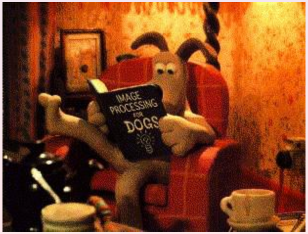
Introduction to Image Processing #5/7

Thierry Géraud

EPITA Research and Development Laboratory (LRDE)



2006

Outline

Outline

Outline

Outline

Outline

Outline

Filtering

Filtering images

- is a low-level process
- can be linear or not (!)
- is often useful
  - either to get “better” data
    e.g., with enhanced contrast, less noise, etc.
  - or to transform data to make it suitable for further processing

New Approach

An image is a function.

We have

- **sampling**: image values are only known at given points I_p with $p = (r, c) \in \mathbb{N}^2$
- **quantization**: image values belong to a restricted set for instance $[0, 255]$ for a gray-level with 8 bit encoding

An image is a *digital* signal
(contrary: *analog*).

Background

This lecture background is

digital signal processing.

Outline

Dirac Delta Function (1/2)

The Dirac delta function, denoted by $\uparrow$, is defined by:

$$\int_{-\infty}^{\infty} s(t)\uparrow(t)\,dt \;=\; s(0)$$

where s is a test function.

Dirac Delta Function (2/2)

Please note that:

- $\uparrow$ is *not* a function,
- it is a *distribution* (or generalized function).

We have:

$$\int_{-\infty}^{\infty} \uparrow(t)\, dt \; = \; 1.$$

Weird Definition (1/2)

Just *think* of $\uparrow$ being something like:

$$\uparrow(t) = \begin{cases} 1 \times \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0. \end{cases}$$

but it cannot be a proper definition!

Weird Definition (2/2)

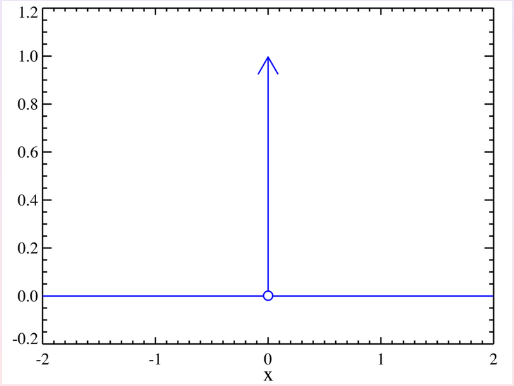
Here α is a constant in $\mathbb{R}$ or $\mathbb{C}$.

We do *not* have $\alpha\uparrow = \uparrow$, but:

$$\uparrow(t) = \begin{cases} \alpha \times \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0. \end{cases}$$

Indeed we should have $\int_{-\infty}^{\infty} \alpha\uparrow(t)dt$ being equal to α.

Dirac Representation



Outline

Foreword

Understand that the Dirac delta function is the *most* important distribution we can think of.

Sinc (1/2)

The (unnormalized, historical, mathematical) sinc function is:

$$sinc(t) = \frac{\sin(t)}{t}.$$

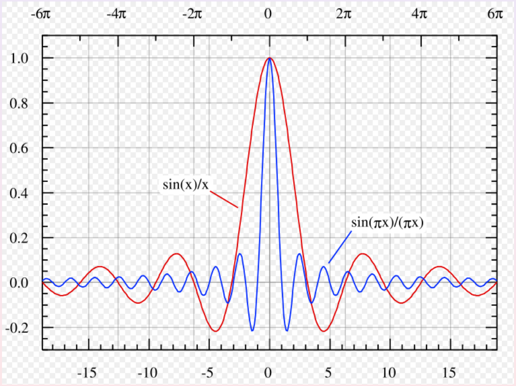
The normalized sinc function is:

$$sinc(t) = \frac{\sin(\pi t)}{\pi t}.$$

so that:

$$\int_{-\infty}^{\infty} sinc(t)\, dt = 1.$$

Sinc (2/2)

