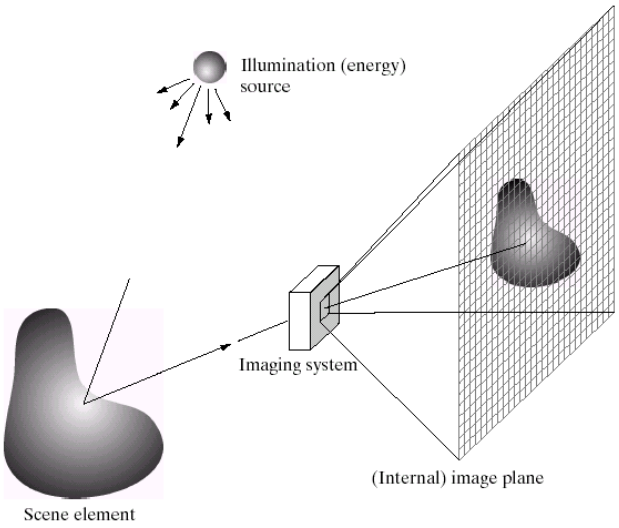
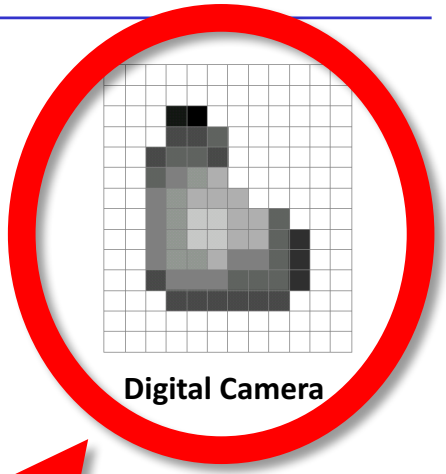
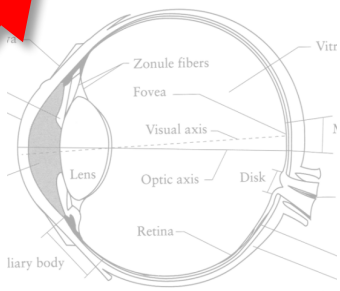
Image processing

images

what is an image?





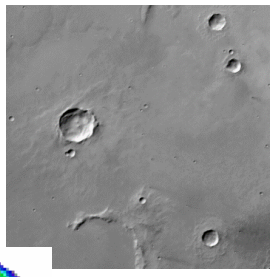


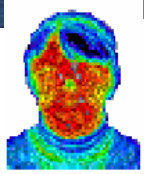
We'll focus on these in this class

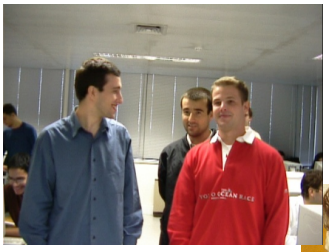
(More on this process later)

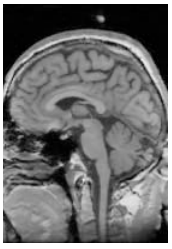
Source: A. Efros

images: what are they?

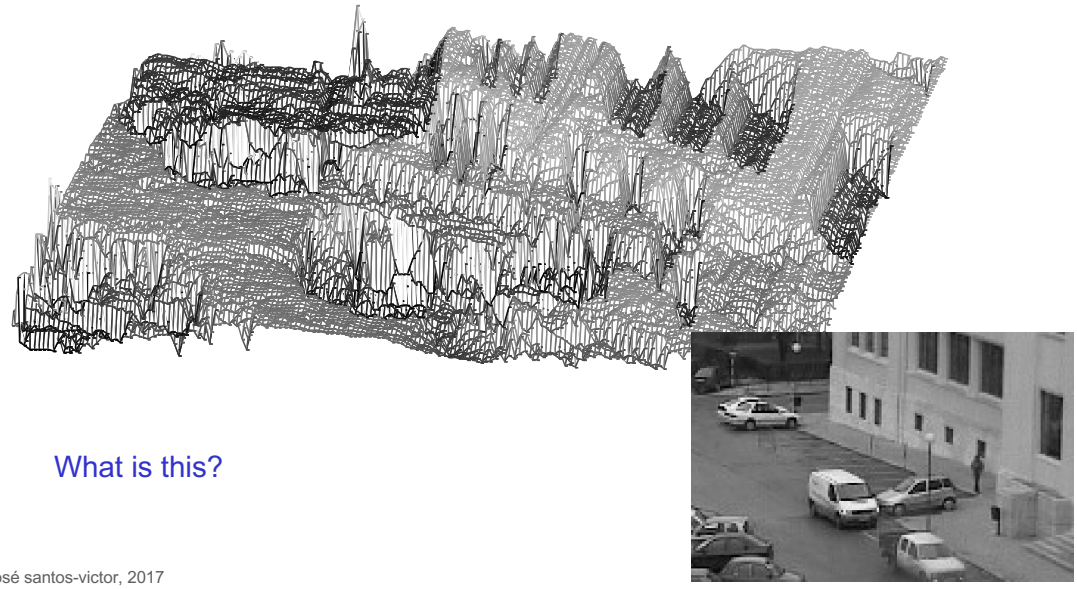








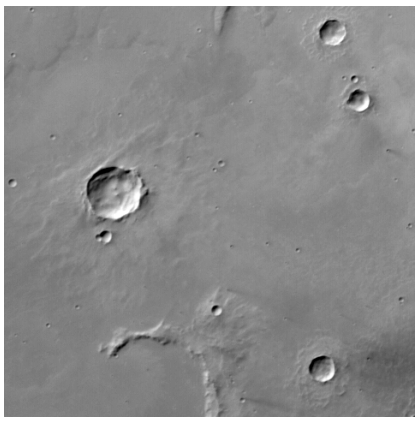
what is shown in this image?



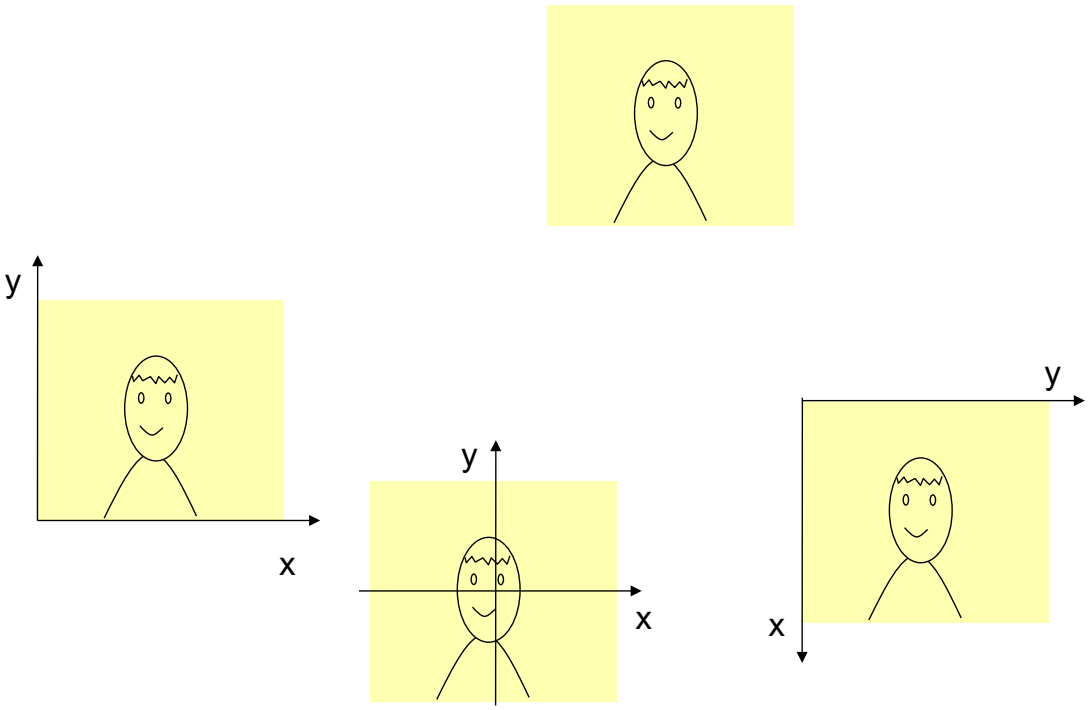
What is this?

what is an image

images describe the evolution of physical variables (intensity, color, reflectance, condutivity) in a plane or in a 3D volume.



