \frac{1}{k} \sin(2\pi k t)$$



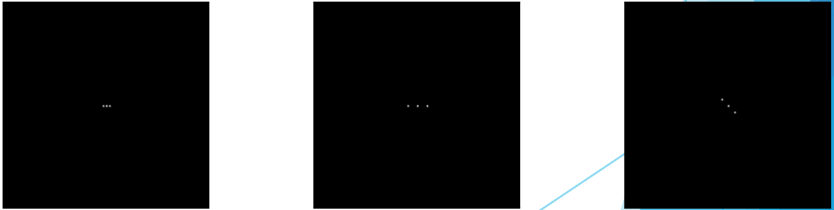
Credit: James Hays, Gatech

Fourier Analysis of Images

Intensity Image



Fourier Image

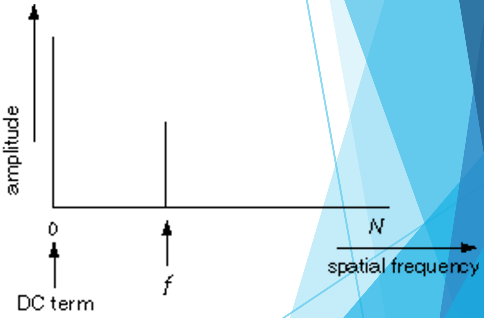


Credit: Derek Hoeim, UIUC

Fourier Analysis of Images

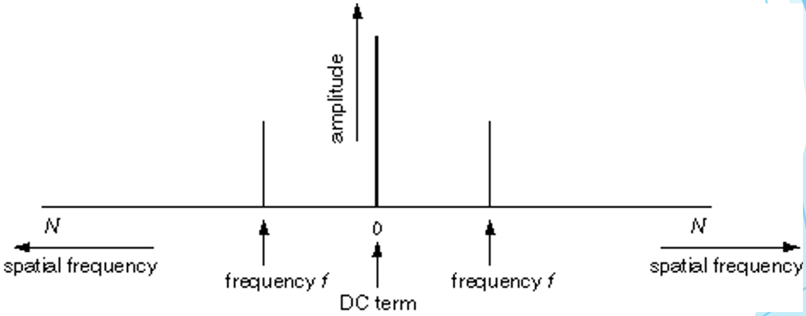
- Encodes a whole series of sinusoids through a range of spatial frequencies from zero all the way up to '*Nyquist frequency*' (more on this later)

- Signal containing only a single spatial frequency of frequency f is plotted as:
  - a single peak at point f along spatial frequency axis
  - height of that peak corresponding to the amplitude, or contrast of that sinusoidal signal