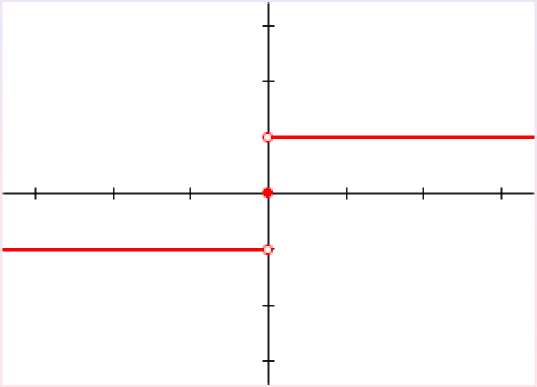

-6π
-4π
-2π
0
2π
4π
6π
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
sin(x)/x
sin(πx)/(πx)
-15
-10
-5
0
5
10
15

Sign Function (1/2)

The sign function is:

$$sgm(t) = \begin{cases} -1 & \text{if } t < 0 \\ 0 & \text{if } t = 0 \\ 1 & \text{if } t > 0. \end{cases}$$

Sign Function (2/2)



Heaviside Step Function (1/2)

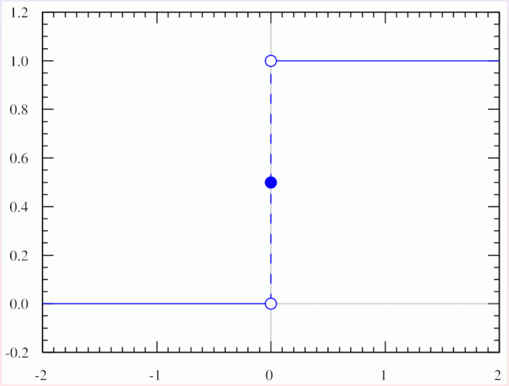
$$u(t) = \frac{1}{2}(1 + sgn(t)) = \begin{cases} 0 & \text{if } t < 0 \\ 1/2 & \text{if } t = 0 \\ 1 & \text{if } t > 0. \end{cases}$$

We have:

$$u(t) = \int_{-\infty}^{t} \uparrow(\tau)\, d\tau.$$

Put differently $\dot{u} = \uparrow$.

Heaviside Step Function (2/2)



Rect (1/2)

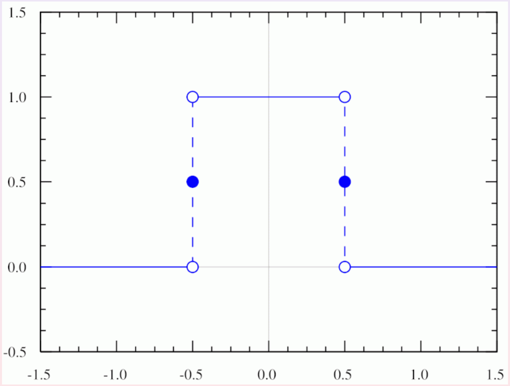
The rectangular function is:

$$rect(t) = \begin{cases} 1 & \text{if } |t| < 1/2 \\ 1/2 & \text{if } |t| = 1/2 \\ 0 & \text{if } |t| > 1/2. \end{cases}$$

We have:

$$rect(t) = u(t + 1/2)\, u(1/2 - t).$$

Rect (2/2)

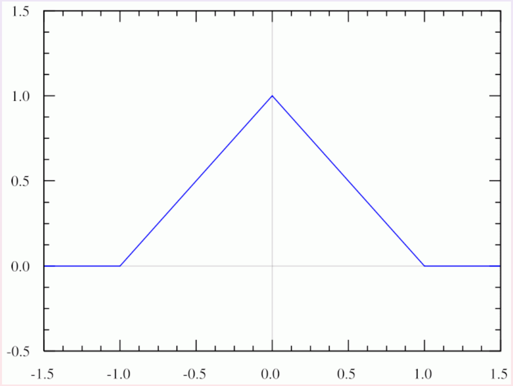


Tri (1/2)

The triangular function is:

$$tri(t) = \begin{cases} 1 - t & \text{if } |t| < 1 \\ 0 & \text{if } |t| \geq 1. \end{cases}$$

Tri (2/2)