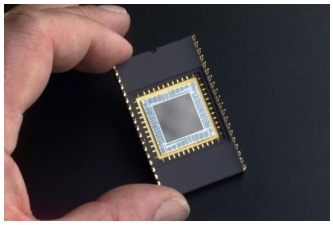
coordinate systems



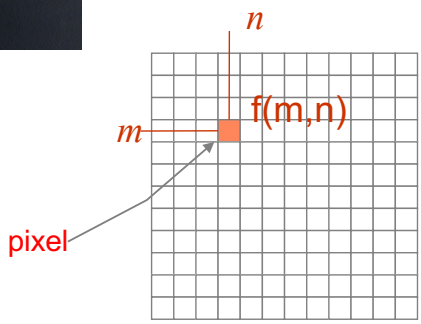
most common choice

discrete images

Most electronic sensors produce discrete images defined on a grid of points



CCD sensor



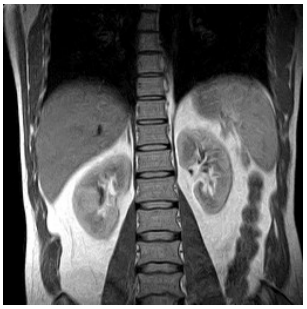
42	45	49	57	65	54	66	63	42	64	95
73	72	68	67	69	75	59	62	55	35	37
82	83	83	84	85	81	79	90	77	76	73
96	95	95	92	90	93	88	79	95	88	93
93	91	93	95	95	96	97	104	94	96	104
110	107	102	104	110	107	103	102	110	97	106
109	112	113	109	105	107	101	108	116	115	111
109	107	109	112	113	101	119	128	143	121	122
114	114	115	115	113	119	146	154	115	110	116
125	125	120	118	121	160	140	112	118	116	122
114	110	115	137	161	132	119	119	107	124	120

pixel – **pi**cture **el**ement

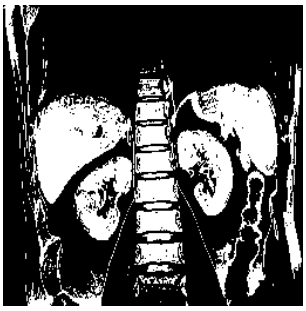
each pixel is characterized by its position and value

image types

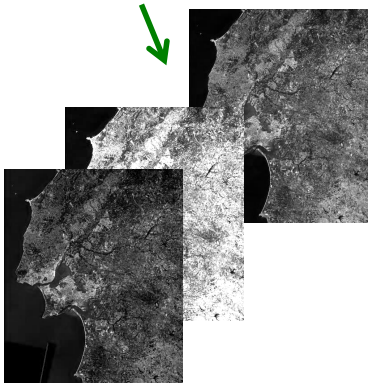
gray scale images



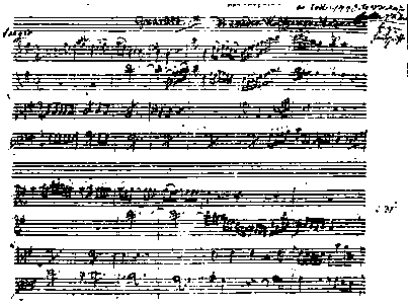
binary images



multi-spectral images

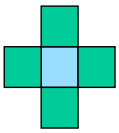


color images



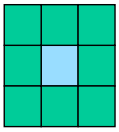
topological concepts

what is the neighborhood of a pixel*(i,j)* ?



neighborhood 4

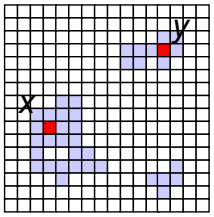
$$N_4(i,j) = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$$



neighborhood 8

$$N_8(i,j) = \{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1),$$
$$(i,j+1),(i+1,j-1),(i+1,j),(i+1,j+1)\}$$

connectivity



binary image

are x, y connected?

NO

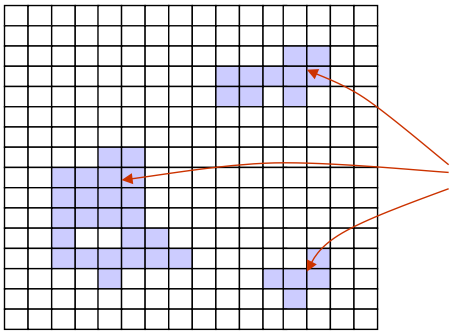
connectivity: two pixels x, y of a binary image u(m,n) are connected iif (if and only if) there a sequence of points $z^1, z^2, ..., z^n$:

$$z^1 = x, z^n = y$$

$$z^i, z^{i+1} \text{ are neighbors } \forall i \in \{1, 2, ..., n-1\}$$

$$u(z^i) = 1 \quad \forall i = 1,...,n$$

regions



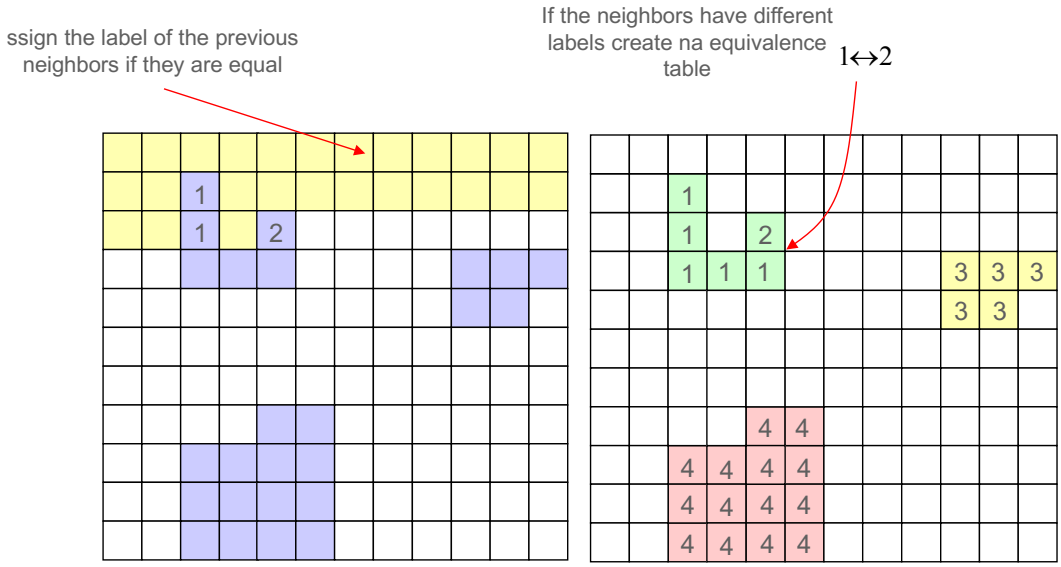
Region R is a set of fully connected pixels i.e.,

$$\forall x, y \in R, \text{ x, y are connected}$$

A maximal set of connected pixels is called a connected component.

extraction of connected components

The algorithm assigns different labels to pixels belonging to different connected components



boundary

The boundary of a region R is the set of R pixels whose neighborhood is not contained in R.

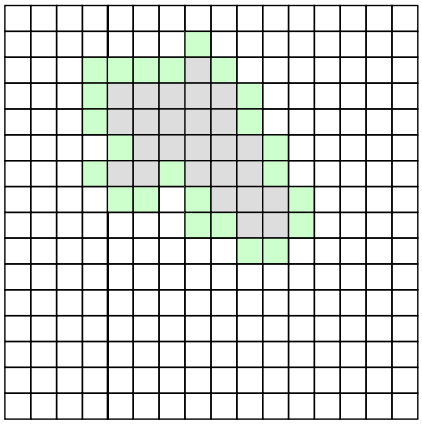


image filtering

Modify the pixels in an image based on some function of a local neighborhood of each pixel

10	5	3
4	5	1
1	1	7

Local image data

Some function

	7	

Modified image data

Source: L. Zhang

linear filtering

examples

original

blurred

sharpened

edge preserving blurred

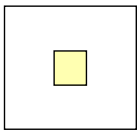
noise reduction

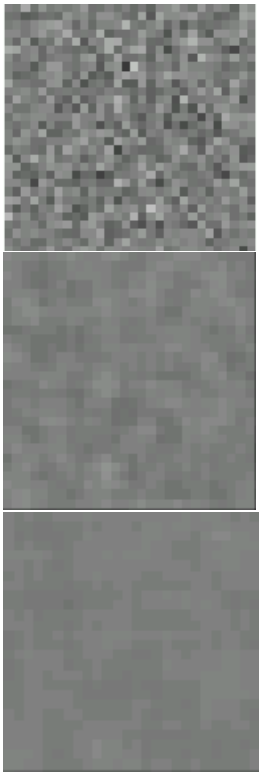
how can we remove the noise in an (homogeneous) image?

