Machine learning for vision

Winter 2015

Roland Memisevic

Lecture 2, January 28, 2015

2d waves

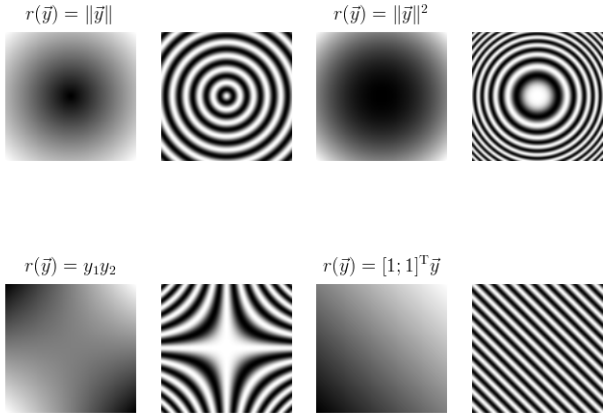
- To represent images, we need to generalize the concept of oscillation to 2d.
- Since oscillations are functions of a scalar t, we can do this by first assigning a scalar to an image position, and then passing the scalar to a phasor:

$$I(\boldsymbol{y}) = e^{ir(\boldsymbol{y})}$$

where $\boldsymbol{y}$ is a 2d coordinate and r is some scalar-valued function.

2d waves

Examples for some functions $r(\boldsymbol{y})$ and the resulting 2d waves (real part)



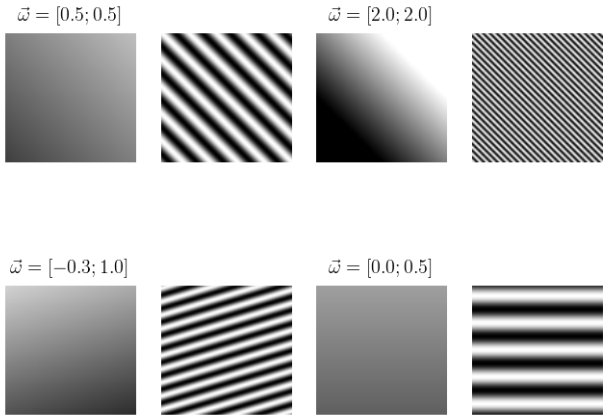
2d waves

- The most commonly used waves are those where the scalar is a linear function of 2d-position:

$$r(\boldsymbol{y}) = \omega^{\mathrm{T}}\boldsymbol{y}, \quad I(\boldsymbol{y}) = \exp(i\omega^{\mathrm{T}}\boldsymbol{y})$$

- Recall that $\omega^{\mathrm{T}}\boldsymbol{y}$ grows in the direction of ω and is constant in the direction orthogonal to ω.
- ω is now called *frequency vector*.
- We can define higher-dimensional waves in the same way. The 3-d case is useful when dealing with videos.

2d waves



2d waves, comments

- Wavefronts point in the direction of $\boldsymbol{\omega}$
- The angle between $\boldsymbol{\omega}$ and the x-axis is $\mathrm{atan}(\frac{\omega_2}{\omega_1})$ (But this angle has of course nothing to do with the phase angle of the oscillation)
- A useful 2d-analog of the 1d angular frequency ω is the *norm* $\|\boldsymbol{\omega}\|$ of the frequency vector $\boldsymbol{\omega}$:
  Set $f(t) = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} t$ with real valued t, and you get a 1d wave in the direction of $\boldsymbol{\omega}$, where t plays the same role as in 1d.
- A projection parallel to a coordinate axis i yields a 1d phasor with angular frequency ω_i.

Separability of complex waves

- Complex valued waves are **separable**:

$$\begin{aligned} I(\boldsymbol{y}) &= \exp(i(\boldsymbol{\omega}^T \boldsymbol{y})) \\ &= \exp(i\omega_1 y_1 + i\omega_2 y_2)) \\ &= \exp(i\omega_1 y_1) \cdot \exp(i\omega_2 y_2) \\ &=: I_1(y_1) \cdot I_2(y_2) \end{aligned}$$

- The same is not true for real valued waves.
- Separability can be used to prove orthogonality of complex waves in 2d.