Dirac Delta Function as a Limit

We can define ↑ as a limit of functions d_α in the sense that:

$$\lim_{\alpha \to 0} \int_{-\infty}^{\infty} s(t)\, d_\alpha(t)\, dt \;=\; s(0).$$

We can choose d_α in:

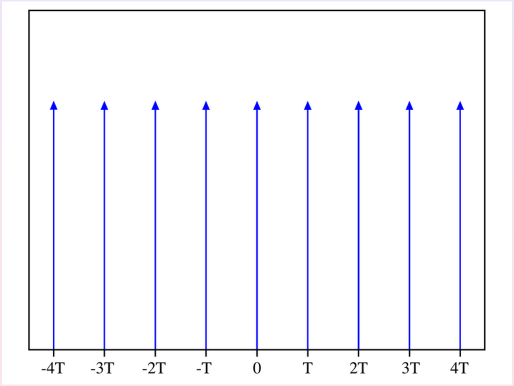
$t \longrightarrow \mathcal{N}(0;\alpha)(t)$	normal distribution
$t \longrightarrow rect(t/\alpha)/\alpha$	rectangular function
$t \longrightarrow tri(t/\alpha)/\alpha$	triangular function
...	...

Dirac Comb (1/2)

The Dirac comb is:

$$\upuparrows_T(t) = \sum_{k=-\infty}^{\infty} \uparrow(t - kT).$$

Dirac Comb (2/2)



URLs (1/2)

- Distribution

  `http://en.wikipedia.org/wiki/Distribution_(mathematics)`
- Dirac delta

  `http://en.wikipedia.org/wiki/Dirac_delta_function`
- Dirac comb

  `http://en.wikipedia.org/wiki/Dirac_comb`

URLs (2/2)

- Sign

  `http://en.wikipedia.org/wiki/Sign_function`
- Sinc

  `http://en.wikipedia.org/wiki/Sinc_function`
- Tri

  `http://en.wikipedia.org/wiki/Triangularfunction`
- Rect

  `http://en.wikipedia.org/wiki/Rectangular_function`
- Heaviside step

  `http://en.wikipedia.org/wiki/Heaviside_step_function`

Fourier Series

You know that:

$$\begin{aligned} s(x) &= \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx)) \\ &= \sum_{n=-\infty}^{\infty} S_n\, e^{inx}. \end{aligned}$$

its generalization is...

Discrete Fourier Transform

...the discrete Fourier transform of a discrete function s:

$$\begin{aligned} s_k &= \sum_{n=0}^{N-1} a_n \, e^{-i2\pi nk/N} \\ \text{where } \; a_n &= \frac{1}{N} \sum_{k=0}^{N-1} f_k \, e^{i2\pi nk/N} \end{aligned}$$

Fourier Transform

In the continuous case, S is the Fourier transform of s:

$$\begin{aligned} S(f) &= \int_{-\infty}^{\infty} s(t)\, e^{-i2\pi ft}\, dt \\ s(t) &= \int_{-\infty}^{\infty} S(f)\, e^{i2\pi ft}\, df. \end{aligned}$$

where f is a frequency.

Notations

We will denote with capital letters the Fourier transforms.

Considering that $\mathcal{F}$ is the Fourier operator on the set of complex-valued functions:

$$S = \mathcal{F}(s).$$

Some Nice Properties

Parseval's theorem:

$$\int_{-\infty}^{\infty} |s(t)|^2 \, dt \; = \; \int_{-\infty}^{\infty} |S(f)|^2 \, df \, .$$

$$\begin{aligned} \mathcal{F}^2(s)(t) &= s(-t) \\ \mathcal{F}^* &= \mathcal{F}^{-1} \end{aligned}$$

Amazing:

$$g(t) \; = \; \alpha e^{-(t-\beta)^2/2}$$

is an eigenvalue of $\mathcal{F}$.

Convolution (1/2)

The convolution of two functions *h* and *s* is the function:

$$s'(t) \;=\; h(t) * s(t) \;=\; \int_{-\infty}^{\infty} h(\tau)\, s(t-\tau)\, d\tau.$$

The convolution operator is denoted by “$*$”.

Convolution (2/2)

FIXME: figure.