idea: replace each pixel by an average

$$g(i,j) = \frac{1}{(2N+1)^2} \sum_{k=i-N}^{i+N} \sum_{l=j-N}^{j+N} f(k,l)$$

mean filter





N=1

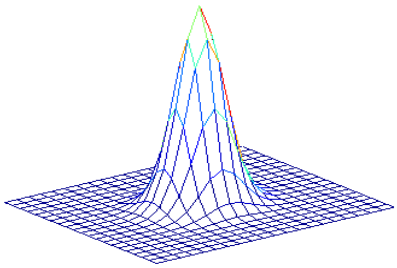
N=2

gaussian filter

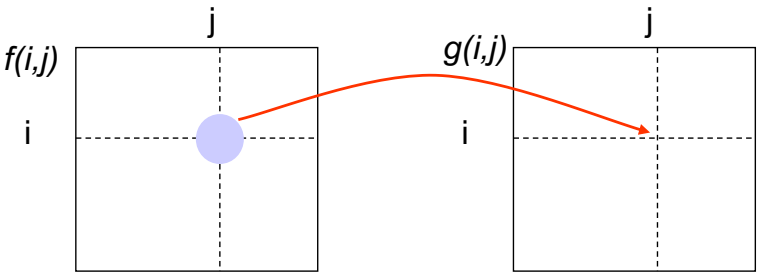
filtered image

$$g(i,j) = \sum_{k,l} G(k-i,l-j) f(k,l)$$

$$G(k,l) = C\, e^{-\frac{k^2+l^2}{2\sigma^2}}$$



weighted average



Gaussian filtering

orig.

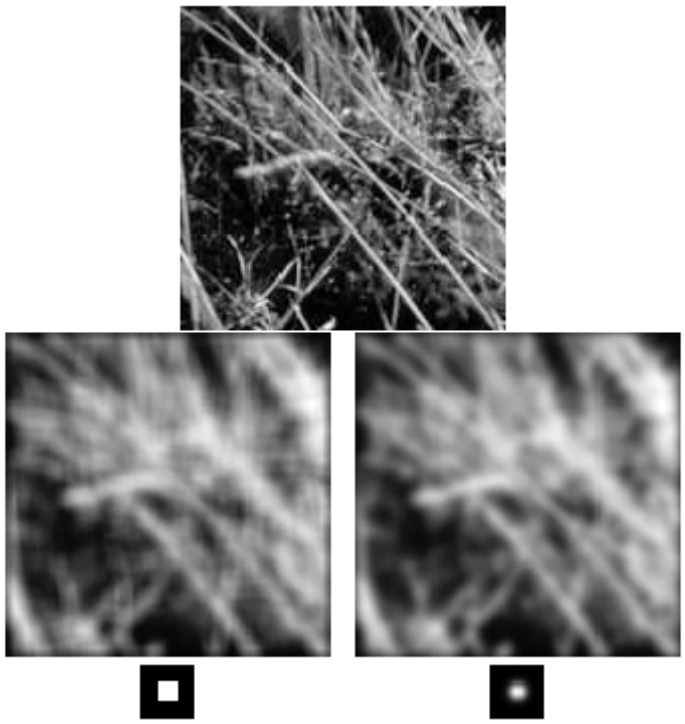
σ=0.2

σ=1

σ=2

the gaussian filter destroys the transitions!!

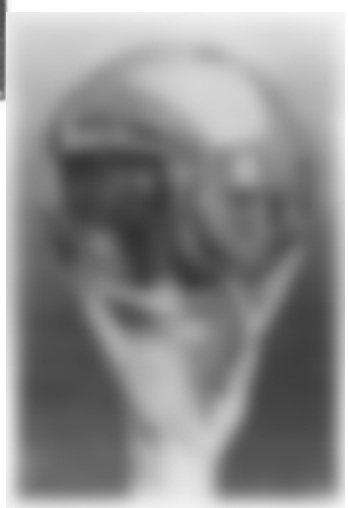
Mean vs. Gaussian filtering



$\sigma = 0$

$\sigma = 1$

$\sigma = 3$

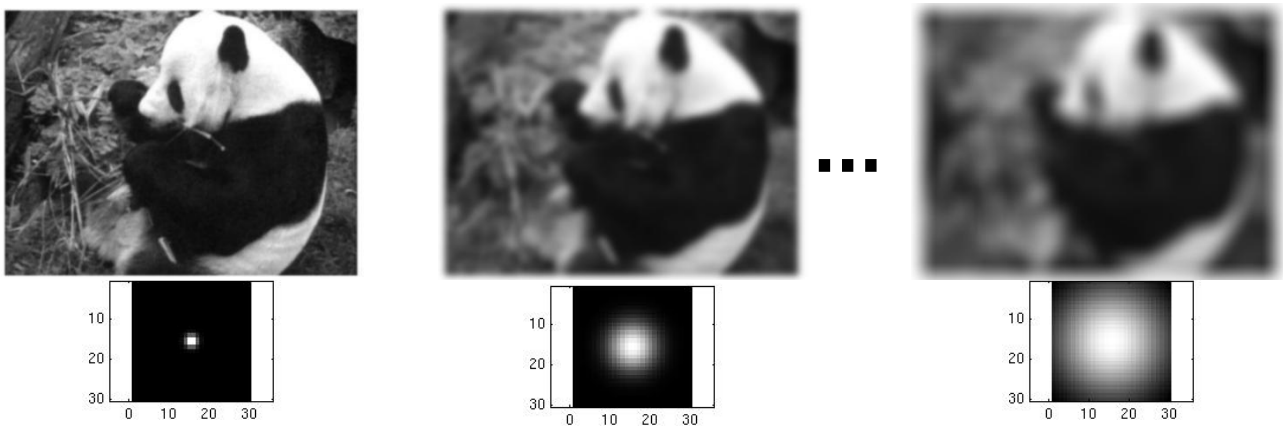


$\sigma = 6$

smoothing with a Gaussian

Recall:
parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.

Kristen Grauman, UT-Austin



linear filtering

$$g(i,j) = \sum_{k,l} M(k-i,l-j) f(k,l) = M(i,j) \otimes f(i,j)$$

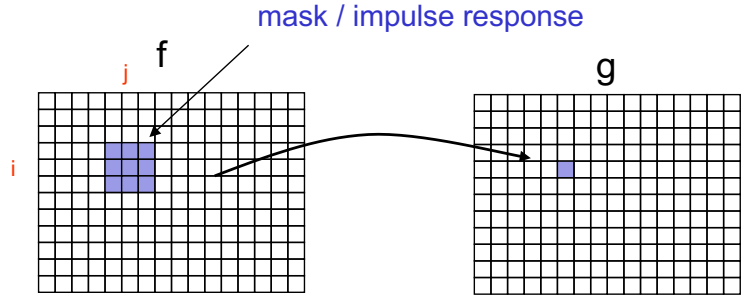
2D correlation

$M(i,j)$ mask

$$g(i,j) = \sum_{k,l} h(i-k,j-l) f(k,l) = h(i,j) * f(i,j)$$

$h(i,j) = M(-i,-j)$ impulse response

2D convolution sum



the mask/impulse response is shifted by an amount (i,j)

separable filters

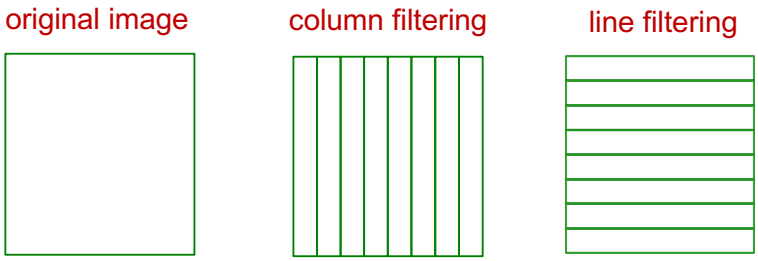
Separable filters have the following property:

$$M(i,j) = M_1(i)\, M_2(j), \quad h(i,j) = h_1(i)\, h_2(j)$$

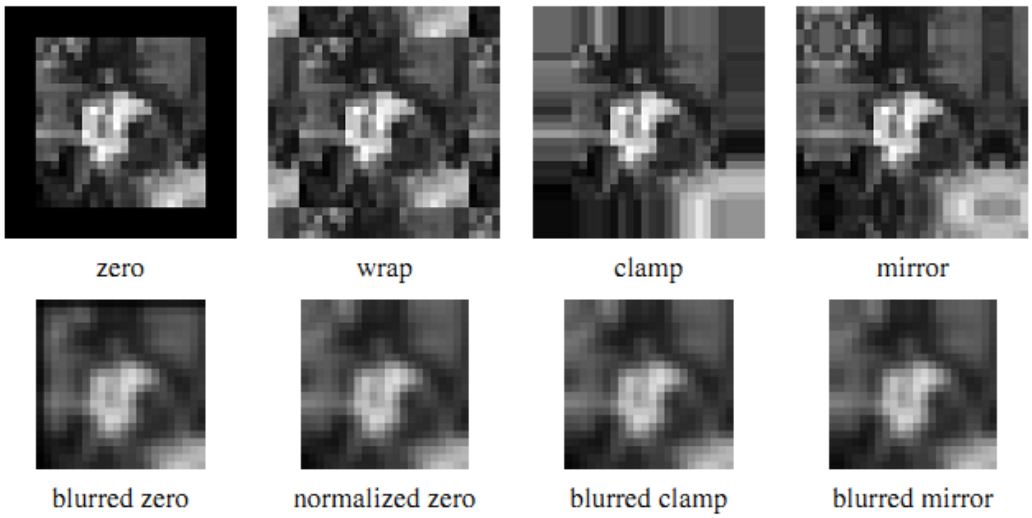
Separable filters can be implemented in a very efficient way.

Algorithm:

1. filter all the image columns using filter h_1 (1D)
2. filter all the lines of the previous image using filter h_2 (1D)



border effects



matrix notation

Linear filtering can be written in matrix notation.

$$g = Hf \qquad g, f \in IR^p \,, \quad H \in IR^{p \times p}$$

g, f are column vectors with all the pixel intensities;
H is a (huge) square matrix;
p is the number of pixels in the image.

This notation is very elegant but image filtering cannot be implemented in this way.

examples

$$M(i,j) = \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

weighted average of neighboring pixels

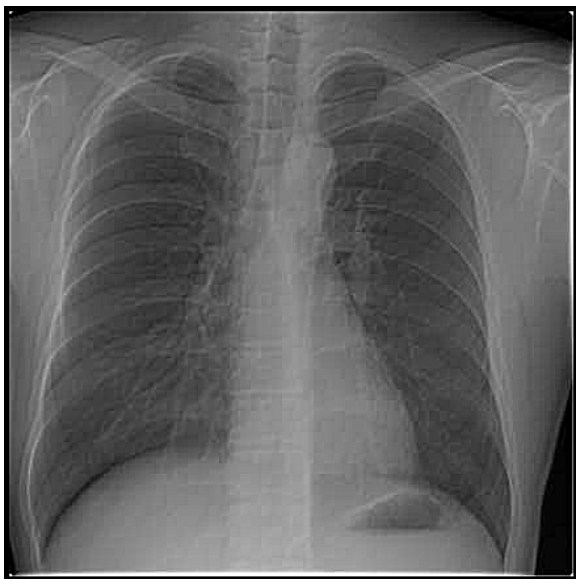
$$M(i,j) = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

vertical difference, horizontal mean

example

$$J(m,n) = h(m,n) * I(m,n) \qquad h(m,n) = \delta(m,n) - G(m,n)$$

original



filtered

1st order derivatives: gradient vector

gradient vector:

$$\nabla g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x)}{\partial x} \\ \frac{\partial f(x)}{\partial y} \end{bmatrix}$$

f(x) continuous image

$x=(x_1,x_2)$ – continuous variables

length and direction

$$|| g(x) || = \sqrt{g_1^2(x) + g_2^2(x)}$$

$$\angle g(x) = \arctan\left(\frac{g_2(x)}{g_1(x)}\right)$$

gradient discretization

1) Derivatives can be approximated by finite differences

$$g_1(m,n) \approx \frac{I(m+1,n) - I(m-1,n)}{2}$$

$$g_2(m,n) \approx \frac{I(m,n+1) - I(m,n-1)}{2}$$

-1
0
1

-1	0	1

Linear filtering!

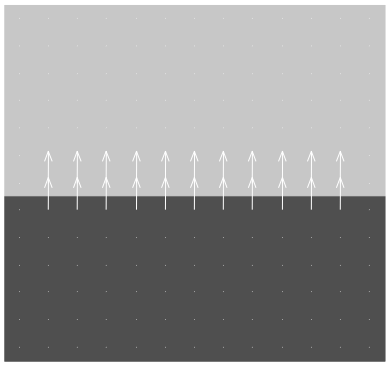
2) a better choice: Sobel masks

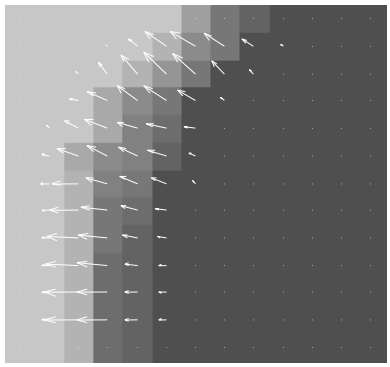
-1	-2	-1
0	0	0
1	2	1

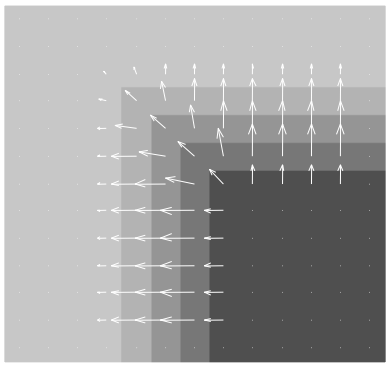
-1	0	1
-2	0	2
-1	0	1

High pass filtering in one direction, and low pass filtering in the orthogonal direction

