TADI: Scale space theory
Master IMA/DIGIT
Sorbonne University

Dominique.Bereziat@lip6.fr

`http://lip6.fr/Dominique.Bereziat/cours/master/tadi`

Year 2024-2025 – version 1

Part 1: linear scale space

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Scale and image primitive (1)

- In most of image processing problems:

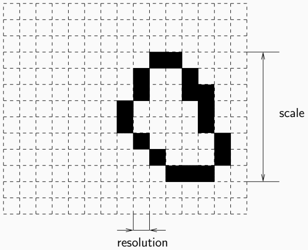
Axiom
Detection of an image primitive is possible only at an optimal scale

Axiom ([Marr and Hildreth, 1980])
An image is characterized by its local intensity variation at each scale

- An image primitive: edges, contours, regions, pixels structures,
- Scale: an abstract measurement of image structures size, in practice, we measure in "pixels"

Scale and image primitive (2)

- Image resolution: number of pixels in an image
- and then: $\boxed{\text{scale} \neq \text{resolution}}$



- Range of scales are bounded:
  - minimal scale: 1 pixel, the real size may be known if we knows the pixel "size"
  - maximal scale: given by the image resolution (i.e. the number of pixels)
- Primitive ⇒ a local brightness variation ⇒ a differential operator to approximate ⇒ linear filter size suited to the image primitive size. Ex: $f'(x) \simeq f(x+1) - f(x)$

Importance in Image processing

- Optimization $\Rightarrow$ calculus of variation $\Rightarrow$ partial differential equations $\Rightarrow$ approximation of differential operators
- Continuous and discrete solutions belong to a scale space (admitted)
- A dedicated conference: *Conference on Scale Space and Variational Methods in Computer Vision*

  Scope: **Image analysis Scale-space methods Level sets methods PDEs in image processing, Inverse problems in imaging Compressed sensing, Optimization methods in imaging Convex and non convex modeling and optimization in imaging, Restoration and reconstruction, Multi-scale shape analysis, 3D imaging modalities, 3D vision, Wavelets and image decomposition, Segmentation, Stereo reconstruction, Optical flow, Motion estimation Registration, Surface modeling, Implicit surfaces, Shape from X, Inpainting, Color enhancement, Perceptual grouping, Selection of salient scales, Feature analysis, Cross-scale structure, Multi-Orientation Analysis, Differential geometry and Invariants, Mathematics of novel imaging methods, Sub-Riemannian geometry, Medical imaging.**

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

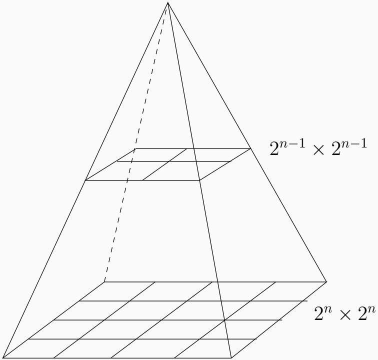
APPENDIX

A first description of scale spaces

- The idea is not novel:
    - [Rosenfeld and Thurston, 1971]: bank of various operators with different sizes for edges detection:
    - [Klinger, 1971, Uhr, 1972]: representation at various resolution (obtained by under-sampling) $\Rightarrow$ pyramid of resolutions
        - consequence: primitive scales decreases with the image resolution
        - advantage: any primitive can be detected with the same operator: it exists an optimal representation (a level in the pyramid) to detect a given primitive

Pyramid of resolutions (1)

- Let's $K > 0$ an natural number, and $0 < k \leq K$
- $I^K \equiv I$ image with finest resolution
- I^k image at resolution (level) k is obtained from superior resolution $k+1$:
    - applying an anti-aliasing filter (a Gaussian smoother)
    - and downsampling of factor 2 in the two image direction (image size divided by 4)



Pyramid of resolutions (2)

- Reduction operator R, in 1-D:

$$f^{(k-1)}(x) = \mathcal{R}(f^{(k)}) = \sum_{n \in \mathbb{N}} c(n) f^{(k)}(2x - n)$$

  with c a low-pass filter

- In 2-D: we get a separable filter, $c(n, p) = c(n)c(p)$

$$I^{(k-1)}(x, y) = \mathcal{R}(f^{(k)}) = \sum_{n \in \mathbb{N}} \sum_{p \in \mathbb{N}} c(n, p) I^{(k)}(2x - n, 2y - p)$$

Characterization of c (1-D)

- Spatial properties:
  - finite support in $\mathbb{Z}$: $\exists N$ such as $\forall |n| > N, \quad c(n) = 0$
  - positivity: $c(n) \geq 0$
  - unimodality: $c(|n|) \geq c(|n+1|)$
  - symmetry: $c(n) = c(-n)$
  - normalization: $\sum c(n) = 1$
  - identical contribution at each level: $\sum c(2n) = \sum c(2n+1)$
- Frequency properties:
  - low-pass filter: high frequencies components are set to zero to avoid aliasing
  - should approximate an ideal low-pass filter (ideal filter have an infinite support in frequency domain)
- Examples:
  - $N = 3$: unique filter: $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$
  - $N = 5$: $(\frac{1}{4} - \frac{a}{2}, \frac{1}{4}, a, \frac{1}{4}, \frac{1}{4} - \frac{a}{2})$, with $a = 0.4$ we are closest to a Gaussian

Band-pass pyramid

- between each level, use a pass-band instead of low-pass
- construction: subtract adjacent levels of a low-pass pyramid,

$$
\begin{aligned}
L^{(k)} &= f^{(k)} - \mathcal{E}(f^{(k-1)}) && (1)\\
L^{(0)} &= f^{(0)} && (2)
\end{aligned}
$$

- $\mathcal{E}$ upsample operator to retrieve the previous resolution
- but: loss of information

$$\mathcal{E}(f^{(k-1)}) = 2\sum c(n) f^{(k-1)}\left(\frac{x-n}{2}\right)$$

Band-pass pyramid: applications

- Structure detection: image structures are localized in the pyramid levels compatible with their sizes
- Information is not redundant between the pyramid levels: low correlation between levels, each level may be efficiently compressed
- Easy to select some details levels
- Imperfect reconstruction is possible by inverting Eqs.(1, 2):

$$\begin{aligned}\tilde{f}^{(0)} &= L^{(0)}\\ \tilde{f}^{(k)} &= L^{(k)} + \mathcal{E}(\tilde{f}^{(k-1)})\end{aligned}$$

Pyramid of resolutions: pros and cons

- Pro:
    - Easy to implement, low complexity
    - Allows to implement efficient multiresolution algorithms
- Cons:
    - non invariant by translation
    - non invariant by rotation
- To go further: [Burt, 1981], [Crowley, 1981]

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

First definition of scale scale (in $\mathbb{R}$)

- Continuous case: scale parameter and spatial domain are continuous
- First definition appears in founding paper [Witkin, 1983] and addresses 1-D signals
- Given a signal f, we derive a family of signals L indexed by a scale parameter $t \in \mathbb{R}^+$:

$$\begin{array}{rc} f : \mathbb{R} \to \mathbb{R} & \text{original signal} \\ L : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R} & \text{scale representation} \end{array}$$

- Definition of L:

$$\begin{aligned} L(x, t = 0) &= f(x) \\ L(x, t) &= g_{\sigma(t)} \star f(x), \quad t > 0 \end{aligned}$$

Choice for g

- g is a convolution kernel
- a Gaussian of standard deviation σ:

$$g_\sigma = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

- Rem: $\sigma(t) = \sqrt{t}$
- Action on a signal:

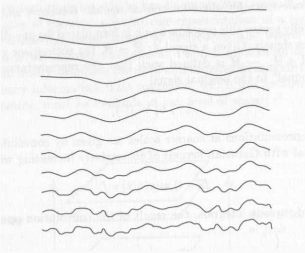


Figure 1: From [Witkin, 1983]

One more dimension

- L belongs to a space having one more dimension than those of f
- Supplementary dimension: a space of scale factor t:

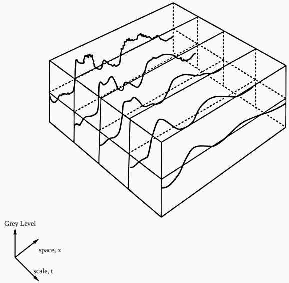


Figure 2: From [Witkin, 1983]

- A first application: look for a local extrema at a given scale and downscale

Link with the heat equation

- The diffusion equation (in 1-D):

$$\begin{aligned} \frac{\partial}{\partial t} L(x,t) &= \frac{1}{2}\frac{\partial}{\partial x^2}{}^{2} L(x,t) & (3) \\ L(x,0) &= f(x) \quad \text{(initial condition)} & (4) \end{aligned}$$

- This equation rules the heat diffusion (L is a temperature) in an homogeneous 1-D medium

Theorem
A solution of system (3,4) is given by:

$$L(x,t) = g_{\sqrt{t}} \star f(x)$$

where g is the Gaussian function with standard deviation of $\sqrt{t}$

Proof

- Easy to verify that $g_{\sqrt{t}}$ is solution of (3):

$$\begin{aligned}
g(x,t) &= \frac{1}{t}e^{-\frac{x^2}{2t}} \\
\frac{\partial}{\partial t}g(x,t) &= -\frac{1}{2t^2}e^{-\frac{x^2}{2t}} + \frac{1}{t}\left(\frac{x^2}{2t^2}\right)e^{-\frac{x^2}{2t}} \\
&= \frac{1}{2t^2}e^{-\frac{x^2}{2t}}\left[\frac{x^2}{t}-1\right] \\
\frac{\partial}{\partial x}g(x,t) &= -\frac{x}{t^2}e^{-\frac{x^2}{2t}} \\
\frac{\partial^2}{\partial x^2}g(x,t) &= -\frac{1}{t^2}e^{-\frac{x^2}{2t}} - \frac{x}{t^2}\left(-\frac{x}{t}\right)e^{-\frac{x^2}{2t}} \\
&= \frac{1}{t^2}e^{-\frac{x^2}{2t}}\left[\frac{x^2}{t}-1\right] \\
\Rightarrow \quad \frac{\partial}{\partial t}g(x,t) &= \frac{1}{2}\frac{\partial^2}{\partial x^2}g(x,t)
\end{aligned}$$

Proof (next)

- $g_{\sqrt{t}}(x)$ is not solution of eq. (4) (indetermination for $t = 0$)

Lemma

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

- replace L by $f \star g_{\sqrt{t}}$ in equation (3), apply the lemma, and remark that $f \star g$ is solution of eq. (3):

$$\begin{aligned}
\frac{\partial}{\partial t}(f \star g_{\sqrt{t}}) - \frac{1}{2}\frac{\partial^2}{\partial x^2}(f \star g_{\sqrt{t}}) &= f \star \frac{\partial}{\partial t} g_{\sqrt{t}} - \frac{1}{2} f \star \frac{\partial^2}{\partial x^2} g_{\sqrt{t}} \\
&= f \star (\frac{\partial}{\partial t} g_{\sqrt{t}} - \frac{1}{2}\frac{\partial^2}{\partial x^2} g_{\sqrt{t}}) \\
&= 0
\end{aligned}$$

Proof (next)

- Is eq. (4) verified?
- $g_{\sqrt{t}} \star f(x) = \int_{\mathbb{R}} f(x-u) g_{\sqrt{t}}(u) du$
- Admitted:

$$\lim_{t \to 0} \int_{\mathbb{R}} f(x) \frac{1}{t\sqrt{2\pi}} e^{-\frac{x^2}{2t}} dx = \int_{\mathbb{R}} f(x) \delta(x) dx$$

  for all function f integrable in $\mathbb{R}$

- Dirac delta function (δ) is a generalized function (distribution) implicitly defined by:

$$\int_{\mathbb{R}} f(x) \delta(x) dx = f(0)$$

- by definition of δ, it comes:

$$\lim_{t \to 0} \int_{\mathbb{R}} f(x-u) g_{\sqrt{t}}(u) du = \int_{\mathbb{R}} f(x-u) \delta(u) du = f(x)$$

Proof (lemma)

- Distributivity:

$$\begin{aligned}
\frac{\partial}{\partial x}(f \star g)(x) &= \frac{\partial}{\partial x}\int_{\mathbb{R}} f(y)g(x-y)dy \\
&= \int_{\mathbb{R}} \frac{\partial}{\partial x}(f(y)g(x-y))dy \\
&= \int_{\mathbb{R}} f(y)\frac{\partial}{\partial x}g(x-y)dy
\end{aligned}$$

- Linearity:

$$\begin{aligned}
f \star (g+h) &= \int_{\mathbb{R}} f(x-y)(g(y)+h(y))dy \\
&= \int_{\mathbb{R}} f(x-y)g(y)dy + \int_{\mathbb{R}} f(x-y)h(y)dy
\end{aligned}$$

- Recall that convolution is commutative:

$$f \star g(x) = \int_{\mathbb{R}} f(x-y)g(y)dy = \int_{\mathbb{R}} f(y)g(x-y)dy$$

  by change of variable (domain reversal)

Causality principle

- A fundamental principle is verified: non creation of "structures" when the scale increases
- Hypothesis: a structure in an image is characterized by its edges, and then, by its image value extrema
  - at scale t_0, if x_0 is a local maxima, then $\frac{\partial^2}{\partial x^2} L(x_0, t_0) < 0 \Rightarrow \frac{\partial}{\partial t} L(x_0, t_0) < 0$
  - Taylor expansion at t: $L(x_0, t) \sim L(x_0, t_0) + \frac{\partial}{\partial t} L(x_0, t_0)(t - t_0)$
  - then: $\forall t > t_0, \quad L(x_0, t) < L(x_0, t_0)$
  - at scale t_0, if x_0 local minimum, then $\nabla^2 L(x_0, t_0) > 0 \Rightarrow \frac{\partial}{\partial t} L(x_0, t_0) > 0$
  - then $\forall t > t_0, \quad L(x_0, t) > L(x_0, t_0)$
- Local maxima decrease, local minima increase: the image tends to be more and more homogeneous

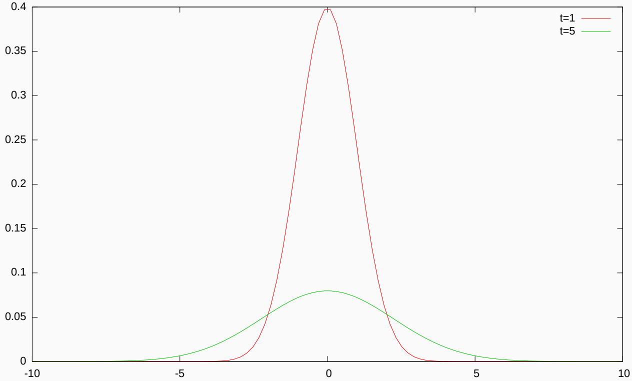
Properties of the Gaussian kernel

- It is a smoothing filter: it does not create new image structures
- It is a weighted mean filter:

$$\int_{\mathbb{R}} g_\sigma(x) dx = 1 \tag{5}$$

- It has fast decreasing toward zero: beyond $|x| > \sqrt{t}, g(x) \approx 0$
  $\Rightarrow$ image structures smaller than $\sqrt{t}$ are suppressed

Gaussian kernel graph



Semi-group property

- The family of Gaussian functions $(F, g\star)$ has a structure of semi-group, i.e.:
  - $g(.,t) \star g(.,s) = g(.,t+s)$
- A representation at scale t_2 can be deduced from any inferior scale t_1:

$$t_2 > t_1, L(t_2, x) = g_{\mathbf{t_2 - t_1}} \star L(t_1, x)$$

- This property is used in the pyramid of resolutions
- In frequency domain, we have:

$$\widehat{L}_{\mathbf{t_2}}(w) = \widehat{g}_{\mathbf{t_2 - t_1}}(w) \times \widehat{L}_{\mathbf{t_1}}(w)$$

Concluding remarks

- Gaussian filter: details smaller than a scale disappear
- Question: does it exist another filters than Gaussian?
- A desirable property: the causality property (no new image structures at higher representation scale)
- In particular: noise should decrease with the scale, and should never be amplified
- Causality is a central property of the scale space theory, see [Koenderink, 1984]
- How to generalize to $\mathbb{R}^2$?