Exercise

Show that:

$$\mathit{rect} \, * \, \mathit{rect} \; = \; \mathit{tri}\,.$$

Convolution Properties

We have:

commutativity	$a * b = b * a$
associativity	$a * (b * c) = (a * b) * c$
distributivity	$a * (b + c) = (a * b) + (a * c)$
associativity with scalar	$\alpha(b * c) = (\alpha b) * c = b * (\alpha c)$

Differentiation rule:

$$\mathcal{D}(a * b) = \mathcal{D}(a) * b = a * \mathcal{D}(b).$$

Dirac and Convolution (1/2)

We have:

$$(\uparrow * s)(t) \;=\; \int_{-\infty}^{\infty} \uparrow(\tau)\, s(t-\tau)\, d\tau \;=\; s(t) \;\; \forall t.$$

So $\uparrow * s \;=\; s$:

$\uparrow$ is the neutral element for the $*$ operator.

Dirac and Convolution (2/2)

Let us note by:

$$\uparrow_{t'}(t) \;=\; \uparrow(t - t')$$

the translation to t' of the Dirac delta function
put differently it is a Dirac impulse centered at t'

We have:

$$\begin{aligned}(\uparrow_{t'} * s)(t) &= \int_{-\infty}^{\infty} \uparrow(\tau - t')\, s(t - \tau)\, d\tau \\ &= s(t - t').\end{aligned}$$

Convolving a function with a Dirac delta function centered at t' means *shifting* this function at the value $t = t'$.

Convolution and Fourier = a Theorem

The convolution theorem is:

$$\mathcal{F}(a * b) \;=\; \mathcal{F}(a)\,\mathcal{F}(b)$$

We also have:

$$\mathcal{F}(a\,b) \;=\; \mathcal{F}(a) \,*\, \mathcal{F}(b)$$

Dirac and Fourier

$$\int_{-\infty}^{\infty} 1(t)\, e^{-i2\pi ft}\, dt \;=\; \uparrow(f).$$

Some Fourier Transforms (1/2)

	$\mathcal{F}$
$rect(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} sinc(\frac{f}{\alpha})$
$sinc(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} rect(\frac{f}{\alpha})$
$sinc^2(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} tri(\frac{f}{\alpha})$
$tri(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} sinc^2(\frac{f}{\alpha})$
$1(t)$	$\uparrow(f)$
$\uparrow(t)$	$1(f)$
$\cos(\alpha t)$	$1/2(\uparrow(f - \frac{\alpha}{2\pi}) + \uparrow(f + \frac{\alpha}{2\pi}))$

Some Fourier Transforms (2/2)

A remarkable Fourier transform:

$$\mathcal{F}(ШШ_T)(t) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f - k/T) = \frac{1}{T}\, ШШ_{1/T}(f).$$

URLs

- Fourier series
  http://en.wikipedia.org/wiki/Fourier_series
- Discrete Fourier transform
  http://en.wikipedia.org/wiki/Discrete_fourier_transform
- Fourier transform
  http://en.wikipedia.org/wiki/Fourier_transform
- Parseval's theorem
  http://en.wikipedia.org/wiki/Parseval's_theorem
- Convolution
  http://en.wikipedia.org/wiki/Convolution
- Convolution theorem
  http://en.wikipedia.org/wiki/Convolution_theorem

Analog Function

Consider an analog function:

$$s : \begin{cases} \mathbb{R} & \longrightarrow \mathbb{R} \\ t & \longmapsto s(t) \end{cases}$$

FIXME: figure.

From Analog to Digital (1/2)

A discrete function s_d is sampled from s with the sampling frequency $f_d = 1/T$:

$$\begin{aligned} s_d(t) &= \textstyle\sum_{k=-\infty}^{\infty} s(kT)\uparrow(t-kT) \\ &= s(t) \times \Uparrow_T(t). \end{aligned}$$

So:

$$\begin{aligned} S_d(f) &\propto S(f) * \Uparrow_{f_d}(f) \\ &\propto S(f) * \textstyle\sum_{k=-\infty}^{\infty} \uparrow(f-kf_d) \\ &\propto \textstyle\sum_{k=-\infty}^{\infty} S(f) * \uparrow(f-kf_d) \\ &\propto \textstyle\sum_{k=-\infty}^{\infty} S(f-kf_d) \end{aligned}$$

From Analog to Digital (2/2)

FIXME: figure.