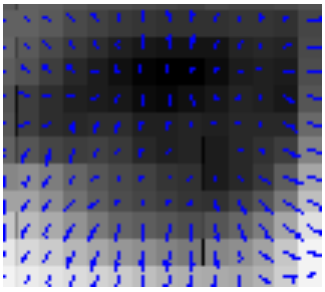
example

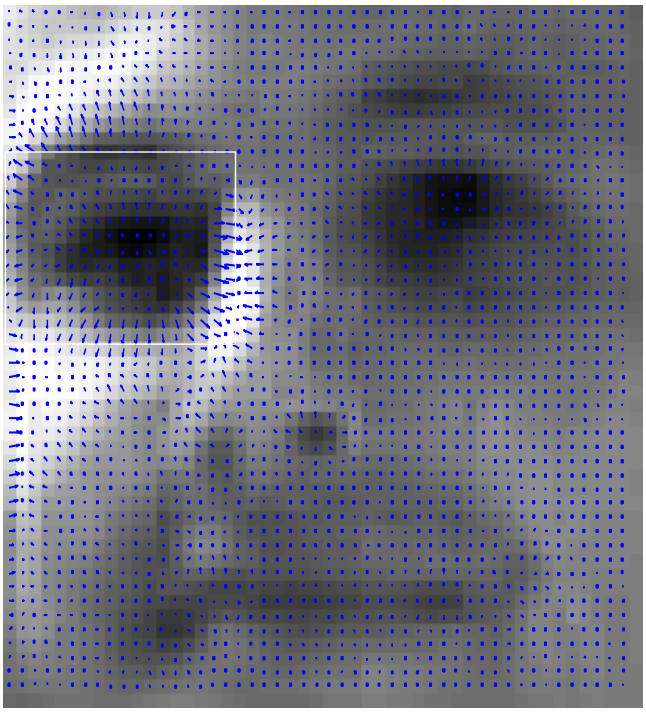






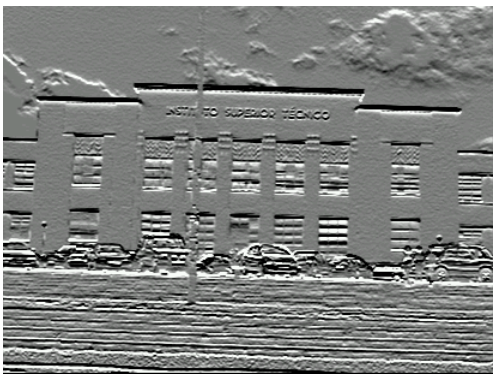
example



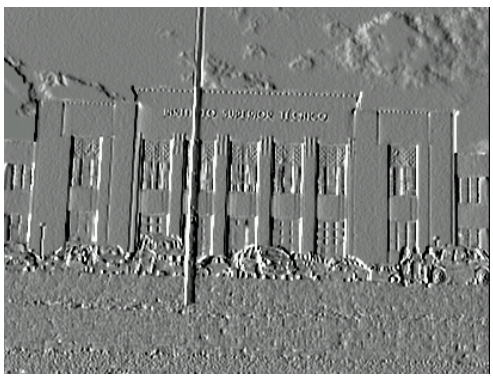


example: gradient

g_1



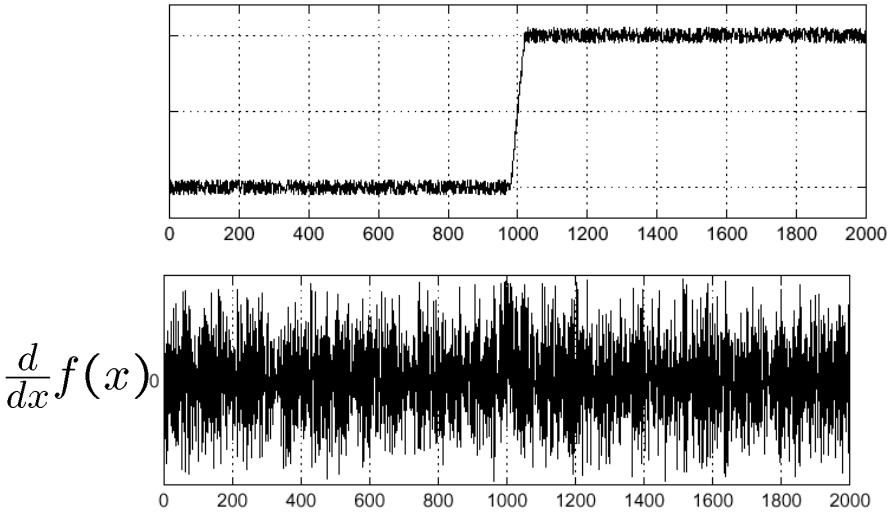
g_2



$|g|$

effects of noise

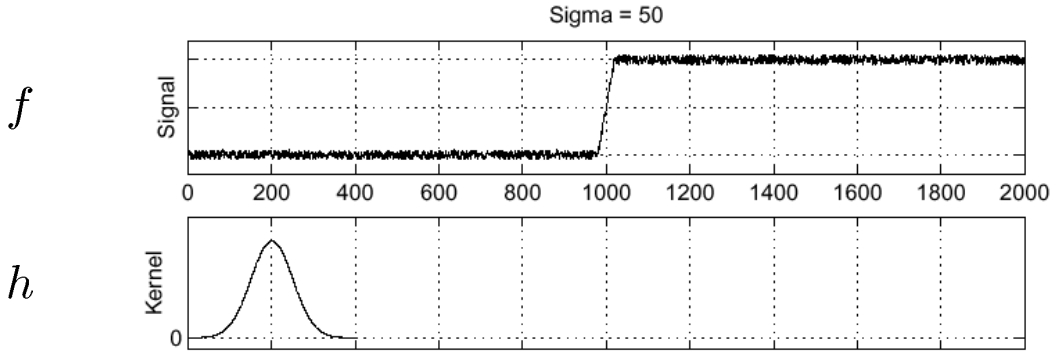
- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



Where is the edge?

Source: S. Seitz

solution: smooth first



Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Source: S. Seitz

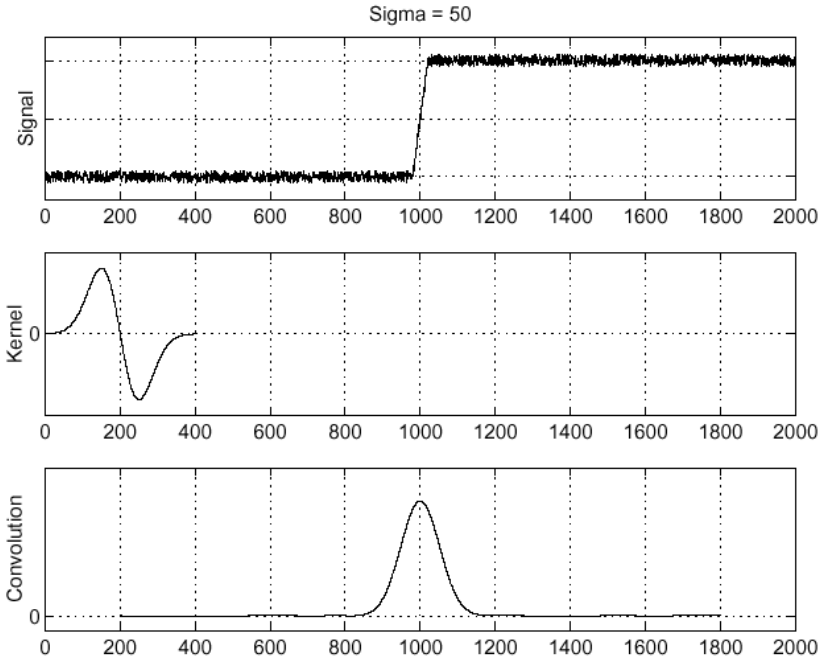
Derivative theorem of convolution

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



Slide credit Steve Seitz

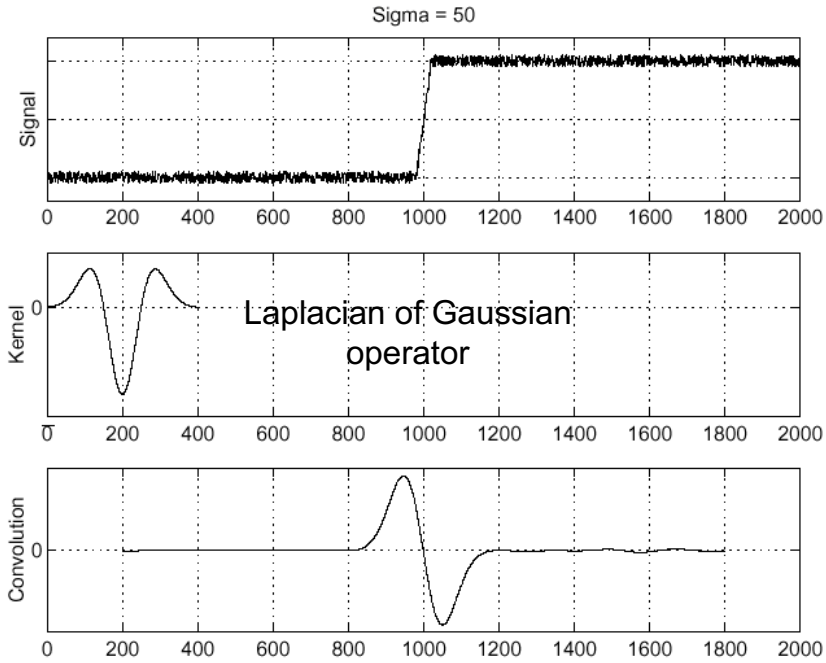
Laplacian of Gaussian

Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

$f(x)$

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



Where is the edge? Zero-crossings of bottom graph

2nd order derivatives: Laplacian filter

Laplacian of a 2D image

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

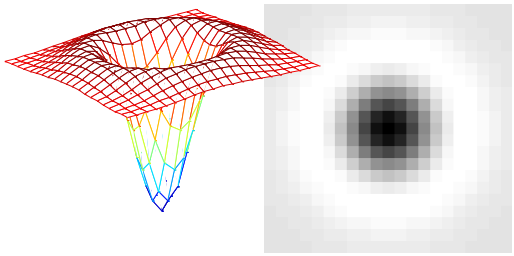
zero crossings are associated to transitions

Since the derivatives increase noise, the image is girst filtered using a Gaussian filter

$$\nabla^2 [G(x,y) * f(x,y)] = [\nabla^2 G(x,y)] * f(x,y)$$

where

$$\nabla^2 G(x,y) = \left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G(x,y)$$



Directional filters: Gabor filters

Some cells of the visual cortex are activated by directional stimula. Gabor filters approximate the behavior of these cells.