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Continuous linear scale space in $\mathbb{R}^2$

Definition (Continuous scale space for an image)
Let $I : \mathbb{R}^2 \to \mathbb{R}$ be an image. Let L be a familly of images, derived from I, such as:

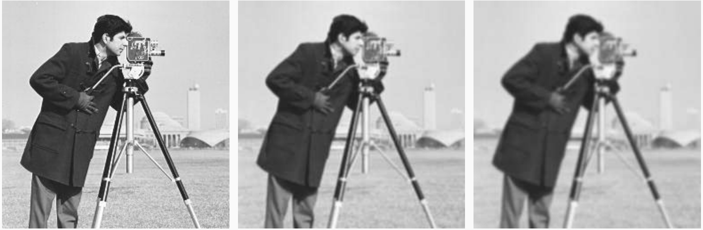
$$\begin{aligned} L : &\quad \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R} \\ L(\mathrm{x}, 0) &= I(\mathrm{x}), \forall \mathrm{x} \in \mathbb{R}^2 \\ L(\mathrm{x}, t) &= g_{\sqrt{t}} \star I(\mathrm{x}), \forall \mathrm{x} \in \mathbb{R}^2, t \in \mathbb{R}^+ \end{aligned} \tag{6}$$

with:

$$\begin{aligned} g_{\sqrt{t}}(\mathrm{x}) &\equiv \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}} \\ f \star g(\mathrm{x}) &\equiv \int_{\mathbb{R}^2} g(x, y) f(x' - x, y' - y) dx' dy' \end{aligned}$$

L is called *scale representation* of the image I, parameter t is the scale

Example



(a) Original Image (b) $t = 1.0$ (c) $t = 1.5$

Figure 3: Example of scale decomposition

- Larger structures are suppressed with higher scales. . .

Example (animation)

Loading data ...

Figure 4: Animating scale space

The 2-D Gaussian function

- The 2-D Gaussian function is separable:

$$\begin{aligned} g_\sigma(\mathrm{x}) &= g_\sigma(x) \times g_\sigma(y) \\ &= \frac{1}{\sqrt{2\pi t}} e^{-x^2/2t} \frac{1}{\sqrt{2\pi t}} e^{-y^2/2t} = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \end{aligned}$$

- Implementation of 2-D Gaussian smoothing is efficient:

$$I \star g_\sigma(\mathrm{x}) = \int_{\mathbb{R}} \left(\int_{\mathbb{R}} I(x - x', y - y') g_\sigma(x') dx' \right) g(y') dy'$$

- This property holds for superior dimensions

The 2-D Heat equation

Theorem
The scale space representation L of a function $f : \mathbb{R}^2 \to \mathbb{R}$ verifies the following system of equations:

$$\begin{aligned} \frac{\partial}{\partial t} L(\mathrm{x}, t) &= \frac{1}{2}\nabla^2 L(\mathrm{x}, t) && (7) \\ L(\mathrm{x}, 0) &= I(\mathrm{x}) && (8) \end{aligned}$$

- $\nabla^2 L = \nabla^T \nabla = \sum_{i=1}^{2} \frac{\partial^2}{\partial x_i^2} L$ is the Laplacian operator
- System (7 ,8) is the Heat equation in a 2-D homogeneous medium

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Differentiation in scale space

- Let's consider an image at scale t, and its partial derivative respect to spatial coordinate:

$$\begin{aligned} \frac{\partial}{\partial x} L(x,t) &= \frac{\partial}{\partial x}\left(I \star g_{\sqrt{t}}(x)\right) \\ &= I \star \frac{\partial}{\partial x} g_{\sqrt{t}}(x) \end{aligned}$$

- Function $g_{\sqrt{t}}$ is C^∞ and $\frac{\partial^n}{\partial x^n} g_{\sqrt{t}}$ is integrable $\forall n$
- If I is integrable, and not necessarily differentiable, then $\frac{\partial^n}{\partial x^n} L$ is defined
- Allows a *weak* definition of derivative, that can be applied to any functions even not differentiable:

$$\begin{aligned} I \star g_\sigma &\underset{\sigma \to 0}{\longrightarrow} I \\ I \star \frac{\partial^n}{\partial x^n} g_\sigma &\underset{\sigma \to 0}{\longrightarrow} \frac{\partial^n}{\partial x^n} I \end{aligned}$$

Properties of derivatives in scale space

- Space scale properties apply to image derivatives
- Indeed: if $\mathcal{L}$ solution of $\frac{\partial}{\partial t}L = \frac{1}{2}\nabla^2 L$ then $\partial\mathcal{L}$ also a solution
- Moreover:

$$
\begin{aligned}
g_{\sqrt{t_1}} \star \frac{\partial^n}{\partial x^n} g_{\sqrt{t_2}} &= \frac{\partial^n}{\partial x^n} g_{\sqrt{t_1+t_2}} && (9) \\
\frac{\partial^m}{\partial x^m} g_{\sqrt{t_1}} \star \frac{\partial^n}{\partial x^n} g_{\sqrt{t_2}} &= \frac{\partial^{n+m}}{\partial x^{n+m}} g_{\sqrt{t_1+t_2}} && (10)
\end{aligned}
$$

- If a computer vision problem is formulated as solution of linear PDEs, solutions live in a scale space

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Uniqueness of continuous scale space representation (1)

- Let us consider an *axiomatic* definition of scale space: the set of parametric functions derived from an image I and an operator H_t such as
    1. linearity: $I \mapsto L_t = H_t(I)$ is linear
    2. translation invariant system: $H(I \circ \mathcal{T}) = H(I) \circ \mathcal{T}$, $\mathcal{T}$ translation, $\circ$ composition operator
    3. semi-group structure: $H_{t_1+t_2} = H_{t_1} \circ H_{t_2}$
- What operators H may be available?

Theorem (Uniqueness of Gaussian kernel)
All continuous space scale representations write:

$$L(\mathrm{x}, t) = h_t \star I(\mathrm{x}) = \int_\Omega h_t(\mathrm{x}')f(\mathrm{x} - \mathrm{x}')d\mathrm{x}' \qquad (11)$$

with h_t the Gaussian function of variance t

Uniqueness of continuous scale space representation (2)

- Axioms 1. (linearity) and 2. (translation invariant system) lead to a convolution with a kernel to be determined
- Translation invariant: a change of variable $\mathsf{x} \to \mathsf{x} - a$ under the integral (convolution) leaves unchanged the spatial domain ($\mathbb{R}^2$)
- Uniqueness relies on the Pi-theorem (admitted)

Uniqueness of continuous scale space representation (3)

Lemma (Pi-theorem (Vaschey-Buckingham))
Consider any physical system f relying n state variables (with dimension) such as $f(q_1, \cdots, q_n) = 0$.
Then it exists a family of variables π_i without dimension and a function F such as:

$$\begin{cases} F(\pi_1, \cdots, \pi_p) = 0 \\ \pi_i = \prod_{k=1}^{n} q_k^{l_k} \end{cases} \tag{12}$$

Variables π_i are without dimension as explained as a product (Pi) of variables q_j.

- Pi-theorem means that a physical system never depend on the choice of units (and the scale)
- Main lines of the proof are given in appendix

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Implementation (Matlab)

```matlab
function g = gD( f, scale, ox, oy)
%%% Perform a gaussian derivative convolution
%%%  f: input image
%%%  scale: smoothing parameter
%%%  ox,oy: x and y derivate order (0,1,2,3)

% Build the gaussian kernel
K = ceil(3*scale);
x = -K:K;
Gs = exp(-x.^2/(2*scale^2));
Gs = Gs/sum(Gs);

% Calculate the derivatives en x and y-direction
Gsx = gDerivative(ox,x,Gs,scale);
Gsy = gDerivative(oy,x,Gs,scale);

% Do the convolution
g = convSepBrd( f, Gsx, Gsy);
```

Implementation (Matlab)

```
function r = gDerivative( order, x, Gs, scale)
%%% Compute a derivative of a 1D Gaussian kernel
%%% order: order derivation 0,1,2,3
%%% Gs: discretized, centered gaussian kernel
%%% scale: variance of Gs
switch order
 case 0
  r = Gs;
 case 1
  r = -x/(scale^2) .* Gs;
 case 2
  r = (x.^2-scale^2)/(scale^4) .* Gs;
 case 3
  r = x.*(3*scale^2-x.^2)/(scale^6) .* Gs;
 otherwise
  error('only␣derivatives␣up␣to␣third␣order␣are␣supported');
end
```

Implementation (Matlab)

```
function g = convSepBrd( f, w1, w2)
%%% convolve along colums and rows with repetition
%%% of the border

  N = size(f,1);
  M = size(f,2);
  K = (size(w1(:),1)-1)/2;
  L = (size(w2(:),1)-1)/2;
  %% 1 1 1 1 1 | 1 2 ... N | N N N N N
  %%  K times                K times
  iind = min(max((1:(N+2*K))-K,1),N);
  jind = min(max((1:(M+2*L))-L,1),M);
  %% f(1,.) K times then f(1,.) ... f(N,.)
  %%                then f(N,.)  K times
  %% then border repetition...
  fwb = f(iind,jind);
  g = conv2(w1,w2,fwb,'valid');
```

Examples

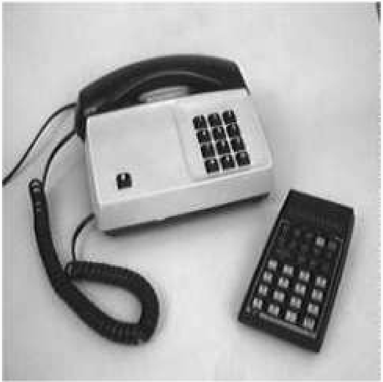


Figure 5: `phone=imread("phone.pgm");imshow(phone)`

Examples

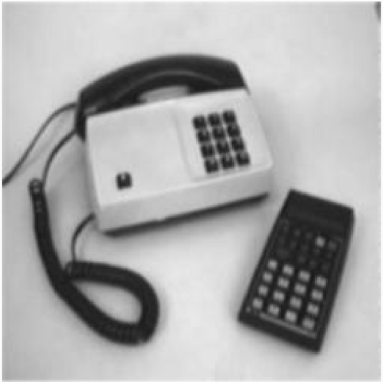


Figure 6: `imshow(gD(phone,1,0,0)`

Examples

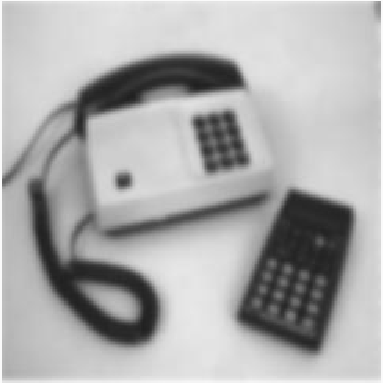


Figure 7: `imshow(gD(phone,2,0,0)`

Examples

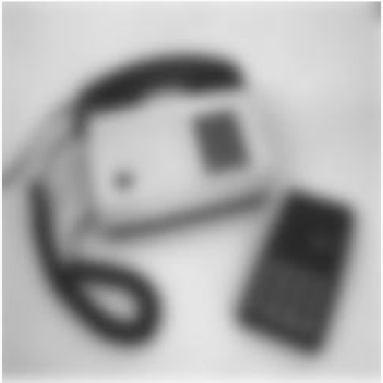


Figure 8: `imshow(gD(phone,5,0,0)`

Examples

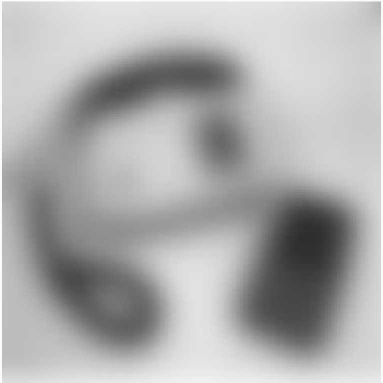


Figure 9: `imshow(gD(phone,10,0,0)`

Examples

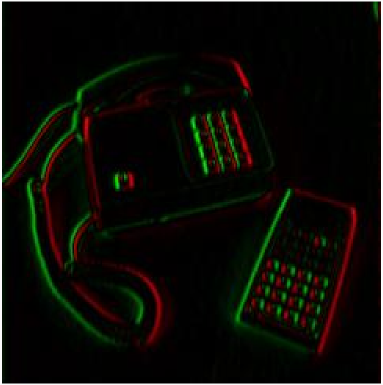


Figure 10: `imshow(gD(phone,1,1,0)`

Examples

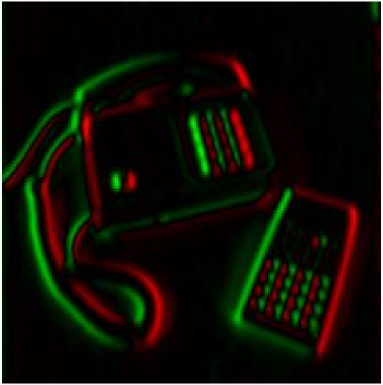


Figure 11: `imshow(gD(phone,2,1,0)`

Examples

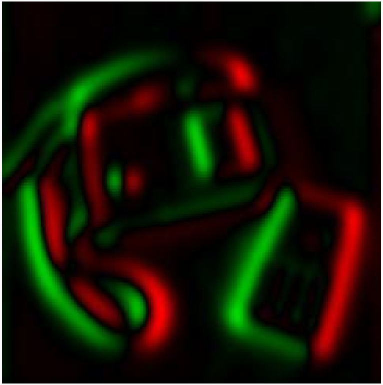


Figure 12: `imshow(gD(phone,5,1,0)`

Examples

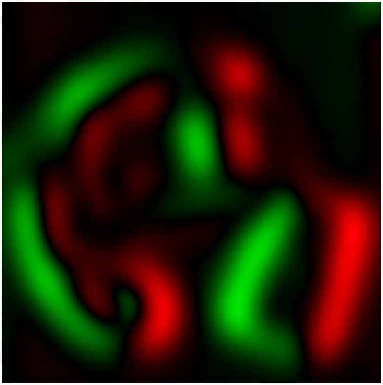


Figure 13: `imshow(gD(phone,10,1,0)`

Examples

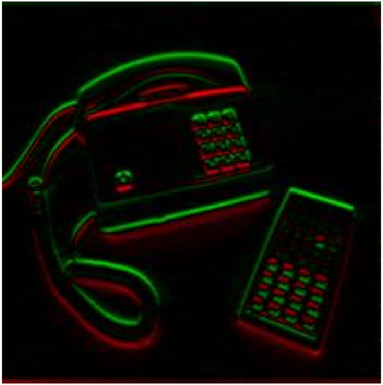


Figure 14: `imshow(gD(phone,1,0,1)`

Examples

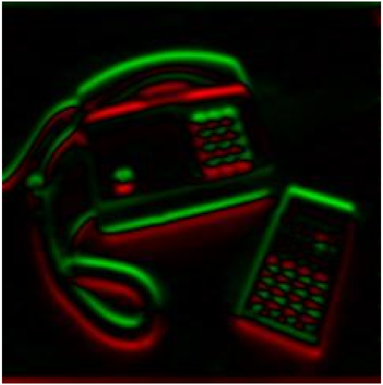


Figure 15: `imshow(gD(phone,2,0,1)`

Examples

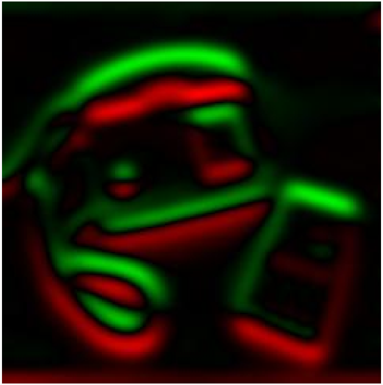


Figure 16: `imshow(gD(phone,5,0,1)`

Examples

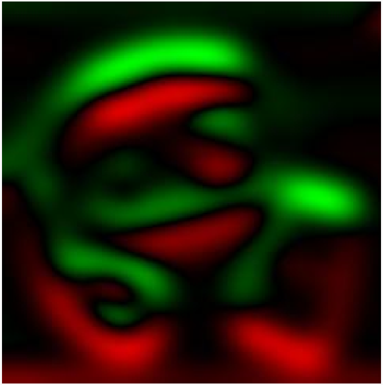


Figure 17: `imshow(gD(phone,10,0,1)`

Examples

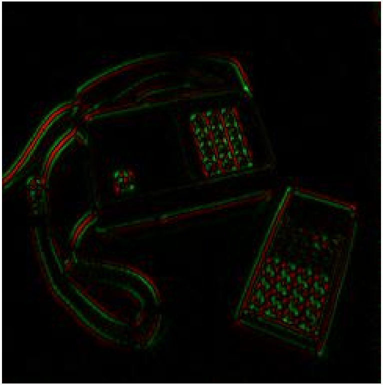


Figure 18: `imshow(gD(phone,1,1,1)`

Examples

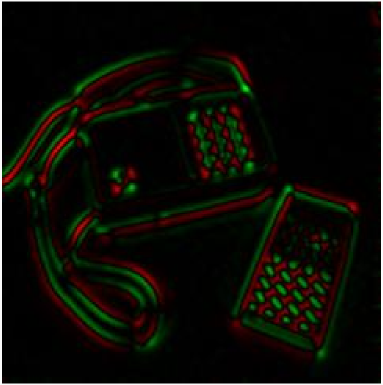


Figure 19: `imshow(gD(phone,2,1,1)`

Examples

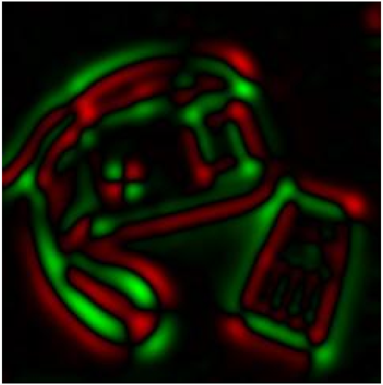


Figure 20: `imshow(gD(phone,5,1,1)`

Examples

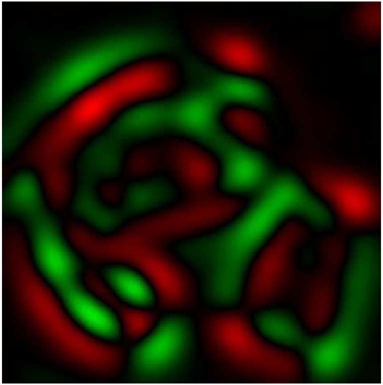


Figure 21: `imshow(gD(phone,10,1,1)`

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Automatic scale selection (1)

- scale space techniques can detect objects whose the size is known. Practically, this size is not known.
- In the following, we model a 1-D image structure by the following signal:

$$f(x) = \sin \omega_0 x$$

  ω_0, pulsation, represents a structure of period $\frac{1}{\omega_0}$
- Scale space representation of this signal:

$$\begin{aligned} L(x;t) &= f \star g_{\sqrt{t}}(x) \\ &= e^{-\omega_0^2 t/2} \sin \omega_0 x \end{aligned}$$

- We remark that $t \mapsto |L(.,t)|_\infty = e^{-\omega_0^2 t/2}$ is monotone decreasing

Automatic scale selection (2)

- Derived signal:

$$\begin{aligned} L^{(m)}(x,t) &= \frac{\partial^m L(x,t)}{\partial x^m} \\ |L^{(m)}(.,t)|_\infty &= \omega_0^m e^{-\omega_0^2 t/2} \end{aligned}$$

- Again $t \mapsto |L^{(m)}(.,t)|_\infty$ is monotone decreasing
  → there is no optimal scale to detect a structure of size $\frac{1}{\omega_0}$
- [Lindeberg, 1998]: let us introduce the *normalized derivative*:

$$\partial_x^\gamma = t^{\gamma/2} \frac{\partial}{\partial x}, \qquad \gamma > 0$$

Automatic scale selection (3)

- Normalized derivative of L at order m:

$$L_\gamma^{(m)}(x,t) = \left(t^{\gamma/2}\right)^m \frac{\partial^m L}{\partial x^m}$$

- The maximum of this signal:

$$|L_\gamma^{(m)}(.,t)|_\infty = t^{m\gamma/2}\omega_0^m e^{-\omega_0^2 t/2}$$

- $t \mapsto |L_\gamma^{(m)}(.,t)|$ is no more monotone decreasing along scale parameter, and has a unique maximum at:

$$t_{\max} = \frac{\gamma m}{\omega_0^2}$$

  It exists an optimal scale to detect a structure of size $\frac{1}{\omega_0}$, the optimal scale value depends on ω_0

- The maximum depends on ω_0:

$$\max(|L_\gamma^{(m)}(.,t)|_\infty) = \frac{(\gamma m)^{\gamma m/2}}{e^{\gamma m/2}}\omega_0^{(1-\gamma)m} \tag{13}$$

Automatic scale selection (4)

- Practically, we get $\gamma = 1$: a special case for where the maximum of $L_\gamma^{(m)}$ is independent of ω_0

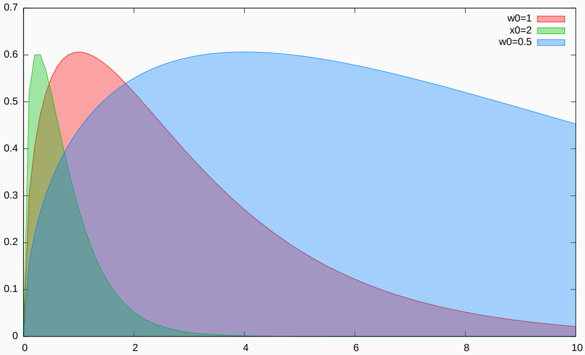


Figure 22: Maximal amplitude of $L_\gamma^{(m)}$

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Blob detection (1)

- a discrete image is a constant piecewise function and a set of *relatively homogeneous* regions
- Applying a Gaussian smoothing, regions get a Gaussian profile

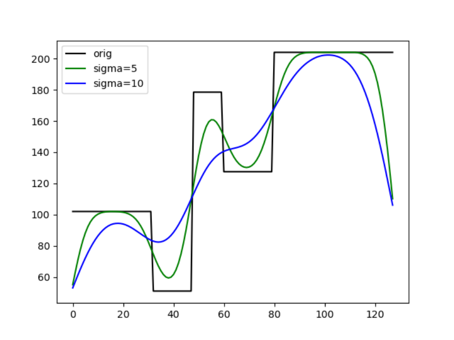


Figure 23: A (Gaussian) blob

Blob detection (2)

- Laplacian of Gaussian: Laplacian operator localizes the extrema of Gaussian blob both in space and in scale

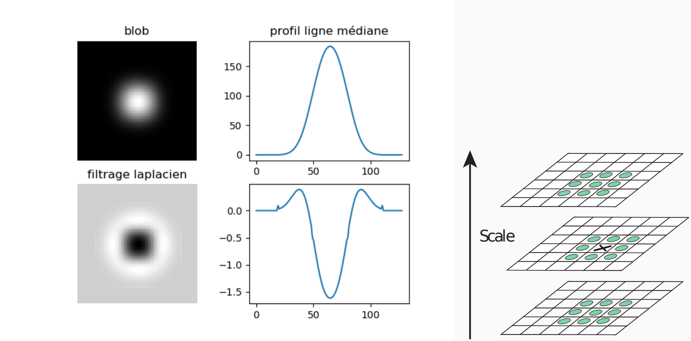


Figure 24: Action of Laplacian on a blob, local extrema in space and scale

Automatic scale selection for blob detection (1)

- Algorithm:
  - Compute $\partial^{\gamma}_{xx} L$, $\partial^{\gamma}_{yy} L$, $\partial^{\gamma}_{xy} L$ for various scales
    Practically, get $\gamma = 1$, then $\partial^{\gamma}_{xx} L = tL_{xx}$, $\partial^{\gamma}_{yy} L = tL_{yy}$, and $\partial^{\gamma}_{xy} L = tL_{xy}$
  - Blob detection: compute

    $$F(t) = \operatorname{trace} H(L) = t(L_{xx} + L_{yy})$$

    or

    $$F(t) = \det H(L) = t(L_{xx} L_{yy} - L_{xy}^2)$$

    for all scales
  - Localize local maxima of F **both in space and in scale**. The optimal scale informs on blob size

Automatic scale selection for blob detection (2)

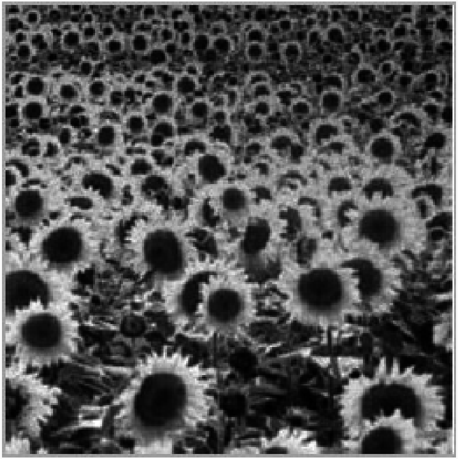
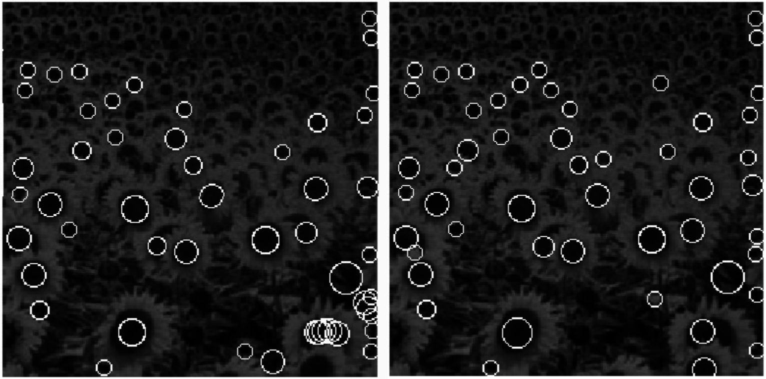


Figure 25: Sunflower field

Automatic scale selection for blob detection (3)



(a) Hessian trace

(b) Hessian determinant

Figure 26: Circle diameters are determined from the optimal scale (credit: Hailin Shi)

Edge detection

- First order detectors (approximation of first derivative)
    - Local maxima of gradient norm at various scales
    - Local maxima of normalized gradient norm both in space and scale
- Second order detectors:
    - Marr operator (Laplacian) at various scales and get zeros crossing
    - No automatic scale selection

Harris detector (recall)

- Consider the following tensor applied on an image I:

$$A(I)(x,y) = \nabla I^T(x,y)\nabla I(x,y) = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

- Practically, we consider $A(I) = G_{\sigma_I} \star \left(\nabla I^T(x,y)\nabla I(x,y)\right)$ with G_{σ_I} Gaussian function of variance ${\sigma_I}^2$ otherwise determinant of $\nabla I^T \nabla I$ is equal to zero
- $\sigma_I =$ *integration scale*: determines the "size" of corners to be detected
- Let's consider $\kappa \in [0.04, 0.15]$, Harris operator is defined by:

$$R(I) = \det(A(I)) - \kappa \operatorname{Tr}^2(A(I))$$

- (x,y) is a corner $\Leftrightarrow R(I)(x,y) > 0$ and (x,y) is a local maxima of R

Harris-Laplace detector

- Harris detector is rotation invariant as $R(I) = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$, λ_1 and λ_2 eigenvalues of $A(I)$
- σ_I is not a scale parameter: R is not scale invariant if a mono-scale approximation of image derivative (Sobel for instance) is used
- Harris-Laplace replaces $I(x, y)$ by $L(x, y, \sigma_D) = G_s \star I(x, y)$ in the definition of A
- Normalized derivatives may also be used ($L_x(x, y, \sigma_D)$, $L_x(x, y, \sigma_D)$)
- $\sigma_D =$ *differentiation scale*
- Harris-Laplace detector: (x, y, σ_D) is a corner $\Leftrightarrow R(I)(x, y, \sigma_I, \sigma_D) > 0$ and (x, y) is a local maximum of R both in space and differentiation scale for an given integration scale σ_I

Harris-Laplace detector and automatic selection of the integration scale

- How to choice σ_I?
- Algorithm [Lindeberg, 1998]:
  1. $k = 1$, s_1 set to an initial scale integration value
  2. (x_i, y_i) the set of detected corners by Harris-Laplace at $\sigma_I = s_1$
  3. $k = k + 1$
  4. Find the scale maximizing the normalized Laplacian of Gaussian of points (x_i, y_i)

     $$\hat{t} = \text{argmax}_{t \in [0.7, \cdots, 1.4]} \left| ts_k(L_{xx}(x_i, y_i, ts_k) + L_{yy}(x_i, y_i, , ts_k)) \right|$$

  5. $s_{k+1} = \hat{t} s_k$, determine (x_i, y_i) the set of corners detected by Harris-Laplace at scale integration s_{k+1}
  6. Iterate 3-4 up to convergence of s_k

Others multi-scale detectors

- Scale invariant features detectors : SIFT, SURF, . . . direct application of space-scale theory
- Not covered here but already studied in other M2 courses

Content

Part 1: linear scale space

- Introduction
- Earlier works: image decomposition
- Continuous linear scale space: 1-D case
- Continuous linear scale space in $\mathbb{R}^2$
- Differentiation in scale space
- Uniqueness of continuous scale space representation
- Implementation
- Automatic scale selection
- Applications
- Concluding remarks on the linear continuous case

Part 2: non linear scale spaces

APPENDIX

Linear continuous space scale: concluding remarks

- Solutions of the homogeneous heat equation can be expressed as a Gaussian smoothing
- Gaussian smoothing allows to select the primitives of a certain scale (i.e. size). This scale depends on the standard deviation of the Gaussian kernel
- Scale selection can be done optimally (using normalized derivatives)
- Scale space can be defined in an axiomatic way (linearity, shift invariant, semi-group) and is unique (Gaussian smoothing)

Perspectives: what's happen in the discrete case?

- Discrete in space, continuous in scale?
  - Important question: practically, the Gaussian kernel is sampled for a given scale.
  - A scale space representation is called *admissible* if:
    - is linear and shift invariant (i.e. discrete convolution)
    - **respects the causality principle** (i.e. low pass filter)
    - verify the semi-group property
  - The set of admissible kernels is a countable family and has been fully described by Schoenberg (1953)
- A direct consequence of Schoenberg: the Gaussian kernel sampling is not an admissible representation: the semi-group property is not respected excepted when t_1 divides t_2

Perspectives: what's happen in the discrete case?

- Another consequence: the closest infinite impulse response (IIR) admissible filter to sampled Gaussian filter is the *discrete analog of Gaussian kernel*

$$T(n,t) = e^{-\alpha t} I_n(\alpha t), \alpha > 0$$

  with $I_n(t) = (-i)^n J_n(it)$ and J_n Bessel function of first kind

- To go further: Lindeberg's book [Lindeberg, 1994] or `http://pequan.lip6.fr/~bereziat/cours/master/amo/amo-ss-cours2.pdf` (in French)

Part 2: non linear scale spaces

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Motivations (1)

- Linear scale space representation has desirable properties but also has some issues
    - loss of important details (contours) at higher scales
    - loss of localization and lower accuracy
- For some applications it's a concern!
    - denoising
    - segmentation
    - regularization
- Principle of non linear scale spaces: find representations that respect the causality principle AND that preserve image discontinuities (two antagonist constraints)

Motivations (2)

- Basic idea: consider a diffusion process **image driven**:
    - Over edges: no diffusion (contours are preserved)
    - Elsewhere: diffusion (smoothing)
- Equivalence between linear scale space and homogeneous heat equation (linear diffusion PDE)
- What's about non linear diffusion PDEs? Is it possible to derive alternative scale spaces?
- Roadmap: consider a non linear diffusion PDE and check if some scale space axioms are verified (causality property, edges preservation)

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Discretization of PDEs

- Numerous methods and depending on type of equation
- Here we present the finite difference method that applies on
    - elliptic equations (such as Poisson equation: $u_{xx} + u_{yy} = f$)
    - hyperbolic equations (such as waves equation: $u_{tt} = cu_{xx}$)
    - and finally parabolic equations (such as diffusion: $u_t = (D_x u)_x$)
- In the following we denote $u_x = \frac{\partial u}{\partial x}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$, ...
- Recall: Taylor expansion of a function f at point x_0:

$$f(x) \simeq \sum_{i=0}^{n} f^{(i)}(x_0)\frac{(x-x_0)^i}{i!} + \mathcal{O}((x-x_0)^n)$$

  with $f^{(i)}$ the i-order derivative, $\mathcal{O}(u) \underset{u\to 0}{\longrightarrow} 0$
- Approximation at order i:

$$f(x) \simeq \sum_{i=0}^{n} f^{(i)}(x_0)\frac{(x-x_0)^i}{i!}$$

  stands only in a neighborhood of x_0

Approximation of first derivative

- Example 1 (first order Taylor expansion): forward difference

$$\begin{aligned} f(x+\triangle) &\simeq f(x)+f'(x)\triangle \\ f'(x) &\simeq \frac{f(x+\triangle)-f(x)}{\triangle} \end{aligned} \tag{14}$$

- Example 2 (first order Taylor expansion): backward difference

$$\begin{aligned} f(x-\triangle) &\simeq f(x)-f'(x)\triangle \\ f'(x) &\simeq \frac{f(x)-f(x-\triangle)}{\triangle} \end{aligned} \tag{15}$$

- Example 3: centered difference (subtracting (15) to (14)

$$f(x+\triangle)-f(x-\triangle) \simeq 2f'(x)\triangle \tag{16}$$

$$f'(x) \simeq \frac{f(x+\triangle)-f(x-\triangle)}{2\triangle} \tag{17}$$

Approximation of second derivative

- Consider the two following Taylor expansions at second order:

$$\begin{aligned} f(x+\triangle) &\simeq f(x) + f'(x)\triangle + f''(x)\frac{\triangle^2}{2} \\ f(x-\triangle) &\simeq f(x) - f'(x)\triangle + f''(x)\frac{\triangle^2}{2} \end{aligned}$$

- Adding them leads to

$$f''(x) \simeq \frac{f(x+\triangle) - 2f(x) + f(x-\triangle)}{\triangle^2}$$

- Others schemes are possible: remark that $f''(x) = (f'(x))'$ and approximate in two steps using first order operators previously seen
- Important remark: loss of uniqueness in the discrete world

Application: approximation of the heat equation (1-D)

- The question: how to approximate the heat equation in a *correct* numerical scheme?

$$\frac{\partial u}{\partial t}(x,t) = c\frac{\partial^2 u}{\partial x^2}(x,t) \quad x \in [x_0, x_1], \quad t \in [t_0, t_1] \tag{18}$$

- Space discretization: $x \mapsto u(x,t)$ is sampled on an uniform grid
    - $x_j = x_0 + j \times \triangle_x$ with $j = 0 \cdots J$ and $J = \frac{x_1 - x_0}{\triangle_x}$, $\triangle_x$ is called space step
    - $\frac{\partial^2 u}{\partial x^2}(x_j, t) \simeq \frac{u(x_{j+1}, t) - 2u(x_j, t) + u(x_{j-1}, t)}{\triangle_x^2}$
- Time discretization: $t \mapsto u(x,t)$ is sampled on an uniform grid
    - $t_n = t_0 + n \times \triangle_t$, $\triangle_t$ is called time step
    - $\frac{\partial u}{\partial t}(x, t_n) \simeq \frac{u(x, t_{n+1}) - u(x, t_n)}{\triangle_t}$

Application: approximation of the heat equation (1-D)

- Replace left and right members in (18) by their approximation:

$$\frac{u_j^{n+1} - u_j^n}{\triangle_t} = c\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\triangle_x^2}$$

- Numerical scheme called FTCS (Forward Time, Centered Space):

$$u_j^{n+1} = u_j^n + c\frac{\triangle_t}{\triangle_x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{19}$$

- Given an initial condition (a vector $U^0 = \begin{pmatrix} u_0^0 & u_1^0 & \cdots & u_J^0 \end{pmatrix}^T$), we compute an approximate solution of $u(x^n, t_j)$
- The scheme is said **explicit** as we can compute in one pass U^n from U^{n-1}

Stability analysis of a numerical scheme

- Does a numerical scheme converge to a correct solution? A necessary condition is *stability*: the error

  $$\epsilon_j^n = u(x_j, t_n) - u_j^n$$

  remains small for all n, u_j^n is a numerical solution of the discrete equation computed with a finite precision, $u(x_j, t_n)$ is the truth solution of the continuous equation
- A tool: the spectral (Fourier) analysis
- Consider solutions in the form of $u(x_j, t_n) = \epsilon_j^n + u_j^n$, replace in (19):

  $$\epsilon_j^{n+1} = \epsilon_j^n - \alpha(\epsilon_{j+1}^n - 2\epsilon_j^n + \epsilon_{j-1}^n) \tag{20}$$

  with $\alpha = \frac{c\triangle_t}{\triangle_x^2}$
- Error verifies the discrete PDE

Spectral analysis

- Spectral analysis aims to study the behavior of solution writing:

$$\epsilon_j = \sum_{k=1}^{\frac{N}{2}} \xi_k e^{ikj\triangle_x}, \quad j = 0 \cdots J$$

- Error also depends on time but the Fourier basis don't, $\xi_k = \xi_k(n)$
- The numerical scheme, eq. (20), is similar to a geometric sequence, the error evolves exponentially in time:

$$\epsilon_j^n = \sum_{k=1}^{\frac{N}{2}} \xi_k^n e^{ikj\triangle_x}, \quad \text{with } \xi_k \text{ constant}$$

- A sufficient condition is to study the stability for each mode k: $\xi_k^n e^{ikj\triangle_x}$

Spectral analysis of discrete Heat equation

- To summarize: the spectral analysis of a discrete PDE studies the behavior of solutions writing $u_j^n = \xi_k^n e^{ikj\triangle_x}$ when n tends to ∞
- Application to the discrete heat equation (FTCS scheme): let's replace u_j^n by $\xi_k^n e^{ikj\triangle_x}$ in Eq. (20):

$$\begin{aligned}
\xi_k^{n+1} e^{ikj\triangle_x} &= \xi_k^n e^{ikj\triangle_x} + \alpha(\xi_k^n e^{ik(j+1)\triangle_x} - 2\xi_k^n e^{ikj\triangle_x} \\
&\qquad + \xi_k^n e^{ik(j-1)\triangle_x}) \\
\xi_k &= 1 + \alpha\left(e^{i\triangle_x k} - 2 + e^{-i\triangle_x k}\right) \\
&= 1 + \alpha\left(2\cos(\triangle_x k) - 2\right) \\
&= 1 - 4\alpha \sin^2 \frac{\triangle_x k}{2} \text{ (recall: } \cos 2a = 1 - 2\sin^2 a)
\end{aligned}$$

Stability of discrete Heat equation (FTCS)

- The scheme is stable iff $|\xi_k| < 1$
- If $\xi_k > 0$:

$$\xi_k = 1 - 4\alpha \sin^2 \frac{\triangle_x k}{2} < 1 \quad \forall \alpha > 0$$

- If $\xi_k < 0$:

$$\begin{aligned} \xi_k = 4\alpha \sin^2 \frac{\triangle_x k}{2} - 1 < 1 \quad &\Leftrightarrow \quad 4\alpha \sin^2 \frac{\triangle_x k}{2} < 2 \\ &\Leftrightarrow \quad \alpha < \frac{1}{2} \end{aligned}$$

- CFL condition[1]: the scheme is stable if $\frac{c\triangle_t}{\triangle_x^2} < \frac{1}{2}$
- Example: with $\triangle_x = \triangle_t = 1$, the condition is $c < \frac{1}{2}$
- Practically $\triangle_x$ is fixed, $\triangle_t$ must be chosen in order to respect the CFL condition

[1] Courant-Freidrichs-Lewy

Stability of discrete Heat equation (FTCS)

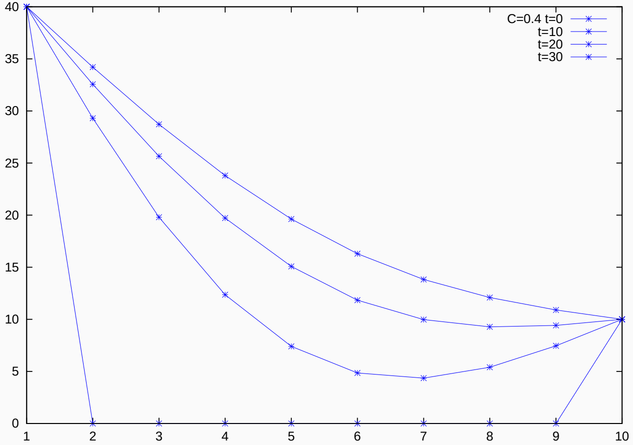


Figure 27: $\alpha = 0.4$, the scheme is stable

Stability of discrete Heat equation (FTCS)

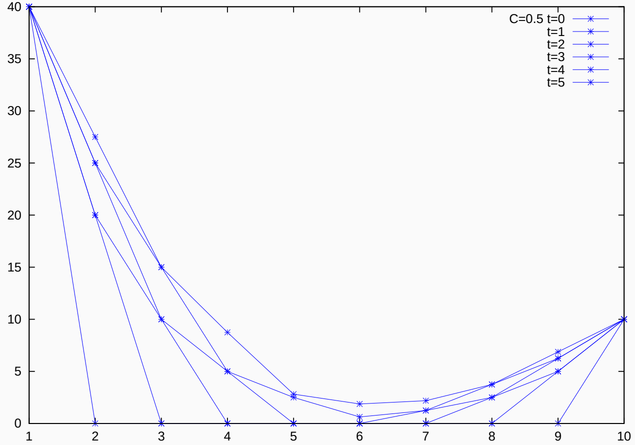


Figure 28: $\alpha = 0.5$, the scheme is locally unstable

Stability of discrete Heat equation (FTCS)

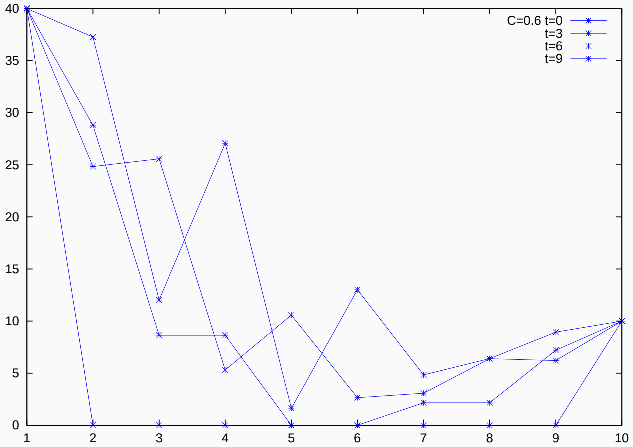


Figure 29: $\alpha = 0.6$, the scheme is unstable

Discrete Heat equation: implicit scheme

- Consider now a backward in time and centered in space scheme (BTCS):
$$\frac{u_j^n - u_j^{n-1}}{\triangle_t} = c\,\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\triangle_x^2}$$
- Leads to the following scheme:
$$-\alpha u_{j+1}^n + (1+2\alpha)u_j^n - \alpha u_{j-1}^n = u_j^{n-1} \tag{21}$$
- It is an **implicit** scheme that requires a matrix inversion: $AU^n = U^{n-1}$ with A tri-diagonal

$$A = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & & & & \\ -\alpha & 1+2\alpha & -\alpha & 0 & \cdots & & & \\ & & \cdots & & & & & \\ & & & & & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix}$$