From Digital to Analog (1/2)

An analog function s_a is reconstructed from the digital one s_d:

$$S_a(f) = S_d(f) \times rect(\frac{f}{2f_d}).$$

So:

$$\begin{aligned} s_a(t) &\propto s_d(t) * sinc(t/T) \\ &\propto \left(\sum_{k=-\infty}^{\infty} s(kT)\uparrow(t-kT)\right) * sinc(t/T) \\ &\propto \sum_{k=-\infty}^{\infty} (s(kT)\uparrow(t-kT) * sinc(t/T)) \\ &\propto \sum_{k=-\infty}^{\infty} \left(s(kT)\, sinc(\tfrac{t-kT}{T})\right) \end{aligned}$$

From Digital to Analog (2/2)

FIXME: figure.

Shanon Sampling Theorem (1/2)

The minimum sampling frequency to be able to perfectly reconstruct an analog signal is twice the maximum signal frequency.

So we should have:

$$f_d > 2\, f_{\max}.$$

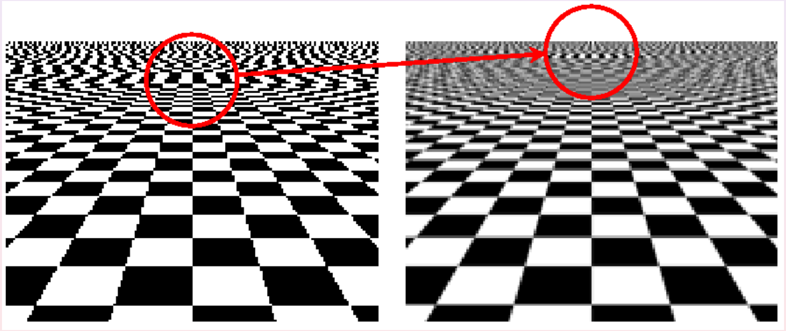
Practically this condition is (usually) never satisfied.

Shanon Sampling Theorem (2/2)

FIXME: figure.

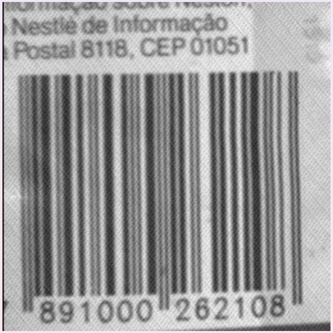
Aliasing

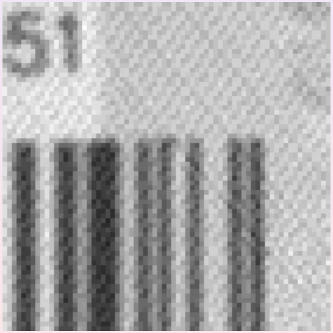
The right image is anti-aliased:



Removing the Moire Effect (1/3)

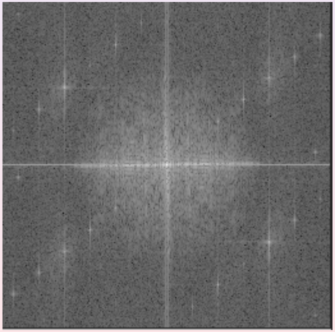
An image with aliasing presence (left) and a detail (right):

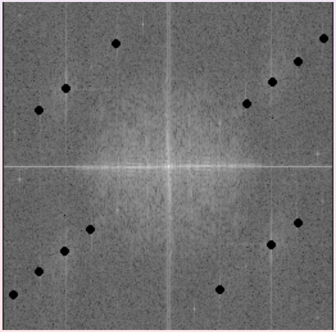




Removing the Moire Effect (2/3)

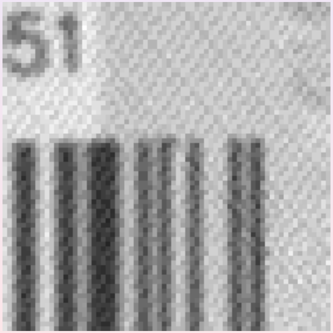
The original Fourier spectrum (left) and after removing folded peaks (right):