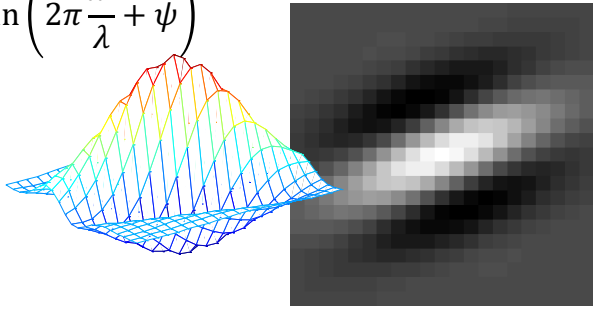
Their impulse response is the multiplication of a sinusoid by a Gaussian envelope in a rotated 2D space.

$$h_1(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \cos\left(2\pi \frac{x'}{\lambda} + \psi \right)$$

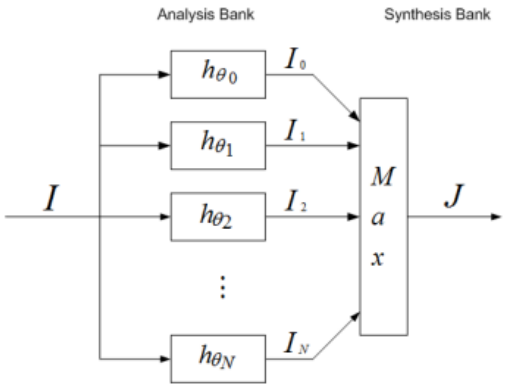
$$h_2(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \sin\left(2\pi \frac{x'}{\lambda} + \psi \right)$$

rotation

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta$$

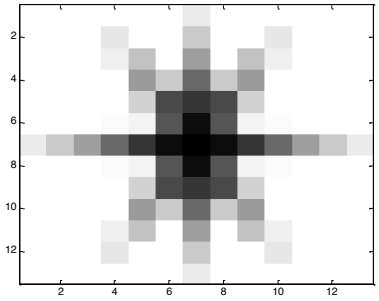


Directional analysis: difference of Gaussians

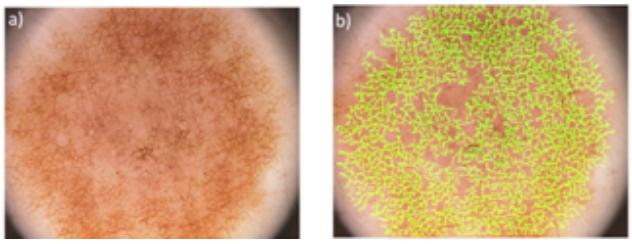


Difference of Gaussians

$$h_{\theta_i}(x,y) = G_1(x,y) - G_2(x,y)$$



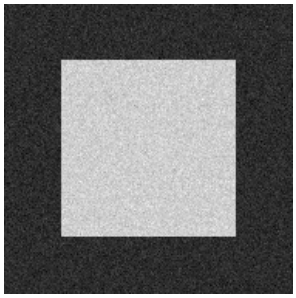
Skin lesion – dermoscopy images



Barata, Transactions on Biomedical Engineering, 2012

nonlinear filtering

linear filters reduce noise but they destroy image contours.

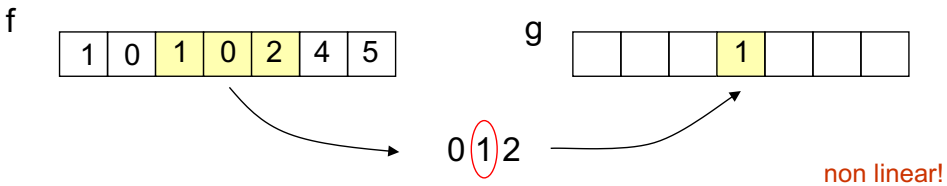


Is it possible to solve this conflict i.e., removing the noise, preserving the contours?

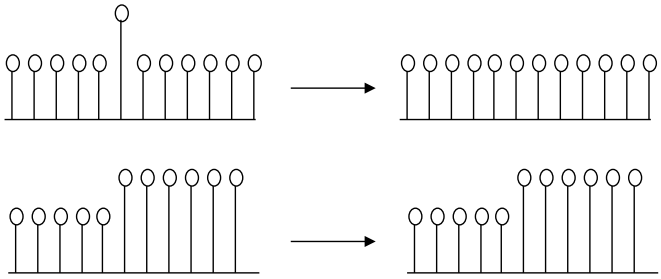
median filter …

1d median filter

$$g(n) = median\{f(n-M),\ldots,f(n+M)\}$$



example



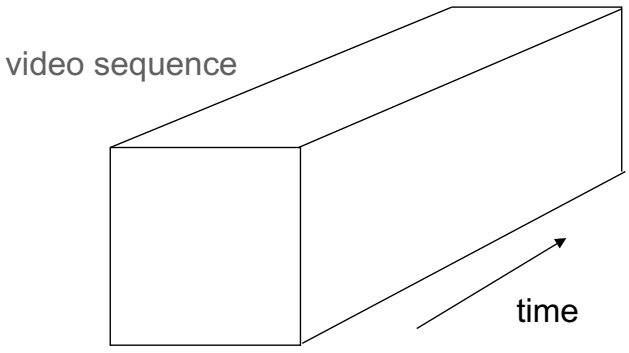
2d median filter

$$g(m,n) = median\{f(m-M,n-N),\ldots,f(m+M,n+N)\}$$

example

original image

sequence filtering



median filter can be applied in space, in time or in space-time.

example – time filtering

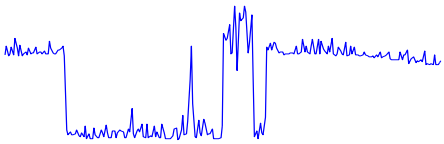


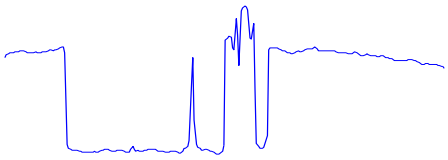



output: background image

non-linear diffusion

non linear diffusion





Perona & Malik

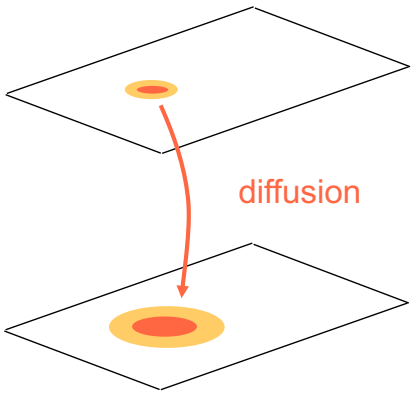
how is this possible?

heat equation

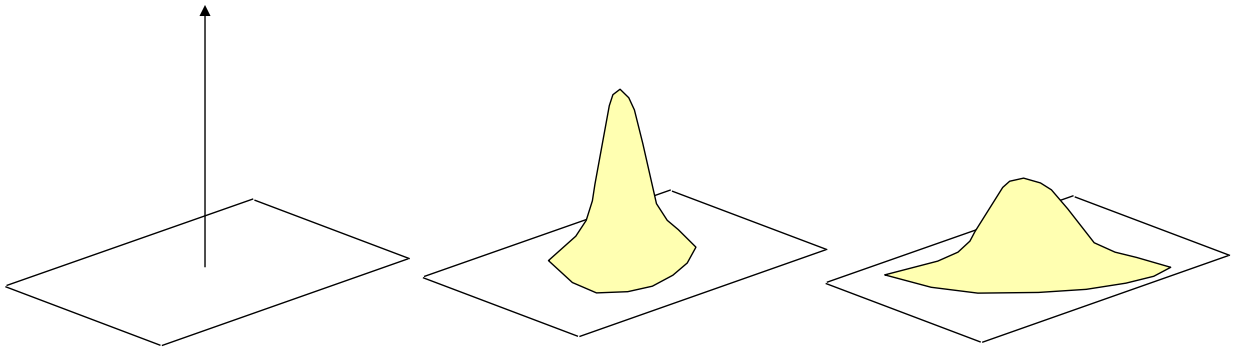
$$\frac{\partial u}{\partial t} = c\,\nabla^2 u$$

$u(x,t)$ temperature

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$



linear diffusion



$t_0=0$ $t_1>0$ $t_2>t_1$

Solution of heat equation:

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{||x||^2}{2\sigma^2}}$$

$$\sigma = \sqrt{2t}$$

(check it!)

non linear diffusion

definition: $\frac{\partial g}{\partial t} = \nabla^2 g$

$g(x,0) = f(x)$, initial condition

The equation has a unique solution

$$g(x,t) = \begin{cases} f(x) & t = 0 \\ (G_{\sqrt{2t}} * f)(x) & t > 0 \end{cases}$$