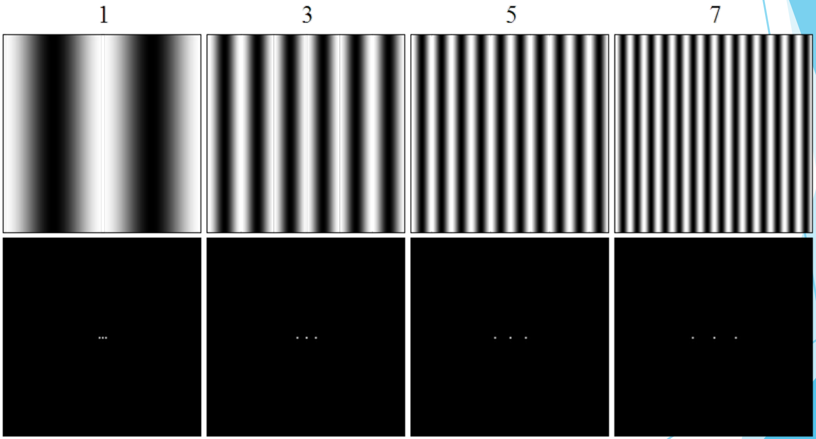
Fourier Analysis of Images

Mirror-image reflections along the axes

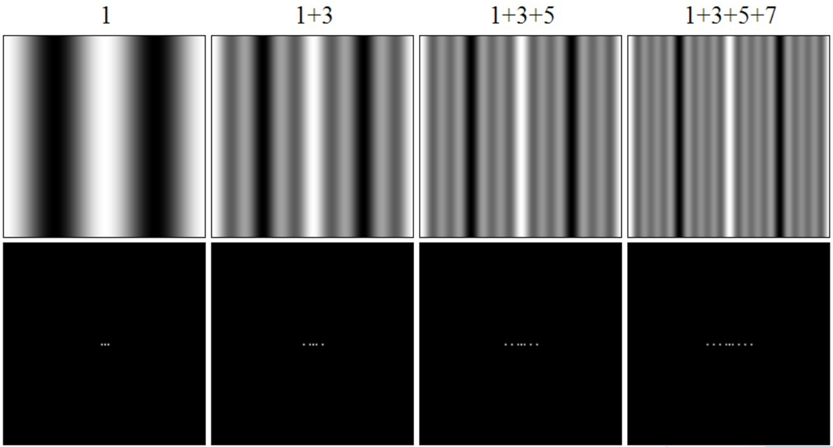


Why? See
http://dsp.stackexchange.com/questions/4825/why-is-the-fft-mirrored

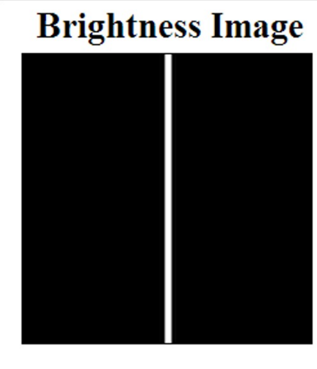
Fourier Analysis of Images: Examples

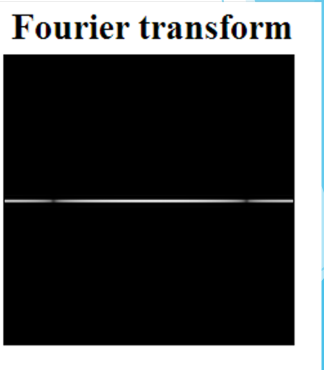


Fourier Analysis of Images: Examples

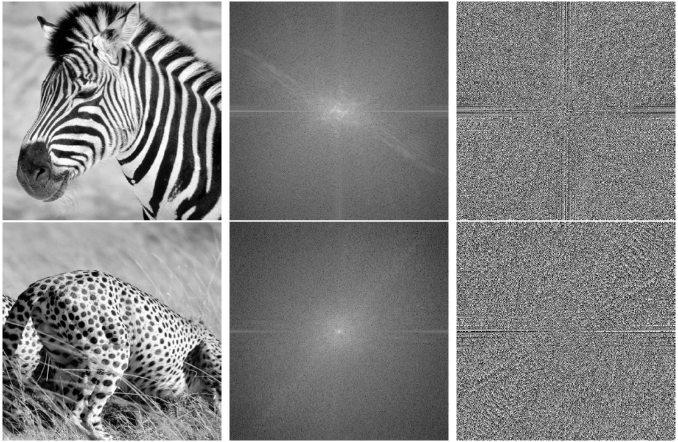


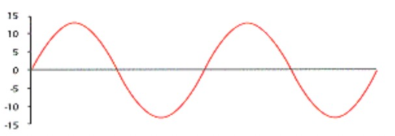
Fourier Analysis of Images: Examples

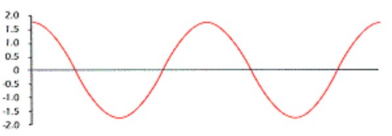




Fourier Analysis of Images: Examples







Credit: Forsyth and Ponce, Computer Vision: A Modern Approach, 2003

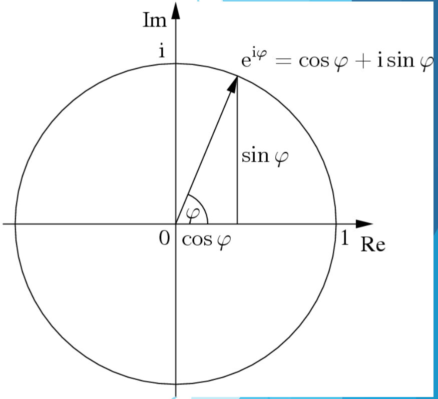
Fourier Transform: Magnitude and Phase

- Fourier transform stores the **magnitude** and **phase** at each frequency
  - For mathematical convenience, this is often denoted in terms of real and complex numbers
  - **Magnitude** encodes how much signal is there at a particular frequency