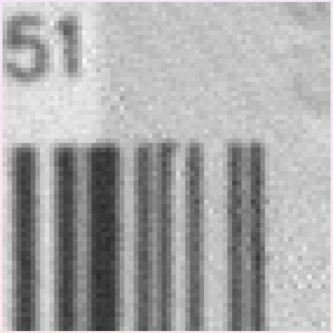




Removing the Moire Effect (3/3)

The result (right) compared to the original (left):





URLs

- Signal processing

  `http://en.wikipedia.org/wiki/Signal_processing`
- Sampling

  `http://en.wikipedia.org/wiki/Sampling_(signal_processing)`
- Shannon's sampling theorem

  `http:`
  `//en.wikipedia.org/wiki/Nyquist-Shannon_sampling_theorem`
- Aliasing

  `http://en.wikipedia.org/wiki/Aliasing`
- Anti-aliasing

  `http://en.wikipedia.org/wiki/Anti-aliasing_filter`

Discrete Convolution

The convolution between *a* and *b*:

$$s'(t) \;=\; h(t) * s(t) \;=\; \int_{-\infty}^{\infty} h(\tau)\, s(t-\tau)\, d\tau.$$

can be turned into a discrete convolution:

$$s'[t] \;=\; (h * s)[t] \;=\; \sum_{\tau=-\infty}^{\infty} h[\tau]\, s[t-\tau] \;=\; \sum_{\tau=-\infty}^{\infty} h[t-\tau]\, s[\tau]$$

where t and τ are now $\in \mathbb{Z}$.

Discrete Convolution in 2D (1/2)

$$\begin{aligned}
s'[r][c] &= (h * s)[r][c] \\
&= \sum_{r'=-\infty}^{\infty} \sum_{c'=-\infty}^{\infty} h[r'][c']\, s[r-r'][c-c'] \\
&= \sum_{r'=-\infty}^{\infty} \sum_{c'=-\infty}^{\infty} h[r-r'][c-c']\, s[r'][c']
\end{aligned}$$

Discrete Convolution in 2D (2/2)

FIXME: figure.

Linear Filters (1/2)

A filter ϕ is linear if

$$\phi(\alpha\, s_1 \,+\, \beta\, s_2) \;=\; \alpha\, \phi(s_1) \,+\, \beta\, \phi(s_2)$$

for all α and β scalars, and s_1 and S_2 functions.

$$\phi \text{ is a linear filter } \Leftrightarrow\; h_\phi \text{ exists such as } \;\phi \;=\; h_\phi\, *$$

Put differently:

- we can write $\phi(s) \;=\; h_\phi \,*\, s$
- convolutions are the only linear filters.

Linear Filters (2/2)

Consider that a filter ϕ is a black box.

If this filter is linear, we want to know h_ϕ.

When you input the Dirac delta function (an impulse) into the black box, the resulting function (signal) is:

$$h_\phi * \uparrow \; = \; h_\phi .$$

h_ϕ is the impulse response of ϕ.

Dirac Delta Function

Before all:

$$\uparrow[r][c] \;=\; \begin{cases} 1 & \text{if } r = 0 \text{ and } c = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Outline

Gradients as Linear Filters (1/3)

Consider the gradient of a 2D function *s*:

$$\nabla s = \begin{pmatrix} \frac{\delta s}{\delta x} \\ \\ \frac{\delta s}{\delta y} \end{pmatrix}$$

The gradient is linear; with $\phi_\nabla(s) = \nabla s$, we have:

$$\phi_\nabla(\alpha s_1 + \beta s_2) = \alpha\, \phi_\nabla(s_1) + \beta\, \phi_\nabla(s_2)$$

...so it can be expressed with convolutions!