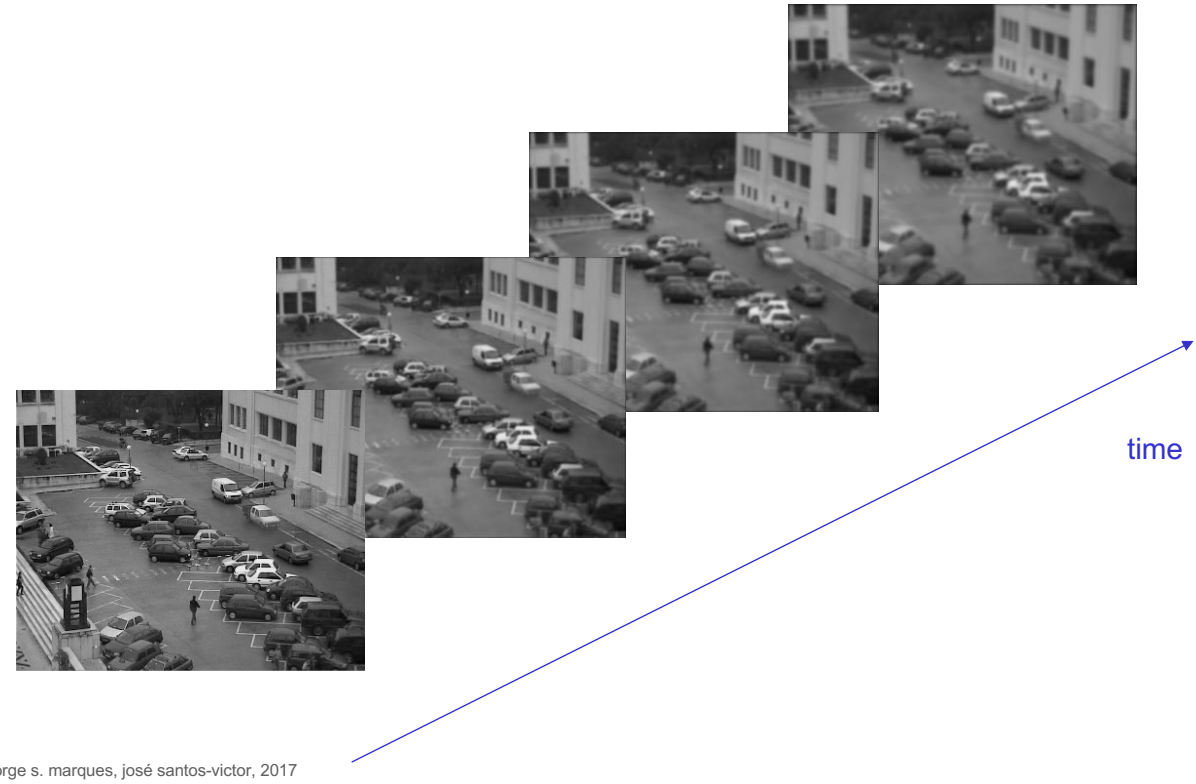
$$(G_\sigma * I)(x) = \int_{R^2} G_\sigma(x-y) f(y)\, dy$$

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{||x||^2}{2\sigma^2}}$$

Gaussian filtering beautiful !!

Linear diffusion leads to Gaussian filtering with parameter $\sigma = \sqrt{2t}$

gaussian filtering



Space-scale

Many images contain structures with different sizes. Therefore it is useful to represent it as a family of images with increasingly less details.

Space scale of an image f(x) is a family of images indexed by a parameter s

$$\{T_s f, s \geq 0\}$$

such that

$$T_0 f = f$$

$$T_{s+t} f = T_s(T_t f)$$ semi-group property

Gaussian space-scale

The Gaussian space-scale is obtained by filtering the image using a Gaussian kernel and changing the parameter s from 0 to infinity.

$$T_s f = G_s * f$$

The semi-group property holds. Prove it !

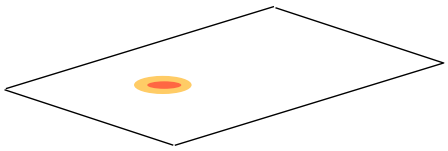
Limitations:

It destroys the contour information

how can we overcome this difficulty?

pode ser gerado através de um processo de difusão linear

Generalization



g(x) - concentration

mass flows from regions of high concentration to regions with smaller concentration taking the gradient opposite direction.

Fick's law

$$j = -c\nabla g$$

Mass conservation law

$$\frac{\partial g}{\partial t} = -div\ j$$

j(x,t) – diffusion flux
c(x,t) – diffusion coefficient
g(x,t) – concentration

homogeneous material: constant diffusivity.

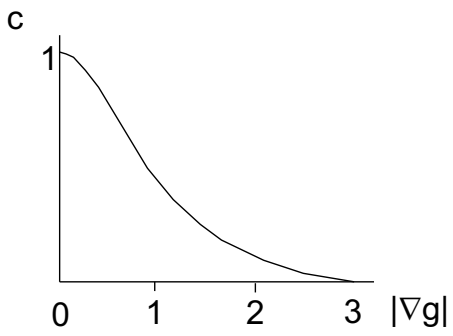
Perona & Malik: space dependent diffusivity

Idea: filter homogeneous regions and not transitions

condutivity

condutivity (Perona & Malik)

$$c(x,t) = \frac{1}{1+\alpha\|\nabla g(x,t)\|^2}$$



Condutivity depends on the concentration at a given position and time instant and changes along the time.

In homogeneous regions ($|\nabla g|=0$) and conductivity is constant c=1.
At the contour points, conductivity is almost zero.

non linear diffusion

Diffusion equation (Perona & Malik)

$$\frac{\partial g}{\partial t} = div\,(c(x,t)\nabla g)$$

$$c(x,t) = \frac{1}{1+\alpha\|\nabla g(x,t)\|^2}$$

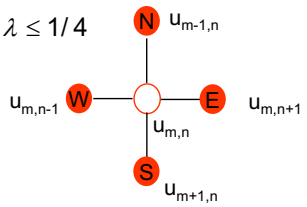
remark: $div\ v = \frac{\partial A}{\partial x} + \frac{\partial B}{\partial y}$ $\quad v = (A,B)$

discretization

Perona & Malik algorithm is based on a discretization of the diffusion equation in which the partial derivatives are replaced by first order differences.

where

$$g_{m,n}^{t+1} = g_{m,n}^{t} + \lambda[c_N \nabla_N g + c_S \nabla_S g + c_E \nabla_E g + c_W \nabla_W g]_{m,n}^{t} \qquad 0 \le \lambda \le 1/4$$



$$\nabla_N u_{m,n} = u_{m-1,n} - u_{m,n} \qquad \nabla_S u_{m,n} = u_{m+1,n} - u_{m,n}$$

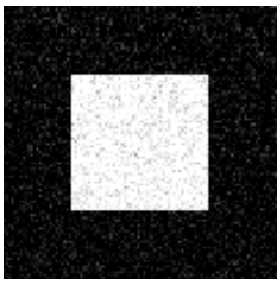
$$\nabla_E g_{m,n} = g_{m,n+1} - g_{m,n} \qquad \nabla_S g_{m,n} = g_{m,n-1} - g_{m,n}$$

$$c_{N\,m,n} = g(\|(\nabla g)_{m+1/2,n}\|) f_{m,n} \qquad c_{S\,m,n} = g(\|(\nabla g)_{m-1/2,n}\|)$$