$$A = \pm\sqrt{Re(\phi)^2 + Im(\phi)^2}$$

  - **Phase** encodes spatial information (indirectly)

$$\varphi = \tan^{-1}\frac{Im(\phi)}{Re(\phi)}$$



Credit: Wikimedia Commons

Credit: Derek Hoeim, UIUC

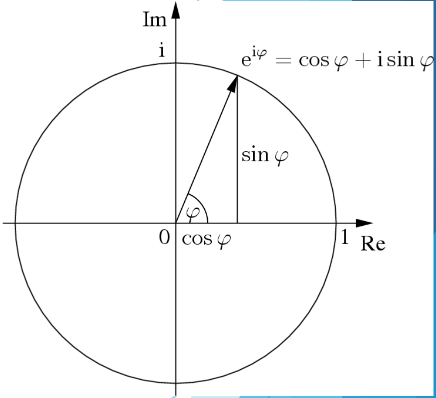
Fourier Transform: Magnitude and Phase

- Fourier transform stores the **magnitude** and **phase** at each frequency
  - For mathematical convenience, this is often denoted in terms of real and complex numbers
  - **Magnitude** encodes how much signal is there at a particular frequency

$$A = \pm\sqrt{Re(\phi)^2 + Im(\phi)^2}$$

  - **Phase** encodes spatial information (indirectly)

$$\varphi = \tan^{-1}\frac{Im(\phi)}{Re(\phi)}$$



Credit: Wikimedia Commons

Credit: Derek Hoeim, UIUC

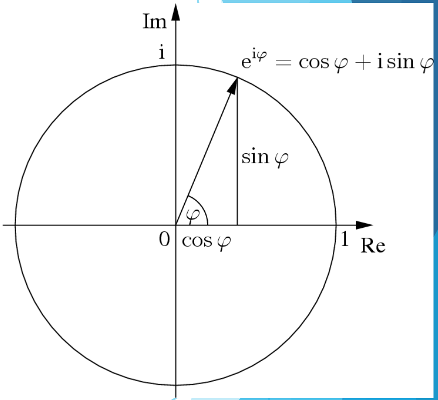
Fourier Transform: Magnitude and Phase

- Fourier transform stores the **magnitude** and **phase** at each frequency
  - For mathematical convenience, this is often denoted in terms of real and complex numbers
  - **Magnitude** encodes how much signal is there at a particular frequency

$$A = \pm\sqrt{Re(\phi)^2 + Im(\phi)^2}$$

  - **Phase** encodes spatial information (indirectly)

$$\varphi = \tan^{-1}\frac{Im(\phi)}{Re(\phi)}$$



Credit: Wikimedia Commons

Credit: Derek Hoeim, UIUC

Continuous vs Discrete Fourier Transform

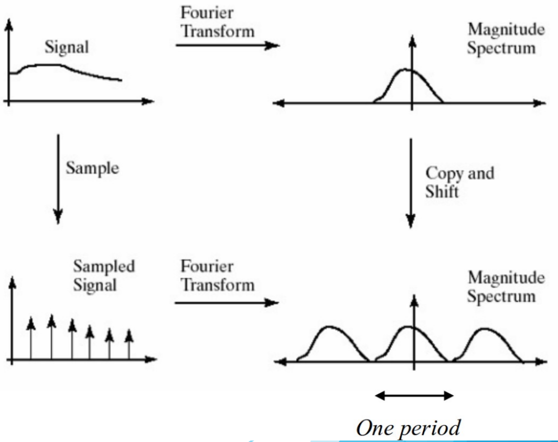
- **Continuous Fourier transform (FT):**

$$H(\omega) = \int_{-\infty}^{\infty} h(x)e^{-j\omega x}dx$$

- **Discrete Fourier Transform (DFT):**

$$H(\omega) = \sum_{0}^{N-1} h(x)e^{-j\frac{2\pi kx}{N}}dx$$

where N is the length of the sampled signal.