Discrete Heat equation: implicit scheme

- Error:

$$-\alpha\epsilon_{j+1}^{n} + (1+2\alpha)\epsilon_j^n - \alpha\epsilon_{j-1}^{n} = \epsilon_j^{n-1}$$

- Spectra analysis:

$$\begin{aligned}
&\xi_k\left(-\alpha e^{i\triangle k} + (1+2\alpha) - \alpha e^{-i\triangle k}\right) = 1\\
&\xi_k = \frac{1}{1+2\alpha-2\alpha\cos(\triangle k)}\\
&\quad = \frac{1}{1+4\alpha\sin^2\frac{k\triangle}{2}}\\
&|\xi_k| < 1
\end{aligned}$$

- The scheme is unconditionally stable

Discrete Heat equation: implicit scheme

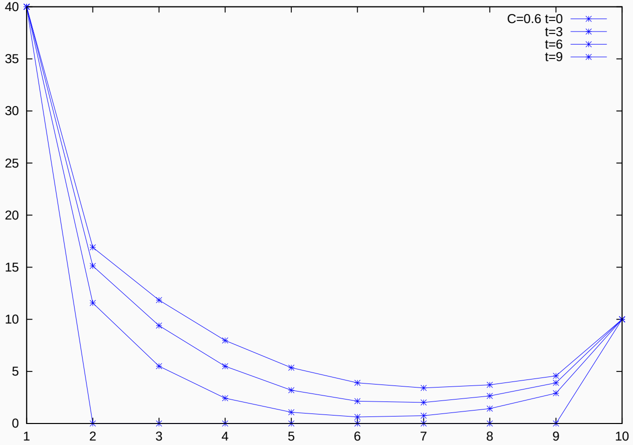


Figure 30: $\alpha = 0.6$, the scheme remains stable

Conclusion on discrete Heat equation

- Pro: the implicit scheme (BTCS) is always stable!
- Cons:
  - require the inversion of a matrix (possibly huge for 2-D diffusion)
  - practically the inversion of huge matrix relies on iterative algorithms (that do not provide an exact solution)
  - For non linear diffusion, the matrix depends on image configuration and changes at each iteration
- Conclusion;
  - if the CFL condition is verified, prefer explicit scheme (more accurate)
  - in other cases, use implicit scheme

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Perona & Malik diffusion: principle

- Founding work on non linear scale space come from [Perona and Malik, 1990]: “Space scale and Edge Detection Using Anisotropic Diffusion”
- The problem is formulated in an axiomatic way:
  - causality principle
  - edges are localized for all scales
  - homogeneous regions are smoothed
- This leads to diffusion whose characteristics depend on local configuration of image values
- Consequently: the PDE describing such a diffusion is no more linear

Perona & Malik diffusion: formulation

- Consider the following PDE:

$$\begin{aligned}
\frac{\partial L}{\partial t}(\mathrm{x}, t) &= \nabla \cdot (c(\mathrm{x}, t)\nabla L(\mathrm{x}, t)) \quad \mathrm{x} \in \mathbb{R}^2, t > 0 \qquad (22)\\
&= \frac{\partial}{\partial x}\left(c(\mathrm{x}, t)\frac{\partial L(\mathrm{x}, t)}{\partial x}\right) + \frac{\partial}{\partial y}\left(c(\mathrm{x}, t)\frac{\partial L(\mathrm{x}, t)}{\partial y}\right)\\
L(\mathrm{x}, 0) &= I(\mathrm{x}) \qquad (23)
\end{aligned}$$

- The right member of (22) may write:

$$\frac{\partial L}{\partial t}(\mathrm{x}, t) = c(\mathrm{x}, t)\nabla^2 L(\mathrm{x}, t) + \nabla c(\mathrm{x}, t).\nabla L(\mathrm{x}, t)$$

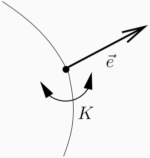
- If $c(\mathrm{x}, t) = c$ then $\nabla c(\mathrm{x}, t) = 0$, and we retrieve the homogeneous heat equation
- c: diffusion coefficient, ruling the "speed" diffusion
  - $c \simeq 0$: no diffusion (stationary), image is locally preserved
  - c is constant, homogeneous diffusion, image is locally smoothed

Choice for c (1)

- According to Perona & Malik: choose c such as
  - $c(\mathrm{x}, t) = 1$ over homogeneous regions: strong smoothing
  - $c(\mathrm{x}, t) = 0$ over edges area: no smoothing
- Let $E(\mathrm{x}, t)$ be an oriented edge map at scale t:

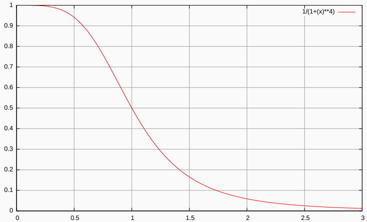
$$E(\mathrm{x}, t) = \begin{cases} K\vec{e}(\mathrm{x}, t) & \text{if x edge} \\ 0 & \text{otherwise} \end{cases}$$

- $\vec{e}$ vector orthogonal to the edge
- K a contrast parameter representing the gray scale value difference between regions adjacent to the edge



Choice for c (2)

- Typically choice: $c(\mathrm{x}, t) = g(\|E(\mathrm{x}, t)\|)$, i.e. an *isotropic* diffusion (doesn't depend on the gradient orientation)
- E depends on L: equation (22) no more linear
- g is a fast decreasing function, for example:



- Perona & Malik's choice for the edges map: $E(\mathrm{x}, t) = \nabla L(\mathrm{x}, t)$

Choice for g (1)

- The solutions of (22) verify the causality principle (derive from a general theorem for parabolic PDEs, see P&M paper)
- Edges enhancement with increasing scale: in 1-D, Equation (22) reduces to

$$\frac{\partial L}{\partial t} = \frac{\partial}{\partial x}\left(c\frac{\partial L}{\partial x}\right)$$

  - Behavior of edges: $\frac{\partial}{\partial t}\left(\frac{\partial}{\partial x}L\right) = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial t}L\right) = \frac{\partial^2}{\partial x^2}\left(c\frac{\partial}{\partial x}L\right)$
    - we chose $c = g(L_x)$
    - let's denote $\phi(L_x) = g(L_x)L_x$ (with $L_x = \frac{\partial}{\partial x}L$), ϕ is called "flux"

$$\begin{aligned}\frac{\partial}{\partial t}L_x &= \frac{\partial^2}{\partial x^2}\phi(L_x)\\ &= \frac{\partial}{\partial x}(L_{xx}\phi'(L_x))\\ &= L_{xxx}\phi'(I_x) + L_{xx}^2\phi''(L_x)\end{aligned}$$

Choice for g (2)

- Behavior of edges:

$$\frac{\partial}{\partial t} L_x = L_{xxx}\phi'(L_x) + L_{xx}^2\phi''(L_x)$$

  - Let an edge such as $L_x > 0$, then $L_{xx} = 0$, and $L_{xxx} < 0$, and

$$\frac{\partial}{\partial t} L_x = L_{xxx}\phi'(L_x)$$

  - Two cases:
    1. $\phi'(L_x) > 0 \Rightarrow \frac{\partial}{\partial t} L_x < 0$: $t \mapsto L_x$ is a decreasing function: attenuation of the edges with increasing scale
    2. $\phi'(L_x) < 0 \Rightarrow \frac{\partial}{\partial t} L_x > 0$: enhancement of the edges with increasing scale
  - If ϕ increases: edges are enhanced and better localized
  - But: the slope should be moderated in order to respect the Causality Principle

Choice for g (3)

- The choice of g is guided by the behavior of ϕ ($\phi(x) = xg(x)$) to obtain an enhancement of edges
- An admissible choice for ϕ, a decreasing function beyond a contrast threshold K:

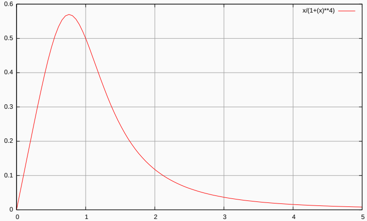


Figure 31: Choice for ϕ, K is the absciss of maximal value

- if $|L_x| > K \Rightarrow \frac{\partial}{\partial t} L_x > 0$: edges are enhanced
- if $|L_x| < K$: edges are attenuated

Choice for g (4)

- Tukey conductivity:
$$g(x) = e^{-\left(\frac{x}{K}\right)^2}$$
- Lorentz conductivity:
$$g(x) = \frac{1}{1 + \left(\frac{x}{K}\right)^{1+\alpha}} \quad \alpha > 0$$
- Function close to 1 in the neighborhood of 0 and fast decreasing. Inflection point at $x = K$.
- Then $\phi(x) = xg(x) \approx x$ in the neighborhood of 0 and decreasing beyond K
- Many other choices are possible

Discretizing P&M equation (1)

- We denote $L(x_0 + i, y_0 + j, k\triangle t) = L^k_{i,j}$, same for g (step space set to 1: pixel unit)
- $\frac{\partial}{\partial t} L \simeq \frac{L^{k+1}_{i,j} - L^k_{i,j}}{\triangle t}$
- $\nabla \cdot (g \nabla L) = \nabla g . \nabla L + g \nabla^2 L$
- ∇g and ∇L are approximated by a forward difference
- $\nabla^2 L = L_{xx} + L_{yy}$ is approximated by a centered difference
- $g(x, y, t) = g(\|\nabla L(x, y, t)\|)$: g depends on the current time iteration k

Discretizing P&M equation (2)

- The right member writes:

$$\begin{aligned} & g^k_{i,j}(L^k_{i,j+1} + L^k_{i,j-1} + L^k_{i+1,j} + L^k_{i-1,j} - 4L^k_{i,j}) \\ +(g^k_{i+1,j} - g^k_{i,j})(L^k_{i+1,j} - L^k_{i,j}) &+ (g^k_{i,j+1} - g^k_{i,j})(L^k_{i,j+1} - L^k_{i,j}) \\ = g^k_{i,j}(L^k_{i-1,j} - L^k_{i,j}) &+ g^k_{i,j}(L^k_{i,j-1} - L^k_{i,j}) \\ +g^k_{i+1,j}(L^k_{i+1,j} - L^k_{i,j}) &+ g^k_{i,j+1}(L^k_{i,j+1} - L^k_{i,j}) \end{aligned}$$

- Final numerical scheme (using P&M notation):

$$L^{k+1}_{i,j} = L^k_{i,j} + \triangle t[C_N \cdot \nabla_N L + C_S \cdot \nabla_S L + C_E \cdot \nabla_E L + C_W \cdot \nabla_W L]^k_{i,j}$$

with:

$$\begin{aligned} [\nabla_N L]^k_{i,j} &= L^k_{i-1,j} - L^k_{i,j} & [\nabla_E L]^k_{i,j} &= L^k_{i,j+1} - L^k_{i,j} \\ [\nabla_S L]^k_{i,j} &= L^k_{i+1,j} - L^k_{i,j} & [\nabla_W L]^k_{i,j} &= L^k_{i,j-1} - L^k_{i,j} \\ [C_N]^k_{i,j} &= g^k_{i,j} & [C_E]^k_{i,j} &= g^k_{i,j+1} \\ [C_S]^k_{i,j} &= g^k_{i+1,j} & [C_W]^k_{i,j} &= g^k_{i,j} \end{aligned}$$

Discretizing P&M equation (3): dual grid and simplification

- The function $g = g(\nabla I)$ should also be discretized. Contours can be localized on the *dual* grid:

$$C_N = g(\|\nabla L^k_{i+\frac{1}{2},j}\|) \qquad C_E = g(\|\nabla L^k_{i,j+\frac{1}{2}}\|)$$
$$C_S = g(\|\nabla L^k_{i-\frac{1}{2},j}\|) \qquad C_W = g(\|\nabla L^k_{i,j-\frac{1}{2}}\|)$$

  $L^k_{i+\frac{1}{2},j}$ may be obtained by linear interpolation

- P&M simplify as:

$$C_N = g(\|\nabla_N u^k_{i,j}\|) \qquad C_E = g(\|\nabla_E u^k_{i,j}\|)$$
$$C_S = g(\|\nabla_S u^k_{i,j}\|) \qquad C_W = g(\|\nabla_W u^k_{i,j}\|)$$

Experimenting Perona & Malik diffusion

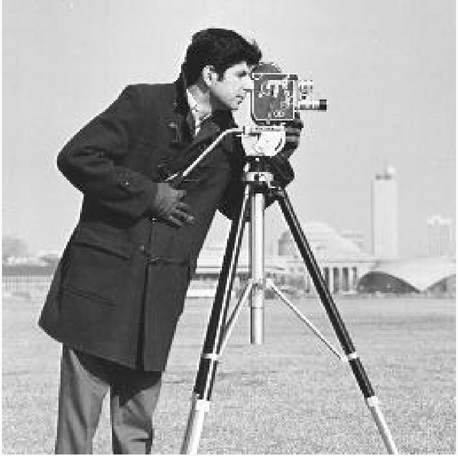


Figure 32: Original image

Experimenting Perona & Malik diffusion

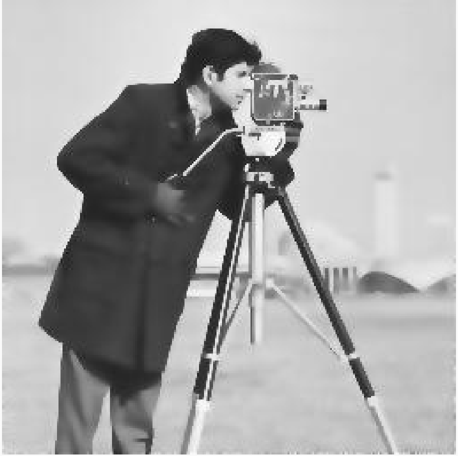


Figure 32: Lorentz conductivity, 10 iterations, $K = 20$

Experimenting Perona & Malik diffusion

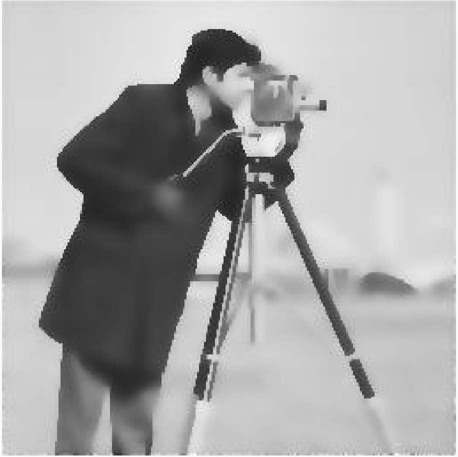


Figure 32: Lorentz conductivity, 30 iterations, $K = 20$

Experimenting Perona & Malik diffusion

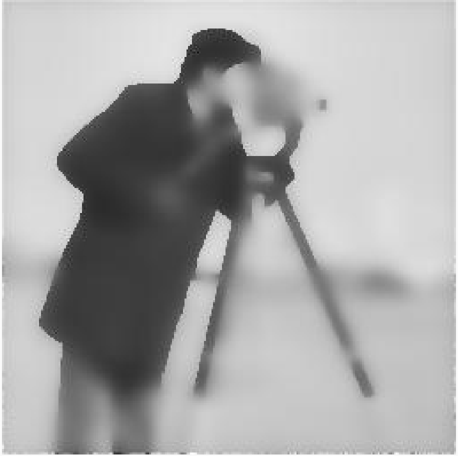


Figure 32: Lorentz conductivity, 100 iterations, $K = 20$

Importance of parameter K

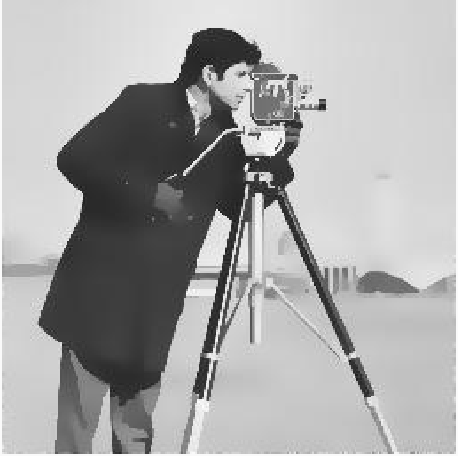


Figure 33: Lorentz conductivity, 100 iterations, $K = 5$

Importance of parameter K

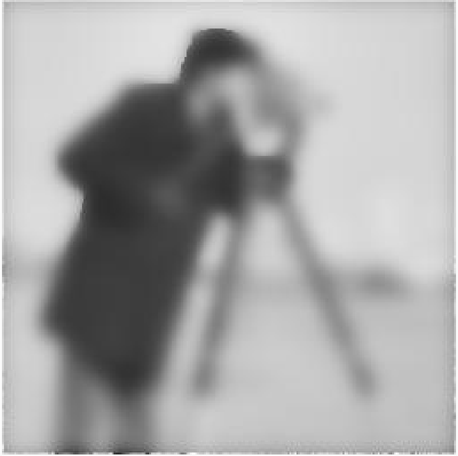


Figure 33: Lorentz conductivity, 100 iterations, $K = 30$

Instability of the scheme

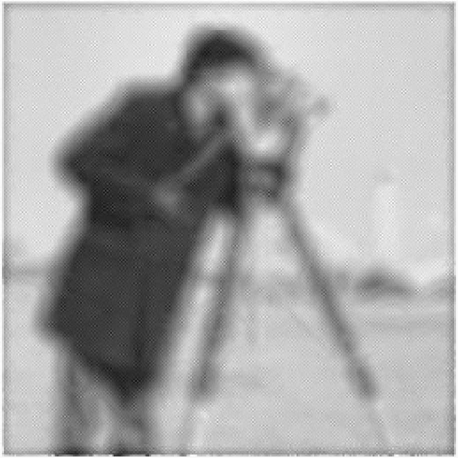


Figure 34: Lorentz conductivity, 100 iterations, $K = 20, \triangle t = 0.5$

Instability of the scheme

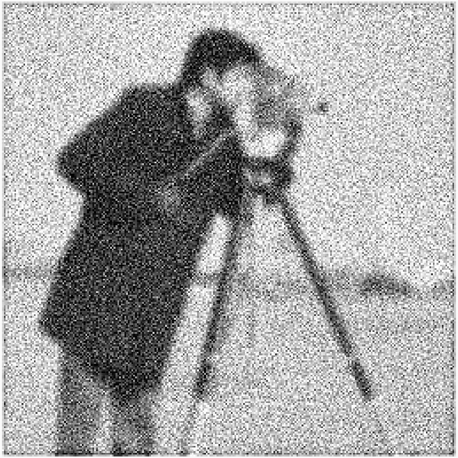


Figure 34: Lorentz conductivity, 10 iterations, $K = 20, \triangle t = 4$

Conclusion on P&M diffusion

- Isotropic diffusion guided by image configurations: image is preserved over edges, elsewhere image is smoothed
- The scheme is explicit and then unstable for some choices of parameters
- A semi-implicit and regularized scheme may be found in [Catté et al., 1992]

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Anisotropic diffusion

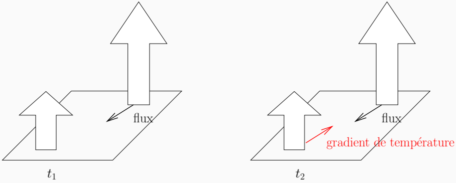
- Isotropic diffusion: diffusion do not depends on the direction (but direction of what?)
- Anisotropic: diffusion depends on the direction
- Physical principles:
    - the same equations describe the diffusion of heat and the diffusion of chemical species
    - particles in a high concentration areas migrate to the lower concentration area: diffusion has a direction

Anisotropic diffusion: again inspired by physics

- Fick's laws:
  1. Flux is the direction of matter transport
  2. Variation of heat or concentration (of a given chemical specie) is equal to the balance of incoming and outgoing fluxes
- Mathematical formalization for $u : \mathbb{R}^3 \to \mathbb{R}$:

$$\frac{\partial u}{\partial t} = -\nabla cdot(\underbrace{-D\nabla u}_{flux}) \tag{24}$$

  - D is a tensor of diffusion (a symmetric and positive definite matrix of size 3 × 3)



2-D anisotropic diffusion

- For an image: $L : \mathbb{R}^2 \to \mathbb{R}$, let's consider:

$$\frac{\partial L}{\partial t} = \nabla \cdot (D \nabla L) \tag{25}$$

- If $D = \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix} \Rightarrow$ isotropic diffusion:

$$\frac{\partial L}{\partial t} = \nabla \cdot (c I_d \nabla L) = \nabla \cdot (c \nabla L)$$

- D is symmetric positive definite, meaning that:
  $D = R \Lambda R^T$ with $\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ and $\lambda_{1,2} \geq 0$
  - R is an orthogonal matrix representing a change of basis: from the canonical basis to the basis of eigenvectors $(\vec{v}_1, \vec{v}_2)$ of D
  - R is a rotation mapping the canonical vector $\vec{i}$ to $\vec{v}_1$, and $\vec{j}$ to $\vec{v}_2$
  - $\lambda_{1,2}$ are the associated eigenvalues

The tensor D

- D is a linear mapping; applied on $\vec{v}_1$, it comes:

$$\begin{aligned} D\vec{v}_1 &= R\Lambda R^T \vec{v}_1 = R\Lambda\vec{i} \\ &= R\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \lambda_1 R\vec{i} \\ &= \lambda_1\vec{v}_1 \end{aligned}$$

  remember that $R^T = R^{-1}$

- and: $D\vec{v}_2 = \lambda_2\vec{v}_2$
  - vectors in the direction of $\vec{v}_1$ are scaled by λ_1
  - vectors in the direction of $\vec{v}_2$ are scaled by λ_2
  - Any vector $\vec{u}$ writes $\vec{u} = c_1\vec{v}_1 + c_2\vec{v}_2$ and then $D\vec{u} = \lambda_1 c_1\vec{v}_1 + \lambda_2 c_2\vec{v}_2$

How to choice D? A first try

- Idea: in the basis $(\vec{v}_1, \vec{v}_2)$ one can control the diffusion in the direction of image gradient and edges
- Get R as:

  $$R = \frac{1}{\|\nabla L\|}\begin{pmatrix} L_x & -L_y \\ L_y & L_x \end{pmatrix}$$

  i.e. a rotation of angle the direction of ∇L
- but in Eq. (25) D applies on ∇L:

  $$D\nabla L = R\Lambda R^T \nabla L = R\Lambda\vec{i} = \lambda_1 \nabla L$$

  and then $\frac{\partial L}{\partial t} = \lambda_1 \nabla^2 L$
- This is the linear isotropic diffusion! Bad idea!

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Edge Enhancing [Weickert, 1998]

- Consider $\nabla_\sigma L$ (gradient of u at scale $\sigma > 0$), the "local" gradient:

$$\begin{aligned} D_\sigma &= R_\sigma \Lambda R_\sigma^T \\ R_\sigma &= \frac{1}{\|\nabla_\sigma L\|} \begin{pmatrix} L_x^\sigma & -L_y^\sigma \\ L_y^\sigma & L_x^\sigma \end{pmatrix} \end{aligned}$$

with $L_.^\sigma = L \star \frac{\partial G_\sigma}{\partial .}$

- Let's apply D_σ on ∇L:

$$\begin{aligned} D_\sigma &= \frac{1}{\|\nabla_\sigma L\|^2} \begin{pmatrix} \lambda_1 (u_x^\sigma)^2 + \lambda_2 (u_y^\sigma)^2 & (\lambda_1 - \lambda_2) L_x^\sigma L_y^\sigma \\ (\lambda_1 - \lambda_2) u_x^\sigma L_y^\sigma & \lambda_2 (L_x^\sigma)^2 + \lambda_1 (L_y^\sigma)^2 \end{pmatrix} \\ D_\sigma \nabla L &= \frac{1}{\|\nabla_\sigma L\|^2} \left(\lambda_1 \nabla_\sigma L \nabla_\sigma L^T + \lambda_2 \nabla_\sigma^\perp L \nabla_\sigma^\perp L^T \right) \nabla L \end{aligned}$$

with $\nabla_\sigma^\perp L = \begin{pmatrix} L_y \\ -L_x \end{pmatrix}$

- In the general case: $\nabla_\sigma^\perp L^T \nabla L \neq 0$

Edge Enhancing (2)

- Eigenvectors of D_σ:

$$\vec{v}_1 = \frac{\nabla_\sigma u}{\|\nabla_\sigma u\|} \text{ and } \vec{v}_2 = \vec{v}_1^\perp$$

- Expression of ∇L in the basis of eigenvalues of D:

$$\nabla L = c_1 \vec{v}_1 + c_2 \vec{v}_2$$

- with $\vec{v}_1 . \nabla L = c_1$ and $\vec{v}_2 . \nabla L = c_2$
- then: $R_\sigma^T \nabla L = \begin{pmatrix} \vec{v}_1^T \\ \vec{v}_2^T \end{pmatrix} \nabla L = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$
- and: $\Lambda \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} c_1 \lambda_1 \\ c_2 \lambda_2 \end{pmatrix}$ and $R_\sigma^T \begin{pmatrix} c_1 \lambda_1 \\ c_2 \lambda_2 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} c_1 \lambda_1 \\ c_2 \lambda_2 \end{pmatrix}$
- finally:

$$D_\sigma \nabla L = c_1 \lambda_1 \vec{v}_1 + c_2 \lambda_2 \vec{v}_2$$

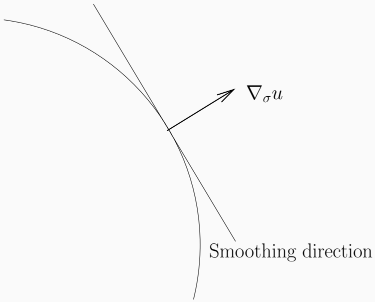
Edge Enhancing (3)

- (c_1, c_2) characterizes ∇L
- (λ_1, λ_2) characterizes D_σ:
  1. if σ close to 0: $\nabla_\sigma L \to \nabla L$ and $c_1 \to 1$ and $c_2 \to 0$: the diffusion is almost isotropic, diffusivity is ruled by λ_1
  2. if σ is high, in general $\nabla_\sigma L$ is not colinear with ∇L and $c_2 \gg 0$: the diffusion direction depends on values of λ_1 and λ_2
  3. if $\lambda_1 = \lambda_2$, then $D\nabla L = \lambda_1(c_1\vec{v}_1 + c_2\vec{v}_2) = \lambda_1 \nabla L$: the diffusion is again isotropic!
  4. if $\lambda_1 = 0$ and $\lambda_2 \gg 0$, then $\nabla_\sigma L \perp \nabla L$: the diffusion direction is oriented along edges, diffusion is not more isotropic

Edge Enhancing (4)

- to summarize:
  - λ_1 tunes diffusion in the direction of $\nabla_\sigma L$
  - λ_2 tunes diffusion in the orthogonal direction of $\nabla_\sigma L$
  - if $\lambda_1 \geq \lambda_2$, diffusion is closely isotropic
  - if $\lambda_1 < \lambda_2$, diffusion is anisotropic
- Possible choice for λ_1 et λ_2 ([Weickert, 1998]):
  - $\lambda_2 = e^{-\frac{\|\nabla_\sigma L\|^2}{k^2}}$
  - $\lambda_1 = \frac{1}{5}\lambda_2 \ll \lambda_2$
    $\Rightarrow$ diffusion is oriented along edges
  - the diffusion is non linear w.r.t. to L (why?)

Edge Enhancing (5)



- image is smoothed along edges (high value of λ_2) ...
- ... and not in the direction of $\nabla_\sigma L$ ($\lambda_1 \simeq 0$)
- In homogeneous regions: $\lambda_1 \simeq 0, \lambda_2 \simeq 0$: no diffusion, image is preserved

Discretizing Edge Enhancing (1)

- Let $D = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$

$$\begin{aligned} L_t &= \nabla \cdot (D \nabla L) \\ &= \frac{\partial}{\partial x}\left(a\frac{\partial L}{\partial x}\right) + \frac{\partial}{\partial x}\left(b\frac{\partial L}{\partial y}\right) + \frac{\partial}{\partial y}\left(b\frac{\partial L}{\partial x}\right) + \frac{\partial}{\partial y}\left(c\frac{\partial L}{\partial y}\right) \end{aligned}$$

- Terms $\frac{\partial}{\partial x}\left(a\frac{\partial L}{\partial x}\right)$ et $\frac{\partial}{\partial y}\left(c\frac{\partial L}{\partial y}\right)$ had already been addressed by P&M
- Remains the crossed terms:
    - $b\frac{\partial u}{\partial y} \simeq b_{i,j}\frac{L_{i,j+1}-L_{i,j-1}}{2} = f_{i,j}$
    - $\frac{\partial}{\partial x}\left(b\frac{\partial L}{\partial y}\right) \simeq \frac{f_{i+1,j}-f_{i+1,j}}{2} =$ $\frac{1}{4}\left(b_{i+1,j}(L_{i+1,j+1} - L_{i+1,j-1}) - b_{i-1,j}(L_{i-1,j} - L_{i-1,j-1})\right)$
    - same discretization of the second crossed term
- These choices lead to a stable numerical scheme

Discretizing Edge Enhancing (2)

- After factoring according to L terms, one finds:

$$\begin{aligned} L^{k+1}_{i,j} = L^k_{i,j} + \triangle t \Bigg[&-\frac{b_{i-1,j}+b_{i,j+1}}{4} L^k_{i-1,j+1} + \frac{c_{i,j+1}+c_{i,j}}{2} L^k_{i,j+1} \\ &+ \frac{b_{i+1,j}+b_{i,j+1}}{4} L^k_{i+1,j+1} + \frac{a_{i-1,j}+a_{i,j}}{2} L^k_{i-1,j} \\ &- \frac{a_{i-1,j}+2a_{i,j}+a_{i+1,j}+c_{i,j-1}+2c_{i,j}+c_{i,j+1}}{2} L^k_{i,j} \\ &+ \frac{a_{i+1,j}+a_{i,j}}{2} L^k_{i+1,j} + \frac{b_{i-1,j}+b_{i,j-1}}{4} L^k_{i-1,j} \\ &+ \frac{c_{i,j-1}+c_{i,j}}{2} L^k_{i,j-1} - \frac{b_{i+1,j}+b_{i,j-1}}{4} L^k_{i+1,j-1} \Bigg] \end{aligned}$$

Plugging D_σ into the numerical scheme

- Recall:

$$\begin{aligned} D_\sigma &= R_\sigma \Lambda R_\sigma^T \\ D_\sigma &= \frac{1}{\|\nabla_\sigma L\|^2} \begin{pmatrix} \lambda_1 (L_x^\sigma)^2 + \lambda_2 (L_y^\sigma)^2 & (\lambda_1 - \lambda_2) L_x^\sigma L_y^\sigma \\ (\lambda_1 - \lambda_2) L_x^\sigma L_y^\sigma & \lambda_2 (L_x^\sigma)^2 + \lambda_1 (L_y^\sigma)^2 \end{pmatrix} \end{aligned}$$

- then:

$$\begin{aligned} a &= \left(\lambda_1 (L_x^\sigma)^2 + \lambda_2 (L_y^\sigma)^2\right) / \|\nabla L^\sigma\|^2 \\ b &= (\lambda_2 - \lambda_1) L_x^\sigma u_y^\sigma / \|\nabla L^\sigma\|^2 \\ c &= \left(\lambda_2 (L_x^\sigma)^2 + \lambda_1 (L_y^\sigma)^2\right) / \|\nabla L^\sigma\|^2 \end{aligned}$$

Experimenting Edge Enhancing diffusion (1)

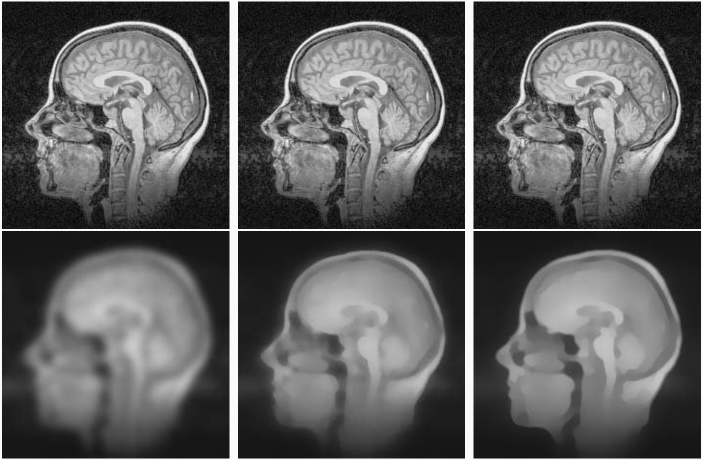


Figure 35: Comparaison Gaussian/P&M/Edge enhancing

Experimenting Edge Enhancing diffusion (2)

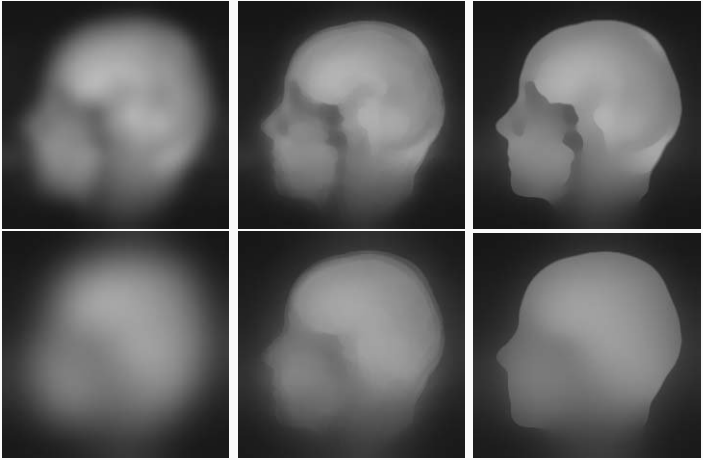
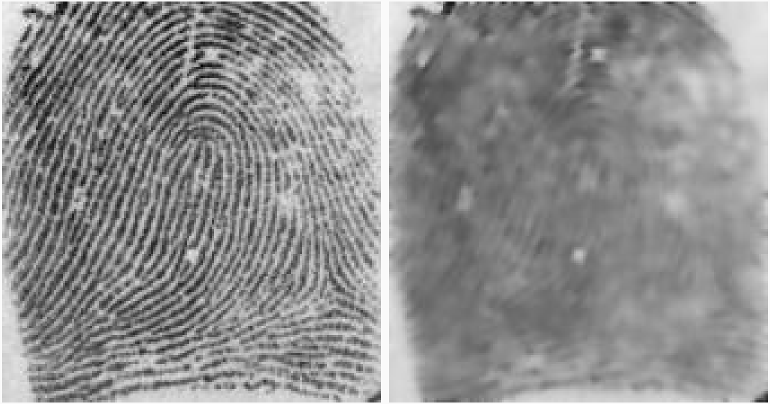


Figure 36: Comparaison Gaussian/P&M/Edge enhancing

Edge Enhancing issue (1)

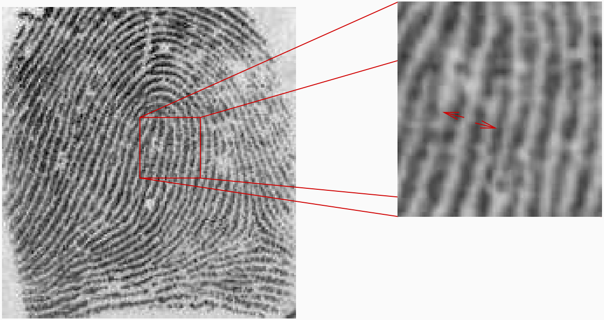


(a) Original image

(b) Edge Enhancing

Edge Enhancing issue (2)

- Why?
- How are gradients oriented?



- In a same neighborhood, gradients may be opposed and compensate each other!
- How to fix that?

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Coherence Enhancing [Weickert, 1998]

- Need to diffuse along edges but not in an oriented way
- Compute the (non oriented) direction of an edge
- Let's define the tensor S "local orientation":

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{pmatrix} = \begin{pmatrix} L_x^s L_x^s \star G^\sigma & L_x^s L_y^s \star G^\sigma \\ L_x^s L_y^s \star G^\sigma & L_y^s L_y^s \star G^\sigma \end{pmatrix}$$

- Applied on ∇L and $-\nabla L$, S gives the same value:

$$\begin{pmatrix} L_x^s L_x^s \star G^\sigma & L_x^s L_y^s \star G^\sigma \\ L_x^s L_y^s \star G^\sigma & L_y^s L_y^s \star G^\sigma \end{pmatrix}$$

Coherence Enhancing (2)

- The tensor D is defined as follow:

$$D = R \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} R^T$$

  and R is formed by the eigenvectors of S

- Eigenvectors of S can be determined, D writes:

$$\begin{aligned}
d_{11} &= \frac{1}{2}\left(c_1 + c_2 + \frac{(c_1 - c_2)(s_{11} - s_{22})}{\alpha}\right) \\
d_{22} &= \frac{1}{2}\left(c_1 + c_2 - \frac{(c_1 - c_2)(s_{11} - s_{22})}{\alpha}\right) \\
d_{12} &= \frac{(c_2 - c_1)s_{12}}{\alpha}
\end{aligned}$$

  with $\alpha = \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}$

Coherence Enhancing (3)

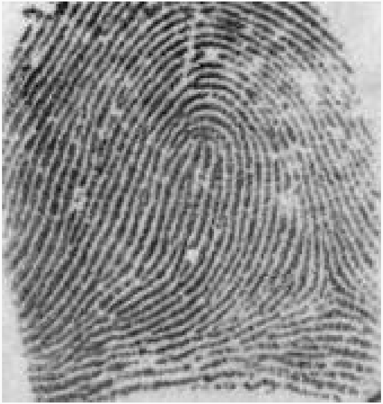
- Weickert chooses for c_1 and c_2:

$$\begin{aligned} c_1 &= \max(0.01, 1 - e^{-(\lambda_1 - \lambda_2)^2/k^2}) \\ c_2 &= 0.02 \end{aligned}$$

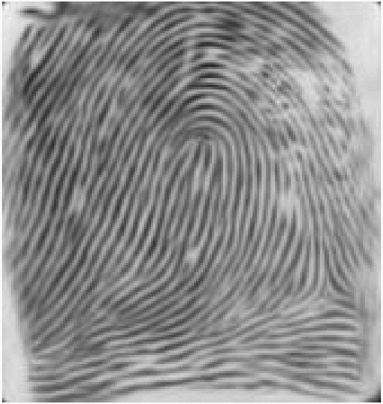
with λ_1 and λ_2 the eigenvalues of S:

$$\begin{aligned} \lambda_1 &= \frac{1}{2}(s_{11} + s_{22} + \alpha) \\ \lambda_2 &= \frac{1}{2}(s_{11} + s_{22} - \alpha) \end{aligned}$$

Experimenting Coherence Enhancing (1)



(c) Original image



(d) Coherance Enhancing

Experimenting Coherence Enhancing (2)

- Artistic filtering?

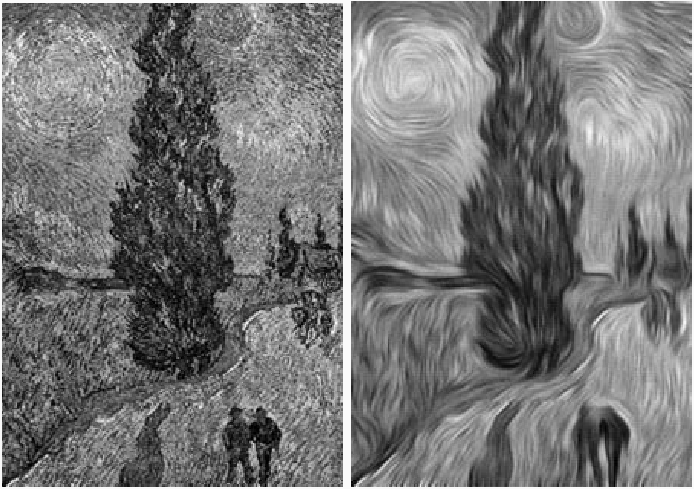


Figure 37: "Route avec cyprès et Ciel étoilé"

Experimenting Coherence Enhancing (3)

- Artistic filtering?

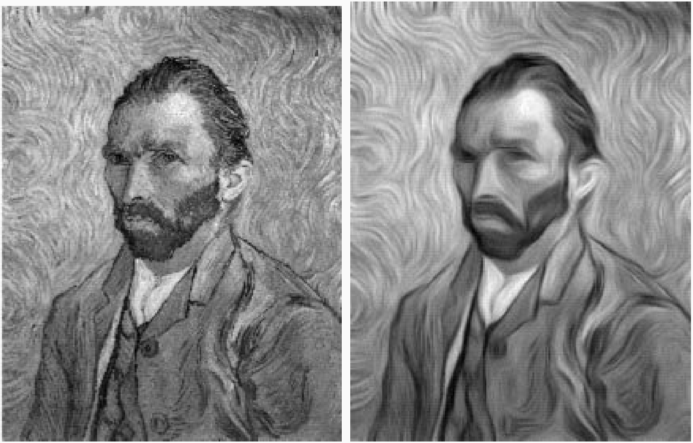


Figure 38: Exacerbed Van Gogh

Experimenting Coherence Enhancing (4)

- color images:

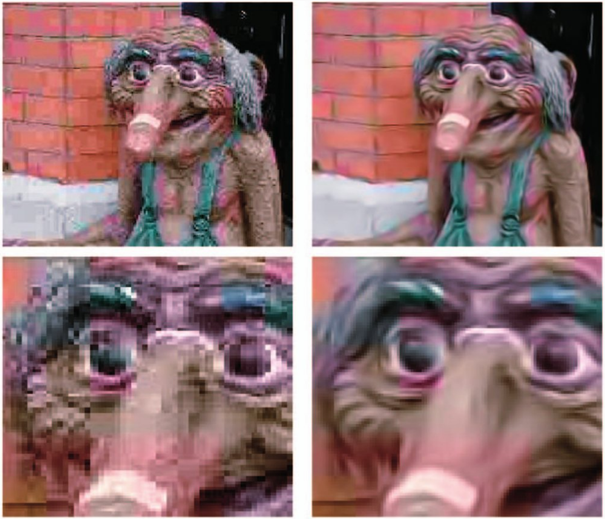


Figure 39: independant diffusion on each channel

Inpainting [Tschumperle and Deriche, 2005]

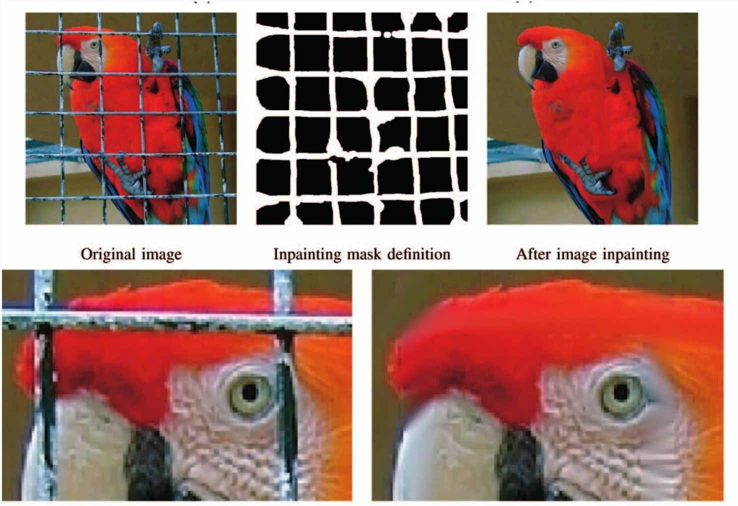


Figure 40: Only missing pixels are written and not read

Content

Part 1: linear scale space

Part 2: non linear scale spaces

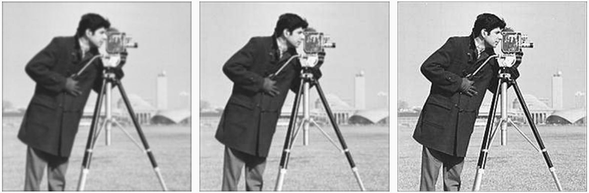
- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

Unblurring images?

- Gaussian blur ⇔ linear and isotropic diffusion
- Question: given L^n (blurred image), can one retrieve L^0?
- May be yes! discretization of heat equation using a retrograde scheme:

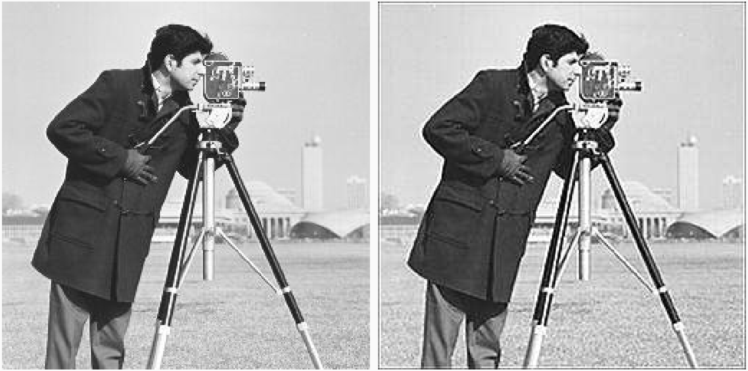
$$\begin{aligned}\frac{L^{k+1} - L^k}{\triangle t} &= \nabla^2 L^{k+1} \\ L^k &= L^{k+1} - \triangle t \nabla^2 L^{k+1}\end{aligned}$$



(a) $\sigma = 1$ (b) 5 itérations (c) 10 itérations

Figure 41: Backward scheme of heat equation

Gaussian unblurring (1)



(a) Blurred image

(b) Unblurring

Gaussian unblurring (2)

- The retrograde scheme is unstable:

Loading data ...

Figure 42: Instability after 32 iterations

Gaussian unblurring (3)

- Example with a stronger blur:

Loading data ...

Figure 43: $\sigma = 2$, 28 iterations

Gaussian unblurring (4)

- But not too much!

Loading data ...

Figure 44: $\sigma = 3$: divergence after 15 iterations

Conclusion on Gaussian unblurring

- The retrograde scheme of heat equation is unconditionally unstable (easy to prove using spectral analysis, exercise to do)
- In a same time, unblurring image implies to enhance / create edges: violation of the Causality Principle
- Rounding errors in the numerical retrograde scheme make appear noise and are amplified with increasing time iterations
- On the contrary, diffusion processes are, by nature, stable and regularizing: numerical errors or noise are smoothed out and eventually disappear

Image unblurring: shock filters (1)

- Let's consider the following PDE:

$$\begin{aligned} L_t &= -\operatorname{sign}(\nabla^2 L)\|\nabla L\| \qquad t > 0 \\ L(\mathrm{x}, 0) &= I(\mathrm{x}) \end{aligned}$$

- Properties:
  - in a neighborhood of a local maxima x_0: $\nabla^2 L(\mathrm{x}, t) < 0 \quad \forall \mathrm{x} \in \mathcal{V}(\mathrm{x}_0)$:

$$L_t = \|\nabla L\|$$

  - in a neighborhood of a local minima x_0: $\nabla^2 u(\mathrm{x}, t) > 0 \quad \forall \mathrm{x} \in \mathcal{V}(\mathrm{x}_0)$:

$$L_t = -\|\nabla L\|$$

- These two cases are equivalent to morphological dilation or erosion
- See [Osher and Rudin, 1990]

Image unblurring: shock filters (2)

- Discretization:

$$L_i^{k+1} - L_i^k = \Delta t \left[- \operatorname{sign}(\nabla^2 L_i^k) \|\nabla L_i^k\| \right]$$

- Variation:

$$L_t = - \operatorname{sign}\left(\frac{\partial L}{\partial \eta^2} \right) \|\nabla L\|$$

  with $\eta = \frac{\nabla L}{\|\nabla u\|}$.

- [Alvarez and Mazorra, 1994]:

$$L_t = - \operatorname{sign}\left(\frac{\partial^2 v}{\partial \eta^2} \right) \|\nabla L\|$$

  with $v = G_\sigma \star L$ and $\frac{\partial}{\partial \eta}$ derivative in the Gradient direction

- Anisotropic shock filter: [Weickert, 2003]

Content

Part 1: linear scale space

Part 2: non linear scale spaces

- Motivations
- Digression: discretization of PDEs
- Perona & Malik diffusion
- Anisotropic diffusion
- Edge Enhancing
- Coherence Enhancing
- Image restoration
- To go further

APPENDIX

To go further with non linear scale spaces (1)

- Unlike the continuous linear case, there are several nonlinear continuous multiscale representations
- It also be possible to describe non linear spaces in an axiomatic way The representation at scale t is obtained with a non linear operator T_t such as:
  - $L_t = T_t(I)$ (representation at scale t of image I)
  - T_t has a semi-group structure
  - T_t respects the Causality Principle
  - T_t is differentiable w.r.t. t
- Let's denote $\partial T = \lim_{t \to 0} \frac{T_t(f) - f}{t}$, the family $(T_t(I))_{t>0}$ is solution of

$$\begin{aligned} L_t &= \partial T(L) \\ L(.,0) &= I \end{aligned}$$

To go further with non linear scale spaces (2)

- Alvarez *et al* characterized the solutions of the previous PDE, see [Alvarez et al., 1993]
- Some interesting examples:
  - if T is linear, the classic diffusion equation is retrieved
  - if T commutes with a map $F(., t)$ monotone increasing, then solutions write:

    $$L_t = |\nabla L| F\left(\nabla . \left(\frac{\nabla L}{|\nabla L|}\right), t\right)$$

    this equation rules the dynamics of a snake evolving with a velocity depending only on the curvature of iso-value lines (implicit snake model)

To go further with non linear scale spaces (3)

- Some interesting examples:
  - Link with mathematical morphology: solutions of

    $$\begin{aligned} L_t &= |\nabla L| \\ L_t &= -|\nabla L| \end{aligned}$$

    with $L(x, 0) = I$ as initial condition are respectively dilatation and erosion of image I with a structuring element tB (B disc of radius 1) $\Rightarrow$ successive dilatation/erosion of an image have the structure of a non linear scale space
  - Regularization: models formalized by a cost function to minimize and embedding a regularization term are equivalent to solve an Alvarez *et al*'s PDE
- More details in `http://pequan.lip6.fr/~bereziat/cours/master/amo/amo-ss-cours3.pdf`

APPENDIX

Alvarez, L., Guichard, F., Lions, P.-L., and Morel, J.-M. (1993).
Axioms and fundamental equations of image processing.
Arch. Rational Mech. Anal., 123:199–257.

Alvarez, L. and Mazorra, L. (1994).
Signal and image restoration using shock filters and anisotropic diffusion.
Journal of Numerical Analysis, 31(2):590–605.

Burt, P. J. (1981).
Fast filters transforms for image processing.
Computer Vision and Image Processing, 16:20–51.

Catté, F., Lions, P.-L., Morel, J.-M., and Coll, T. (1992).
Image selective smoothing and edge detection by nonlinear diffusion.
Journal of Numerical Analysis, 1:182–193.
Regularization of Perona-Malik diffusion.

Crowley, J. L. (1981).

A Representation for Visual Information.
PhD thesis, Carnegie-Mellon University, Robotics Institute, Pittsburgh, Pennsylvania.

Klinger, A. (1971).
Pattern and search statistics.
Optimizing Methods in Statistics.

Koenderink, J. J. (1984).
The structure of images.
Biological Cybernetics, 50:363–370.

Lindeberg, T. (1994).
Scale-Space Theory in Computer Vision.
Kluwer Academic Publishers.

Lindeberg, T. (1998).
Feature detection with automatic scale selection.
International Journal on Computer Vision, 30(2):77–116.

Marr, D. and Hildreth, E. (1980).

Theory of edge detection.
Proc. of the Royal Society of London, B207:187–217.

Osher, S. and Rudin, L. I. (1990).
Feature-oriented image enhancement using shock filters.
Journal of Numerical Analysis, 27(4):919–940.

Perona, P. and Malik, J. (1990).
Space scale and edge detection using anisotropic diffusion.
Transactions on Pattern Analysis and Machine Intelligence, 12(7):629–639.
Article fondateur sur la diffusion non linéaire.

Rosenfeld, A. and Thurston, M. (1971).
Edge and curve detection for visual scene analysis.
IEEE Transaction on Computers, 20(5):563–569.

Tschumperle, D. and Deriche, R. (2005).
Vector-valued image regularization with pdes: A common framework for different applications.

Transactions on Pattern Analysis and Machine Intelligence,
27(4):506–512.

- Uhr, L. (1972).
  Layered 'recognition cone' networks that preprocess, classify and describe.
  IEEE Trans. Comput., pages 759–768.

- Weickert, J. (1998).
  ***Anisotropic diffusion in image processing*.**
  ECMI Series. Teubner-Verlag, Stuttgart.
  ISBN:3-519-02606-6.

- Weickert, J. (2003).
  ***Pattern Recognition*, chapter Coherence-Enhancing Shock Filters, pages 1–8.**
  Lecture Notes in Computer Science. Springer Berlin / Heidelber.
  2781.

- Witkin, A. P. (1983).
  Scale-space filtering.

In *Proc. 8th Int. Joint Conf. Art. Intell.*, pages 1019–1022, Karlsruhe, Germany.

Article fondateur sur les espaces d'échelles (1D).

Proof Eq. (5)

- determine $\int e^{-x^2}dx$:

$$\begin{aligned}\left(\int_{\mathbb{R}} e^{-x^2}dx\right)^2 &= \left(\int_{\mathbb{R}} e^{-x^2}dx\right)\left(\int_{\mathbb{R}} e^{-y^2}dy\right)\\ &= \iint_{\mathbb{R}^2} e^{-x^2-y^2}dxdy\end{aligned}$$

- Change of variable (polar coordinate): $x = r\cos\theta$ et $y = r\sin\theta$

$$\begin{aligned}\left(\int_{\mathbb{R}} e^{-x^2}dx\right)^2 &= \int_{\mathbb{R}^+}\int_0^{2\pi} e^{-r^2}\left|\begin{array}{cc}\cos\theta & \sin\theta\\ -r\sin\theta & r\cos\theta\end{array}\right|drd\theta\\ &= \int\int_0^{2\pi} re^{-r^2}drd\theta = \int_0^{2\pi}d\theta\int re^{-r^2}dr\\ &= 2\pi\left[-\frac{1}{2}e^{-r^2}\right]_{r=0}^{r=+\infty} = \pi\end{aligned}$$

Unicity of Gaussian kernel (1)

- Let $L^{(t)}(\mathrm{x}) = L(\mathrm{x}, t)$
- In Fourier space, equation (11) writes

$$\begin{aligned}
\widehat{L}^{(t)}(\mathrm{w}) &= \widehat{h_t \star I}(\mathrm{w}) = \widehat{h}_t(\mathrm{w})\widehat{I}(\mathrm{w}) \\
\frac{\widehat{L}^{(t)}(\mathrm{w})}{\widehat{I}(\mathrm{w})} &= \widehat{h}_t(\mathrm{w})
\end{aligned}$$

with $\mathrm{w} = (u, v)$ coordinate in Fourier space

- Parameter w is a frequency, then w^{-1} is a period (a length)
- Parameter $\sqrt{t}$ measures a scale (a length)
- Two lengths, then $\mathrm{w}\sqrt{t}$ is without dimension
- Same for $\frac{\widehat{L}^{(t)}(\mathrm{w})}{\widehat{I}(\mathrm{w})}$
- Pi theorem $\Rightarrow$ one can write $\widehat{h}_t(\mathrm{w}) = \widehat{H}(\sqrt{t}\mathrm{w})$

Unicity of Gaussian kernel (2)

- we then have: $\frac{\widehat{L^{(t)}}(\mathrm{w})}{\widehat{I}(\mathrm{w})} = \widehat{H}(\mathrm{w}\sqrt{t})$
- $I = L^{(0)}$ then $\widehat{H}(0) = 1$.
- Semi-group: $\widehat{h}_{\mathbf{t_1}}(\mathrm{w})\widehat{h}_{\mathbf{t_2}}(\mathrm{w}) = \widehat{h}_{\mathbf{t_1}+\mathbf{t_2}}(\mathrm{w})$ then:

$$\widehat{H}(\mathrm{w}\sqrt{t_1})\widehat{H}(\mathrm{w}\sqrt{t_2}) = \widehat{H}(\mathrm{w}\sqrt{t_1+t_2})$$

- Let $\tilde{H}(\mathrm{w}^T\mathrm{w}) = \widehat{H}(\mathrm{w}), \quad \mathrm{w} \in \mathbb{R}^2$, we have:

$$\begin{array}{rcl}
\tilde{H}((\mathrm{w}\sqrt{t_1})^T(\mathrm{w}\sqrt{t_1})) & \times & \\
\tilde{H}((\mathrm{w}\sqrt{t_2})^T(\mathrm{w}\sqrt{t_2})) & = & \tilde{H}((\mathrm{w}\sqrt{t_1+t_2})^T(\mathrm{w}\sqrt{t_1+t_2})) \\
\tilde{H}(t_1\mathrm{w}^T\mathrm{w})\tilde{H}(t_2\mathrm{w}^T\mathrm{w}) & = & \tilde{H}((t_1+t_2)\mathrm{w}^T\mathrm{w}) \\
\tilde{H}(v_1)\tilde{H}(v_2) & = & \tilde{H}(v_1+v_2)
\end{array}$$

with $v_1 = t_1\mathrm{w}^T\mathrm{w}$ and $v_2 = t_2\mathrm{w}^T\mathrm{w}$

Unicity of Gaussian kernel (3)

- Conclusion:

$$\begin{cases} \tilde{H}(0) = 1 \\ \tilde{H}(v_1)\tilde{H}(v_2) = \tilde{H}(v_1 + v_2) \end{cases}$$

- It's the definition of the exponential function ($u \mapsto e^{\alpha u}$)
- Then: $\widehat{H}(\mathrm{w}) = e^{\alpha t \mathrm{w}^T \mathrm{w}}$
- We choose $\alpha < 0$ in order to have $\lim_\infty \widehat{H} = 0$
- The Fourier transform of an Gaussian function is a Gaussian function (with $\alpha > 0$)

$$\int_{\mathbb{R}^2} e^{-\frac{\mathbf{x}^T\mathbf{x}}{\alpha t}} e^{-i\mathrm{w}\mathrm{x}} d\mathrm{x} = e^{-\alpha t \mathrm{w}^T \mathrm{w}}$$

- Normalization: $\int_{\mathbb{R}} e^{-x^2/2t} dx = \sqrt{2\pi}t$, finally we choose $\alpha = \frac{1}{2}$
- Full proof available in the Lindeberg's book [Lindeberg, 1994]