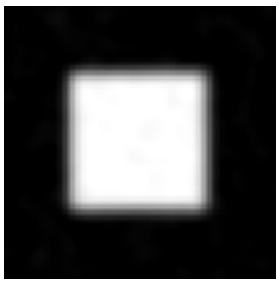
$$c_{E\,m,n} = g(\|(\nabla g)_{m,n-1/2}\|) f_{m,n} \qquad c_{W\,m,n} = g(\|(\nabla g)_{m,n+1/2}\|)$$

example

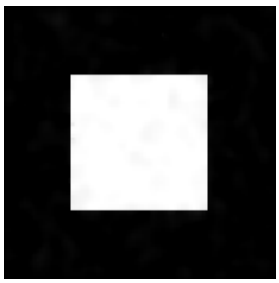
original

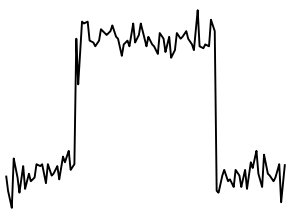


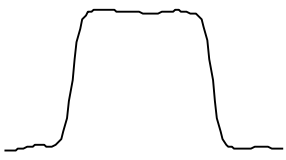
Gaussian filter

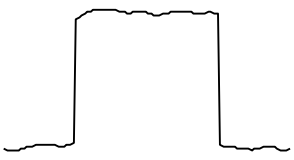


Perona-Malik



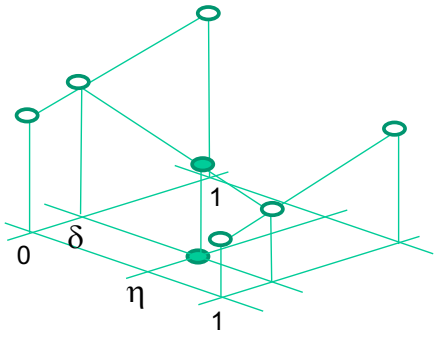






interpolation and pyramids

bilinear interpolation

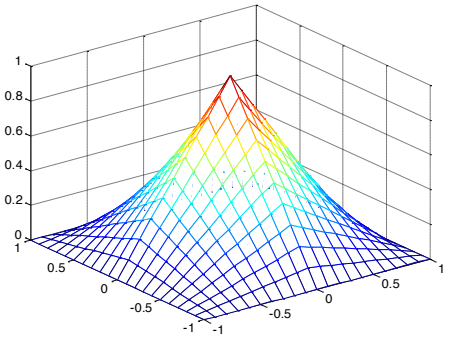


$$Data:\ f(x,y) \quad x,y \in \{0,1\}$$

$$f(0,\delta) = (1-\delta)f(0,0) + \delta f(0,1)$$

$$f(1,\delta) = (1-\delta)f(1,0) + \delta f(1,1)$$

$$f(\eta,\delta) = (1-\eta)f(0,\delta) + \eta f(1,\delta)$$



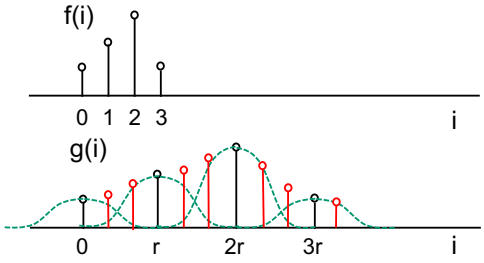
Replacing

$$f(\eta,\delta) = (1-\eta)(1-\delta)f(0,0) + (1-\eta)\delta f(0,1) + \eta(1-\delta)f(1,0) + \eta\delta f(1,1)$$

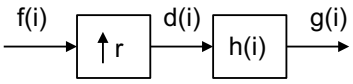
interpolation at equally spaced points

Sometimes we wish to increase the resolution of images. This can be done by interpolation.

1D



$$g(i) = \sum_k f(k)h(i - kr)$$



$$d(i) = \begin{cases} f\left(\frac{i}{r}\right) & i = kr \\ 0 & otherwise \end{cases}$$

$$g(i) = d(i) * h(i) = \sum d(p)h(i - p)$$

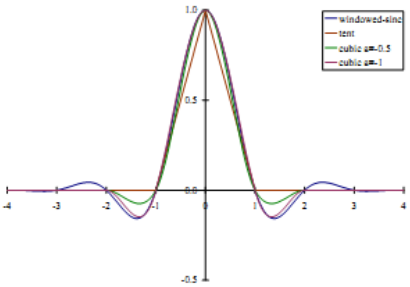
$$= \sum_k f(k)h(i - kr) \qquad p = kr$$

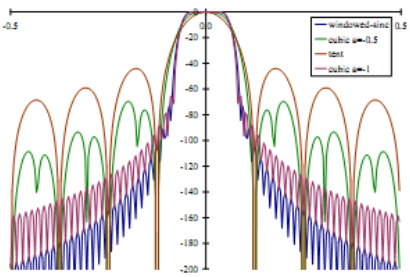
2D

$$g(i,j) = \sum_{k,l} f(k,l)h(i - kr, j - lr)$$

Interpolation results

filters





bilinear

bicubic (a=-1)

bicubic (a=-0.5)

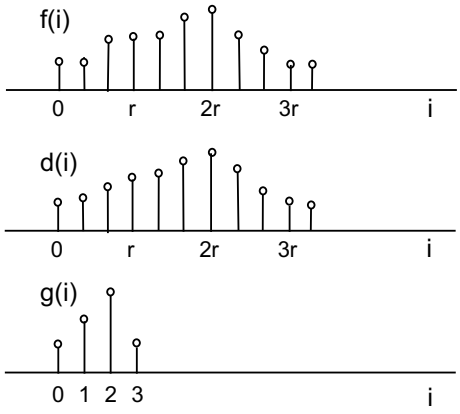
windowed sinc

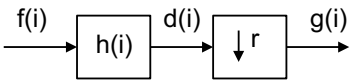
extracted from Szeliski book

decimation

Decimation reduces the image resolution by a factor k. It involves two steps: 1) lowpass filtering (anti-aliasing) and 2) subsampling.

1D





$$d(i) = \sum_k f(k)h(i-k)$$

$$g(i) = d(ri) = \sum_k f(k)h(ri-k)$$

2D

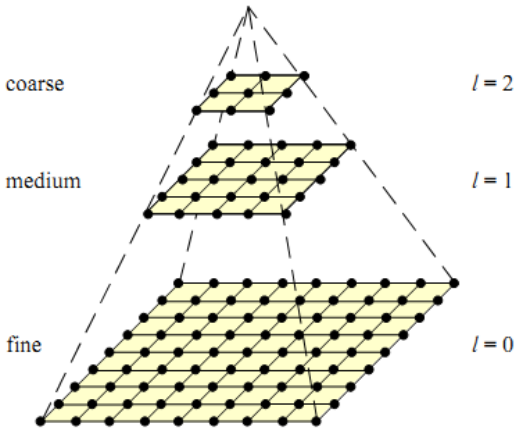
$$g(i,j) = \sum_{k,l} f(k,l)h(ri-k, rj-l)$$

h(k,l) is the decimation kernel

$$g(i,j) = \sum_{k,l} f(k,l)h(i-k/r, j-l/r)$$

if *h(k,l)* is the interpolation kernel

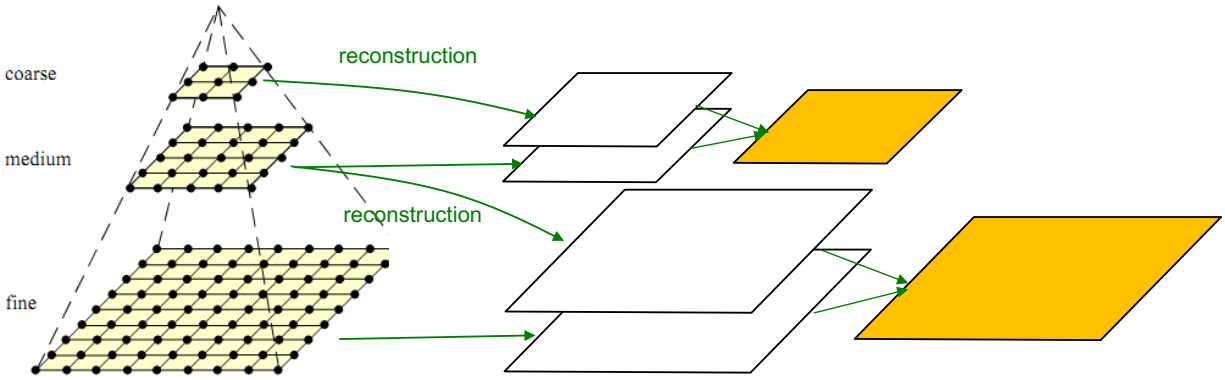
Gaussian pyramid (Burt & Adelson)



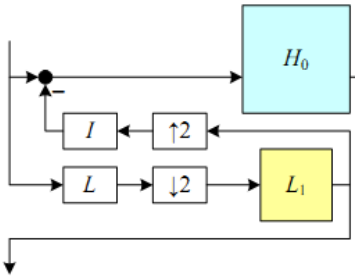
lowpass filter

$$\frac{1}{16}[1\ 4\ 6\ 4\ 1]$$

Laplacian pyramid (Burt & Adelson)



Block diagram



- perfect reconstruction
- ineficient representation

Szeliski book

Credits

Some slides are based on:

Richard Szeliski, Computer Vision: Algorithms and Application, Springer, 2011