Gradients as Linear Filters (2/3)

Assuming that sampling is isotropic ($T_x = T_y = 1$), some discrete approximations of the gradient are:

$$\nabla s\,[r][c] \;\approx\; \nabla^{ante} s\,[r][c] \;=\; \begin{pmatrix} s[r][c] \;-\; s[r][c-1] \\ s[r][c] \;-\; s[r-1][c] \end{pmatrix}$$

or:

$$\nabla s\,[r][c] \;\approx\; \nabla^{post} s\,[r][c] \;=\; \begin{pmatrix} s[r][c+1] \;-\; s[r][c] \\ s[r+1][c] \;-\; s[r][c] \end{pmatrix}$$

and even:

$$\nabla s\,[r][c] \;\approx\; \frac{\nabla^{ante} s\,[r][c] \;+\; \nabla^{post} s\,[r][c]}{2} \;=\; \begin{pmatrix} \frac{s[r][c+1]-s[r][c-1]}{2} \\ \frac{s[r+1][c]-s[r-1][c]}{2} \end{pmatrix}$$

Gradients as Linear Filters (3/3)

With:

$$\nabla s \;=\; \left(\; h_{\nabla}^{/x} \;*\; s h_{\nabla}^{/y} \;*\; s \;\right)$$

and considering the “*post*” version:

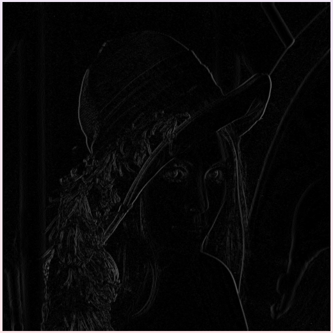
$$\begin{aligned} \nabla^{/x} s[r][c] \;&\approx\; s[r][c+1] \;-\; s[r][c] \\ &\approx\; h^{/x}[0][-1]\, s[r-0][c-(-1)] \\ &\quad + h^{/x}[0][0]\, s[r-0][c-0] \end{aligned}$$

we have:

$$h^{/x}[r][c] \;=\; \begin{cases} 1 & \text{if } r=0 \quad \text{and} \quad c=-1 \\ -1 & \text{if } r=0 \quad \text{and} \quad c=0 \\ 0 & \text{otherwise} \end{cases}$$

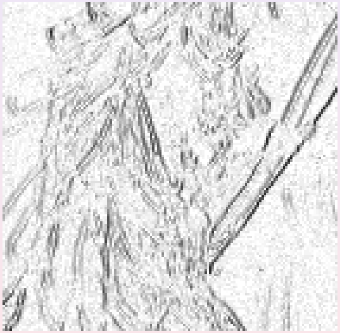
Gradients Illustrated (1/2)

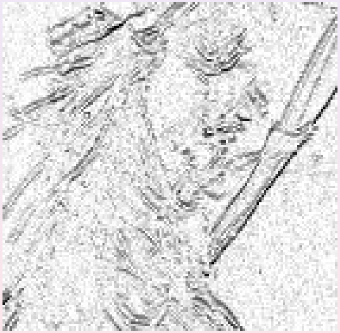
LENA (left) and the x gradient (right):



Gradients Illustrated (1/2)

crops of resp. the x gradient (left) and the y gradient (right):





Please note that, to better view images, contrast is enhanced and values are inverted (the lowest values are now the brightest).

Graphical Representation

To depict functions we use a graphical representation:

$h^{/x} =$

0	0	0
1	-1	0
0	0	0

In such representations, the origin is always centered and we do not represent null values that lay outside the window.

Gradient Magnitude (1/2)

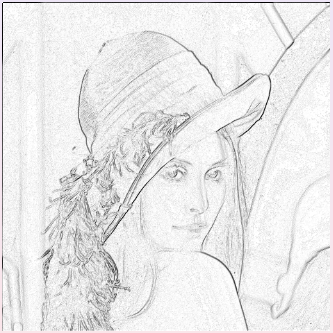
The magnitude is often approximated with a L_1 norm, so:

$$|\nabla s| = \left|\frac{\delta s}{\delta x}\right| + \left|\frac{\delta s}{\delta y}\right|.$$

For instance with the “*post*” version:

$$|\nabla^{post} s|[r][c] = |s[r+1][c] - s[r][c]| + |s[r][c+1] - s[r][c]|$$

Gradient Magnitude (2/2)



Warning: magnitude is also video inverted here.