Credit: Elgammal, Rutgers University

More on the Fourier Transform

If you want to learn more on the Fourier transform

An intuitive explanation (highly recommended if you don’t have a background in signal processing): An Interactive Guide to the Fourier Transform

Other good tutorial-styled references:

Lecture by Lennart Lindegren, Lund University
An Introduction to the DFT
Wikipedia: Discrete Fourier Transform

Convolution Theorem

- Fourier transform of convolution of two functions is a product of their Fourier transforms:

$$F[g * h] = F[g]F[h]$$

- **Convolution** in the spatial domain can be obtained through **multiplication** in the frequency domain!

$$g * h = F^{-1}[F[g]F[h]]$$

Credit: James Hays, Gatech

Properties of Fourier Transform

Property	Signal		Transform
superposition	$f_1(x) + f_2(x)$		$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$		$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$		$F^*(\omega)$
convolution	$f(x) * h(x)$		$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$		$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$		$F(\omega) * H(\omega)$
differentiation	$f'(x)$		$j\omega F(\omega)$
domain scaling	$f(ax)$		$1/aF(\omega/a)$
real images	$f(x) = f^*(x)$	$\Leftrightarrow$	$F(\omega) = F(-\omega)$
Parseval's Theorem	$\sum_x [f(x)]^2$	$=$	$\sum_\omega [F(\omega)]^2$

Credit: Szeliski, Computer Vision: Algorithms and Applications, 2010

Fast Fourier Transform

- Number of arithmetic operations to compute Fourier transform of N numbers (i.e., function defined at N points) is....

Fast Fourier Transform

- Number of arithmetic operations to compute Fourier transform of N numbers (i.e., function defined at N points) is.... proportional to N^2
- Possible to reduce this to $N \log N$ using **Fast Fourier Transform (FFT)**

For more, see https://www.karlsims.com/fft.html

Fast Fourier Transform

- Number of arithmetic operations to compute Fourier transform of N numbers (i.e., function defined at N points) is.... proportional to N^2
- Possible to reduce this to $N \log N$ using **Fast Fourier Transform (FFT)**
- FFT is a **recursive divide-and-conquer algorithm** for computing DFT

For more, see https://www.karlsims.com/fft.html

Fast Fourier Transform

- Number of arithmetic operations to compute Fourier transform of N numbers (i.e., function defined at N points) is…. proportional to N^2
- Possible to reduce this to $N \log N$ using **Fast Fourier Transform (FFT)** FFT is a **recursive divide-and-conquer algorithm** for computing DFT Applications of FFT? Examples: Convolution, correlation

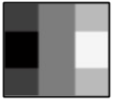
For more, see https://www.karlsims.com/fft.html

Filtering in Spatial Domain

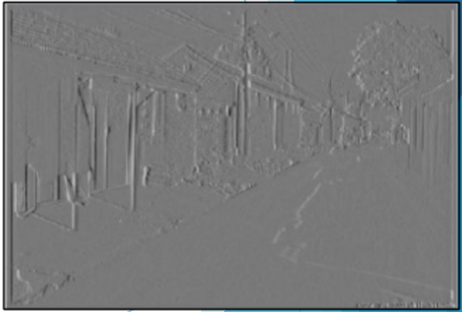
1	0	-1
2	0	-2
1	0	-1

Intensity Image

*

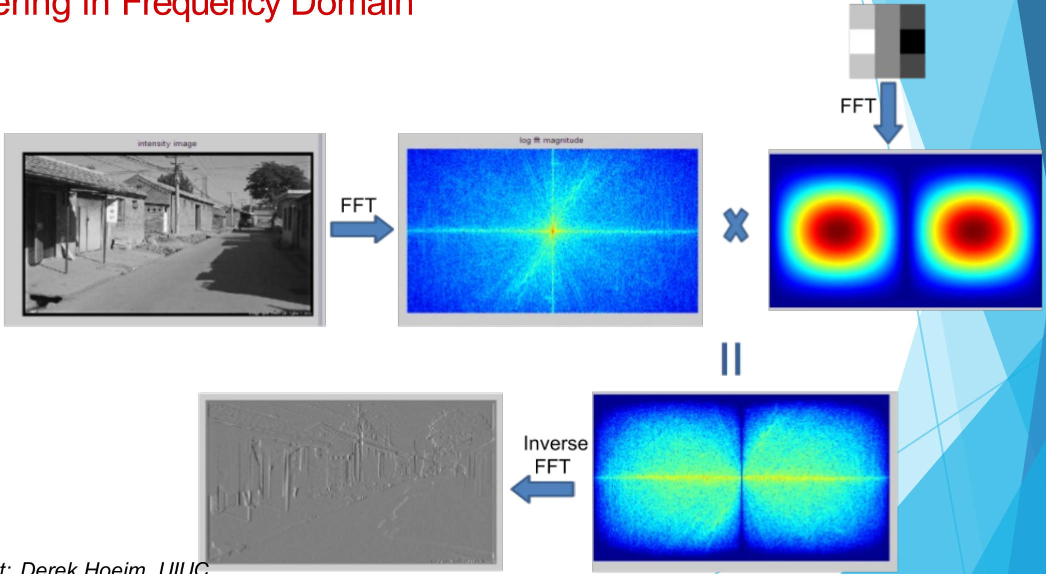


=



Credit: Derek Hoeim

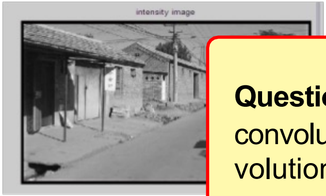
Filtering in Frequency Domain

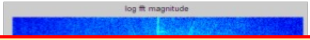


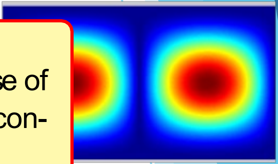
Credit: Derek Hoeim, UIUC

Filtering in Frequency Domain

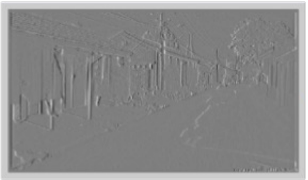
FFT



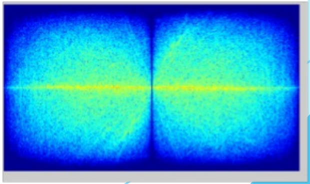




Question: What cost improvement does use of convolution theorem (or FFT over vanilla convolution) give?



Inverse FFT



Low-Pass and High-Pass Filters

Low-Pass Filters: Filters that allow low frequencies to pass through (block high frequencies). Example?

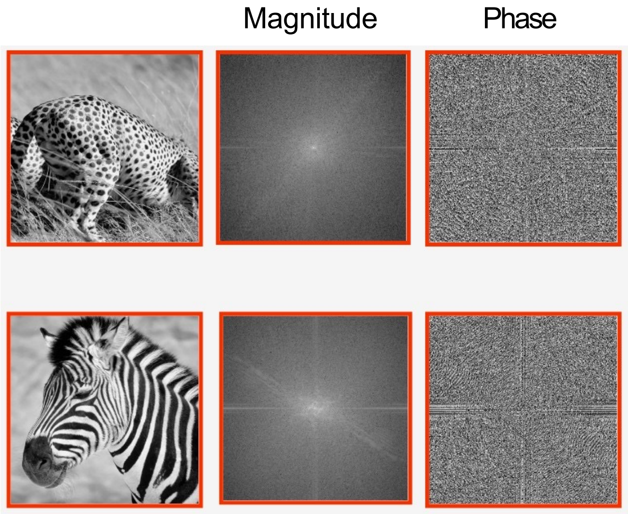
Low-Pass and High-Pass Filters

- **Low-Pass Filters:** Filters that allow low frequencies to pass through (block high frequencies). Example?
  - Gaussian filter
- **High-Pass Filters:** Filters that allow high frequencies to pass through (block low frequencies). Example?

Low-Pass and High-Pass Filters

- **Low-Pass Filters:** Filters that allow low frequencies to pass through (block high frequencies). Example?
  - Gaussian filter
- **High-Pass Filters:** Filters that allow high frequencies to pass through (block low frequencies). Example?
  - Edge filter

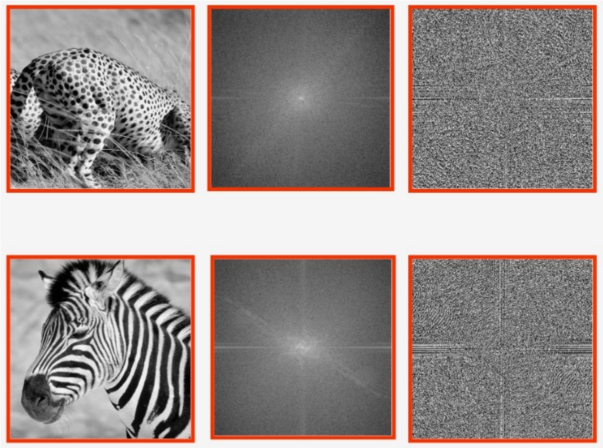
Which has more information: Magnitude or Phase?



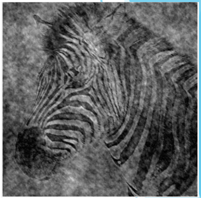
Swap phase and reconstruct?

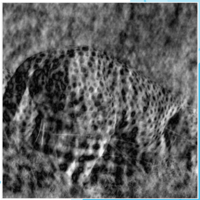
Which has more information: Magnitude or Phase?

Magnitude Phase



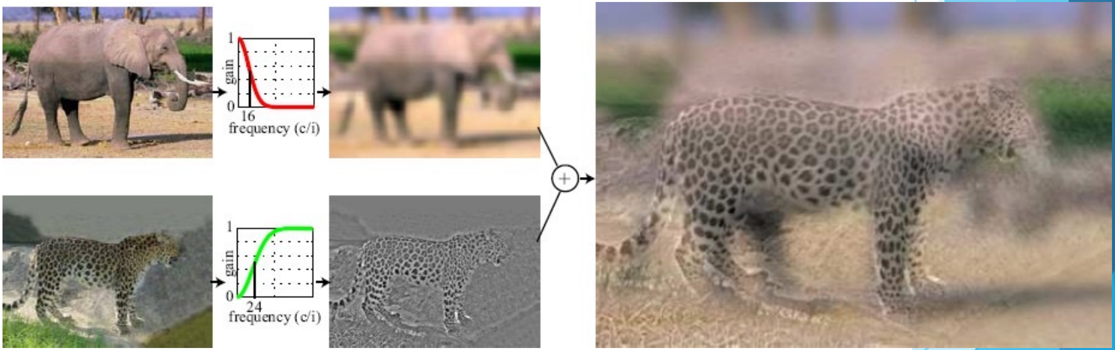
Swap phase and reconstruct?





Credit: Forsyth and Ponce, Computer Vision: A Modern Approach, 2003

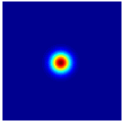
Hybrid Images



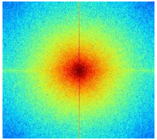
Credit: A. Oliva, A. Torralba, P.G. Schyns, [“Hybrid Images,”]() SIGGRAPH 2006

Exercise: Match spatial domain image to Fourier magnitude image

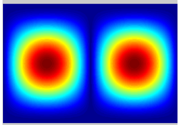
1



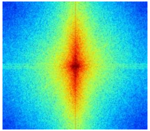
2



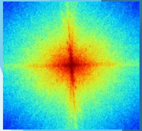
3



4



5

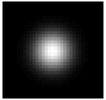


A

B

C

D



E

Homework

Readings

- Chapter 3.4, Szeliski, *Computer Vision: Algorithms and Applications*
- For more information on fourier transforms:
  - http://www.thefouriertransform.com
  - http://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/
  - http://wwwpub.zih.tu-dresden.de/~ds24/lehre/bvme_ss_2013/ip_05_fourier.pdf
- Other links provided on respective slides

Questions/Exercises

- What cost improvement does convolution theorem give?
- Complete the matching exercise

References

- Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. London: Springer-Verlag, 2011.
- David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. 2 edition. Boston: Pearson Education India, 2015.
- *Hays, James, CS 6476 - Computer Vision (Fall 2018)*. URL:
- https://www.cc.gatech.edu/~hays/compvision/ (visited on 04/28/2020).
- *Hoiem, Derek, CS 543 - Computer Vision (Spring 2011)*. URL:
- https://courses.engr.illinois.edu/cs543/sp2017/ (visited on 04/25/2020).
- *Oliva, Aude, 6.819/6.869 - Advances in Computer Vision (Fall 2015)*. URL:
- http://6.869.csail.mit.edu/fa15/ (visited on 04/28/2020).