Real waves are sums of complex waves

- Using the relationships between cos, sin and the phasor,

$$\begin{aligned} \cos\varphi &= \frac{1}{2} \cdot e^{i\varphi} + \frac{1}{2} \cdot e^{-i\varphi} \\ \sin\varphi &= \frac{1}{2i} \cdot e^{i\varphi} - \frac{1}{2i} \cdot e^{-i\varphi} = -\frac{i}{2} \cdot e^{i\varphi} + \frac{i}{2} \cdot e^{-i\varphi}, \end{aligned}$$

  we may write
- $\cos\left(\boldsymbol{\omega}^T \boldsymbol{y}\right) = \frac{1}{2} \exp\left(i\boldsymbol{\omega}^T \boldsymbol{y}\right) + \frac{1}{2} \exp\left(-i\boldsymbol{\omega}^T \boldsymbol{y}\right)$
- $\sin\left(\boldsymbol{\omega}^T \boldsymbol{y}\right) = -\frac{i}{2} \exp\left(i\boldsymbol{\omega}^T \boldsymbol{y}\right) + \frac{i}{2} \exp\left(-i\boldsymbol{\omega}^T \boldsymbol{y}\right)$

DFT on images

- Using the discrete phase vector ω with components $\omega_1 = \frac{2\pi}{M}k$, $\omega_2 = \frac{2\pi}{N}\ell$ we get the

Discrete Fourier Transform (DFT) in 2d

$$S(k,\ell) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} s(m,n)e^{-i2\pi\left(\frac{km}{M}+\frac{\ell n}{N}\right)}$$

Inverse Discrete Fourier Transform in 2d

$$s(m,n) = \frac{1}{MN}\sum_{k=0}^{M-1}\sum_{\ell=0}^{N-1} S(k,\ell)e^{i2\pi\left(\frac{km}{M}+\frac{\ell n}{N}\right)}$$

- The other Fourier transform variations can be generalized to 2d accordingly.

DFT Symmetries

1. For discrete phasors

- 1d: The phasor $p(k)$ is equal to the conjugate phasor $\bar{p}(T-k)$.
- 2d: The phasor $p(k,\ell)$ is equal to the conjugate phasor $\bar{p}(M-k, N-\ell)$.

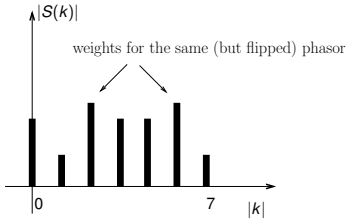
- So a "fast" phasor is the same as a "inverse slow" phasor, and vice versa.
- Here, "conjugate phasor" means elementwise.

DFT Symmetries

2. For real valued signals

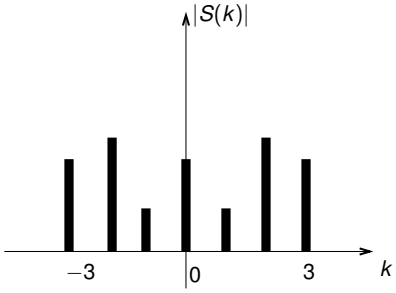
- 1d: $S(k) = \bar{S}(T-k)$
- 2d: $S(k,\ell) = \bar{S}(M-k, N-\ell)$

- So the amplitude spectrum of real valued signals is symmetric.

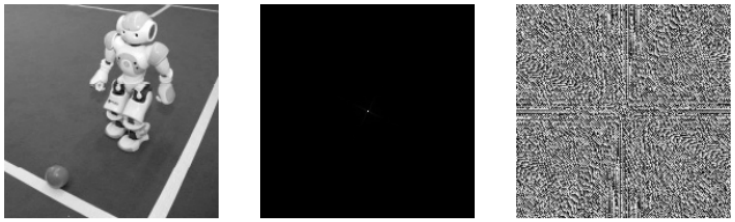


Practical implications of symmetries

- Because of this, the Fourier transform is usually presented with the origin in the center, and increasing (absolute) amplitudes away from the origin.
- All common software packages provide a function for this (such as "*fftshift*").

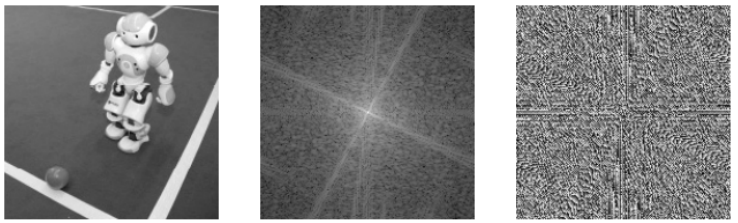


Spectrum example



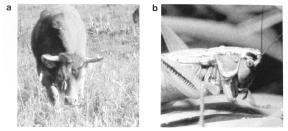
- The amplitude spectrum is usually shown on a logarithmic scale, because it can span several orders of magnitude.

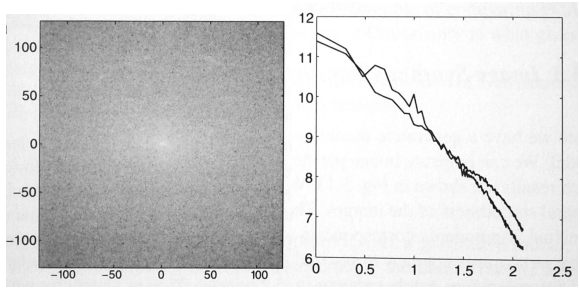
Spectrum example



- The amplitude spectrum is usually shown on a logarithmic scale, because it can span several orders of magnitude.

More amplitude spectra (average over cross-sections on the right)





from: Natural Image Statistics (Hyvarinen, Hurri, Hoyer; 2009)

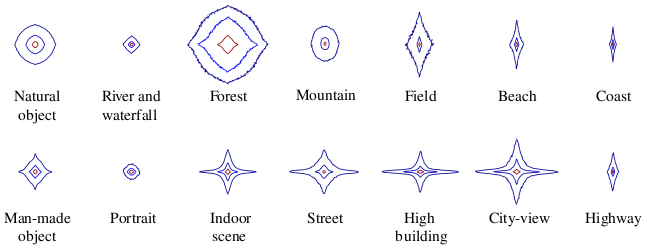
Amplitude spectra of natural images

- Amplitude spectra of natural images show strong regularity.
- They typically are roughly of the form

$$\frac{1}{f}$$

  where f is the absolute value of frequency.
- Equivalently, the power spectrum $|S(f)|^2$ takes the form $\frac{1}{f^2}$
- This is known as **power law**.
- Power law relationships occur frequently in nature and they can be related to scale invariance.
- They can also be related to the covariance structure of natural images.

Statistics on image spectra



- from Torralba, Oliva; 2003

Fourier transform and convolution

- Consider the convolved signal $g(t) = s(t) * h(t) = \sum_k h(k) \cdot s(t-k)$
- Its (discrete-time) Fourier transform is

$$\begin{aligned} G(\omega) &= \sum_t \left[\sum_k h(k) \cdot s(t-k) \right] e^{-i\omega t} \\ &= \sum_t \sum_k h(k) \cdot e^{-i\omega k} \cdot s(t-k)\, e^{-i\omega(t-k)} \\ &= \sum_k h(k) \cdot e^{-i\omega k} \cdot \sum_t s(t-k)\, e^{-i\omega(t-k)} \\ &= H(\omega) \cdot S(\omega) \end{aligned}$$

Convolution in the time-domain is multiplication in the frequency domain.

Fourier transform and convolution

- So to convolve a signal $s(t)$ with a signal $h(t)$ we can either:
- Compute the convolution sum, or
- 1. compute the Fourier transforms $S(\omega)$ and $H(\omega)$
  2. compute the element-wise product $G(\omega) = S(\omega)H(\omega)$
  3. use the inverse Fourier transform to get $g(t) = s * h$
- Depending on the situation (like size/number of images or filters, separable or not, etc.) any one of the two may be more efficient.

Effects of multipyling a signal

- One can show similarly:

Multiplication in the time-domain is convolution in the frequency domain.

- Multiplying a signal or its spectrum elementwise with some function is a very common operation in practice.
- In practice, the relation between convolution and multiplication often leads to unwanted effects.
- Examples include **ringing**, **aliasing** and **leakage**.

Low-pass filtering in the frequency domain

- An easy way to define a low-pass filter is to multiply low frequency components by zero, then to do the inverse transform to get back the resulting signal.
- However, this will yield ringing artefacts:

Leakage from Fourier transform of a box

- The box function is defined as

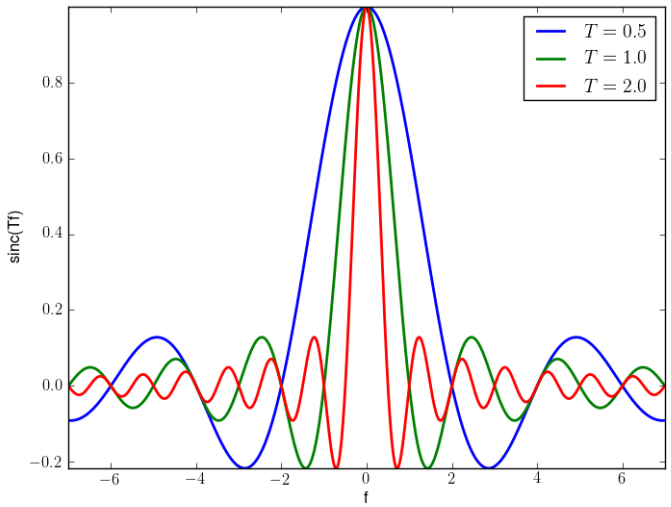
$$b_T(t) = \begin{cases} 1 & \text{if } -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{else} \end{cases}$$

- Its Fourier transform is

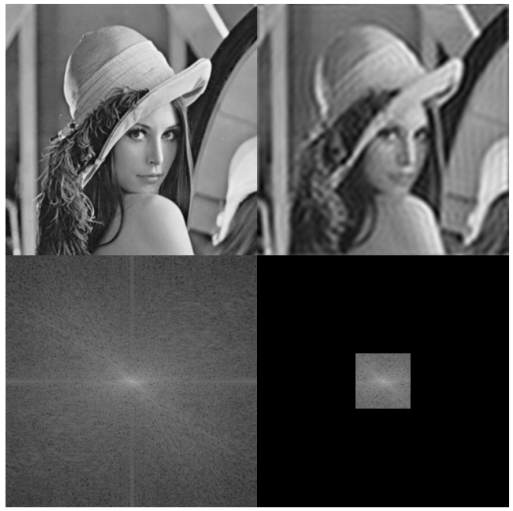
$$\begin{aligned} B_T(f) &= \int_{-\infty}^{\infty} b(t) e^{-i2\pi ft} \, \mathrm{d}t \\ &= \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{-i2\pi ft} \, \mathrm{d}t \propto \frac{\sin(\pi Tf)}{\pi Tf} = \mathrm{sinc}(Tf) \end{aligned}$$

- Similarly, the impulse response corresponding to a frequency "box" response is a sinc function, too.

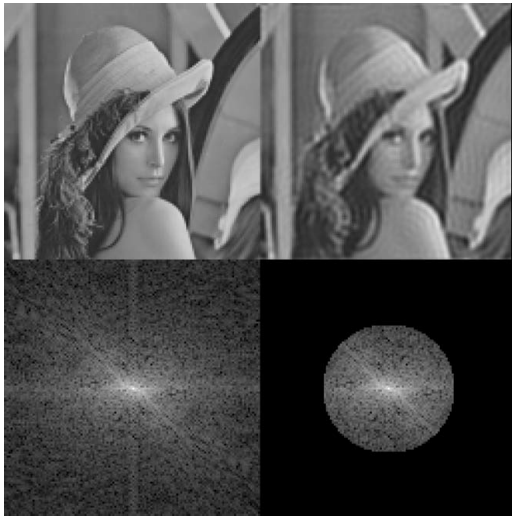
Sinc function $\sin(x)/x$



Ringing



Ringing



Filterdesign

Filterdesign: “window method”

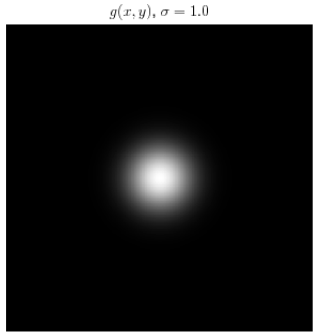
1. Specify desired ideal filter H_{id}, as well as *tolerances* in the frequency domain.
2. Compute the inverse Fouriertransform h_{id} of the desired filter. This will in general not be finite. Therefore:
3. Multiply h_{id} with a window w. Windows have been computed analytically and are tabulated wrt. to their tolerances in the frequency domain.
4. Implement the filter by convolving with mit $h_{\mathrm{id}} \cdot w$

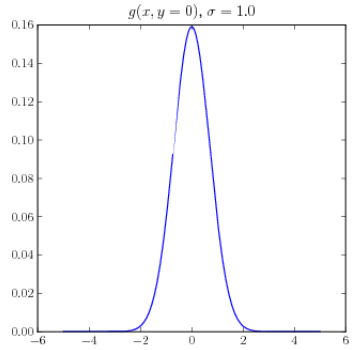
Avoiding sharp transitions

- One can relate the problem of ringing also to the presence of sharp edges in the filter.
- Discontinuities in the signal require the presence of many spectral components (and vice versa).
- An informal rule of thumb for designing filters is therefore to avoid sharp edges.

Gauss filter

$$g(x,y) = \frac{1}{2\pi\sigma} \exp\left(-\frac{1}{2\sigma^2}(x^2+y^2)\right)$$





Gauss filter

The Fourier transform of a Gaussian is a Gaussian.

- The standard deviations are related by $\sigma_{\text{spectrum}} = \frac{1}{\sigma}$
- In practice: Use a sampled Gauss function or some other discrete approximation to the Gauss function.

Example

Gauss filter, $\sigma = 1.0$

Gauss filter, $\sigma = 2.0$

Gauss filter, $\sigma = 3.0$

Gauss filter, $\sigma = 4.0$

Gauss filter, $\sigma = 5.0$

Gauss filter, $\sigma = 10.0$