Other Versions of Gradient Magnitude (1/3)

Many versions of the gradient magnitude exist... for instance this one (the Roberts filter):

$$|\nabla^{roberts} s|[r][c] = |s[r+1][c+1] - s[r][c]| + |s[r][c+1] - s[r+1][c]|$$

Other Versions of Gradient Magnitude (2/3)

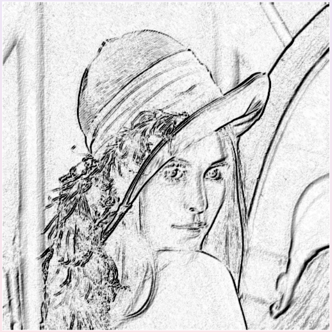
A noise-”insensitive” version of the gradient magnitude is the Sobel filter:

$$h^{/x}_{Sobel} = \frac{1}{4} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Exercise: explain why it is less sensitive to noise.

Other Versions of Gradient Magnitude (3/3)

Result of the Sobel filter:

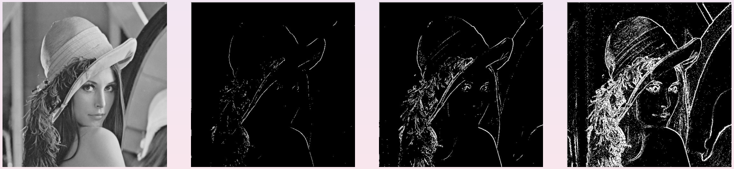


Warning: magnitude is also video inverted here.

Exercise: explain why contours/edges look thicker here than in

Extracting Object Contours (1/2)

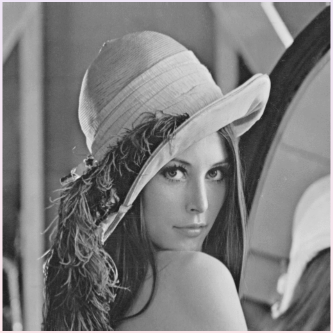
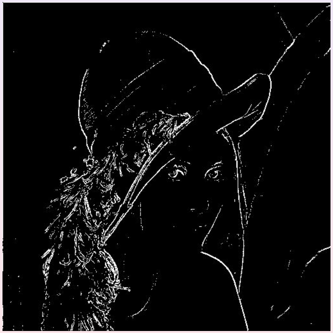
Extracting contours/edges can be performed thru thresholding the gradient magnitude:



Exercise: is it great?

Extracting Object Contours (2/2)

Full size:

Exercise: is it great?

Outline

Definition

The Laplacian of s is:

$$\Delta s = \frac{\delta^2 s}{\delta x^2} + \frac{\delta^2 s}{\delta y^2}$$

Exercise: express h_Δ.

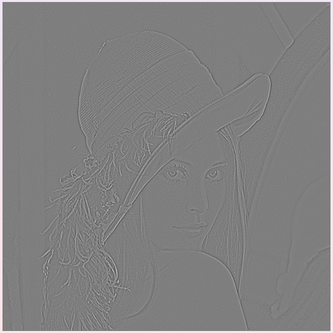
Solution

You should end up with:

$$h_{\Delta} = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

Filtering (1/2)

Result:



Filtering (2/2)

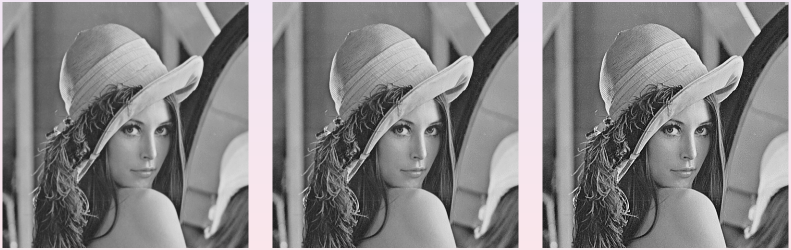
Crop:

Exercise:

- find how to sharpen image contours/edges,
- express $h_{sharpen}$ in function of a strength parameter.

Edge Sharpening (1/2)

Contour/edge sharpening results:



Edge Sharpening (2/2)

Contour/edge sharpening results: