Practice 2:
Image processing with variational approaches

Jean-François Aujol

IMB, Université Bordeaux
351 Cours de la libération, 33405 Talence Cedex, FRANCE

Email : jean-francois.aujol@math.u-bordeaux.fr

1 Denoising with variational approaches

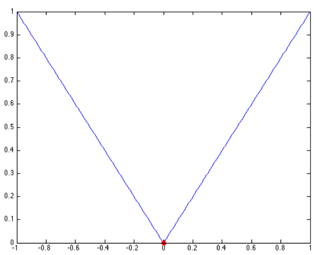
Before coming to variational models, we first recall some properties of convex functionals.

1.1 Minimization of convex and differentiable functionals

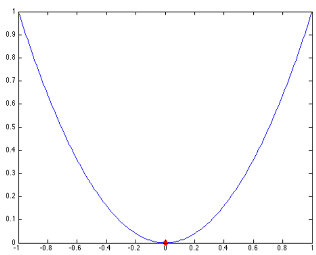
In this practice, we consider $u \in X$, where X is the euclidean space $\mathbb{R}^{m\times n}$. A function $J : X \to \mathbb{R}$ is said convex iff:

$$\forall (u,v) \in X \times X, \quad \mathit{and} \quad \forall t \in [0;1], \quad J(tu + (1-t)v) \leq tJ(u) + (1-t)J(v).$$

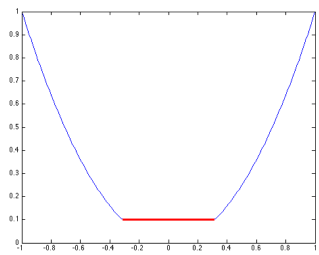
The function is strictly convex if the above inequality is strict $\forall u \neq v$ and $t \in]0;1[$. Classical $1D$ examples (i.e. $m = n = 1$) of convex and strictly convex functions are given below.



$J(u) = |u|$ is convex



$J(u) = ||u||^2$ is strictly convex



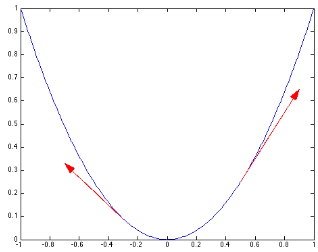
$J(u) = \max(||u||^2, 0.1)$ is convex

Assuming that J is proper, lower semi-continuous and coercive (see Lecture for definitions and details), then a convex function admits at least one global minimizer. If the function is not strictly convex, the minimizers are not necessarily unique (the values of the functions at the minimizers are in red in the previous Figures).

From now, we will only consider differentiable functions J so that $\nabla J(u) \in X$ exists for all $u \in X$ and it is the only vector of X that checks:

$$J(v) \geq J(u) + \langle \nabla J(u), v - u \rangle, \quad \forall v \in X.$$

Notice that ∇ ***here represents the derivative of the function*** J ***with respect to*** u***:*** $\nabla J(u) = \partial_u J(u)$***, and not the spatial gradient as before.*** The vector $\nabla J(u)$ then defines the tangent of J at point u, as illustrated below:



$J(u) = ||u||^2$, the red vectors represent (with a rescaling) $\nabla J(u) = 2u$ at points $u = 0.5$ and $u = -0.3$.

In this differentiable and convex context, we can observe that u is a minimizer of J iff $\nabla J(u) = 0$. The gradient ∇J thus allows to characterize minimizers of J, but it also permits to get closer to one minimizer from a current point u, by going in the opposite direction from $\nabla J(u)$ and thus decrease the function J. Hence, with an adequate time step $\tau > 0$ and with an additional condition on ∇J (discussed in the remark page 13), the well-known gradient descent algorithm:

$$u^{k+1} = u^k - \tau \nabla J(u^k),$$

converges to a global minimizer of J for any $u^0 \in X$.

1.2 Previous PDEs as gradient descent of convex functionals

We recall the PDES:

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} & = \Delta u(t,x) \quad \text{for } t \geq 0 \text{ and } x \in \Omega \\ u(0,x) & = f(x) \quad \text{for } x \in \Omega \\ \frac{\partial u(t,x)}{\partial N} & = 0 \quad \text{for } t > 0 \text{ and } x \in \partial\Omega, \end{cases} \tag{1.1}$$

where Δ is the Laplacian operator, f is the initial temporal condition and $\frac{\partial u}{\partial N} = 0$ are Neumann boundary conditions.

and its discrete version:

$$\begin{cases} u_{i,j}^{k+1} & = u^k + \delta_t (\Delta u^k)_{ij} \\ u_{ij}^0 & = f_{ij}, \end{cases} \tag{1.2}$$

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} & = \operatorname{div}\Big(g\big(||\nabla u(t,x)||\big)\nabla u(t,x)\Big) \quad \text{for } t \geq 0 \text{ and } x \in \Omega \\ u(0,x) & = f(x) \quad \text{for } x \in \Omega \\ \frac{\partial u(t,x)}{\partial N} & = 0 \quad \text{for } t > 0 \text{ and } x \in \partial\Omega, \end{cases} \tag{1.3}$$

where g is a decreasing function from $\mathbb{R}_+$ to $\mathbb{R}_+$.

In the following we will consider

$$g(\xi) = \frac{1}{\sqrt{(\xi/\alpha)^2 + 1}}. \tag{1.4}$$

From now on, we will consider the discrete framework so that the following sums on Ω corresponds to a discretization of the continuous integrals on Ω in the Lecture. Let us now consider the convex function $J_H(u) = \frac{1}{2}\sum_{x\in\Omega} ||\nabla u(x)||^2 = \frac{1}{2}||\nabla u||_Y^2$. From calculus of variations, one obtains $\nabla J_H(u) = -\mathrm{div}\,(\nabla u) = -\Delta u \in X$.

We can then apply the gradient descent algorithm in order to compute a minimizer of this convex function. Initializing $u^0 = f$, it reads:

$$u^{k+1} = u^k + \tau \Delta u^k,$$

which exactly corresponds to the previous discretization of the Heat equation (1.2).

In the same vein, we can show that the Perona-Malik PDE[1] defined in (1.3) corresponds to a gradient descent algorithm applied to the convex functional $J_{PM}(u) = \sum_{x\in\Omega} \sqrt{||\nabla u(x)||^2 + 1}$, since:

$$\nabla J_{PM}(u) = -\mathrm{div}\left(\frac{\nabla u}{\sqrt{||\nabla u||^2 + 1}}\right).$$

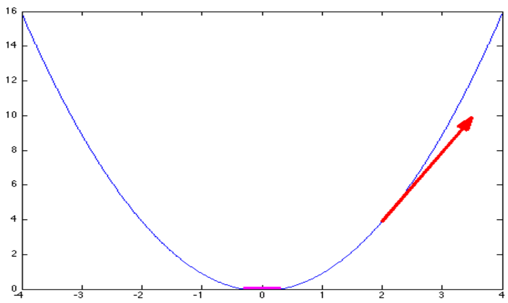
1 Check the convexity and the computation of the gradients of the above functions J_H and J_{PM}.

[1]where $g(t)$ of (1.4) is parameterized with $\alpha = 1$.

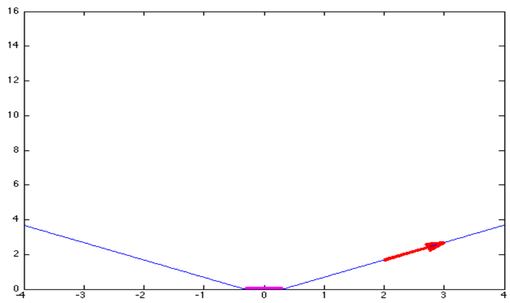
Interpretations From the gradient descent point of view, we see that with an infinite time t, the previous PDEs (1.1) and (1.3) will respectively converge to a global minimizer of the functions J_H and J_{PM}.

We denote as C the set of uniform images (i.e. constant) $u \in X$ (i.e. so that $u \in C$ iff $\nabla u(x) = 0, \forall x \in \Omega$). It is clear that $J_H(u) = 0$ if $u \in C$ and $J_H(u) > 0$ if $u \notin C$. The same observation can be made for J_{PM}. Hence C is the set of global minimizers of functions J_H and J_{PM}. This confirms that the PDEs will converge to constant images[2].

The difference between both PDEs concerns the paths u^k between $u^0 = f$ and $u^\infty = constant$. The isotropic Heat equation corresponds to a quadratic function J_H, whereas the anisotropic Perona-Malik model relies on a differentiable approximation of the piecewise-linear and non differentiable Total Variation regularization $\sum_{x\in\Omega} \sqrt{||\nabla u(x)||^2}$. These functions are displayed below:



$J_H(u)$



$J_{PM}(u)$

From now on, in the Figures, the x-axis is a $1D$ representation of $X = \mathbb{R}^{m\times n}$. The set C of global minimizers is in purple: $J_H(u) = J_{PM}(u)$ for all $u \in C$. The vectors $\nabla J_H(u)$ and $\nabla J_{PM}(u)$ are displayed in red for a same u.

From the shape of these functions, we understand that for an initialization f far from C and the same time steps, the gradient descent on the quadratic J_H will involve gradients $\nabla J_H(u)$ with higher norms than $\nabla J_{PM}(u)$ and it will reach faster the neighborhood of C than the gradient descent on J_{PM}. This explains why the Heat equation gives very smooth images (close to C) with few iterations.

Remark on the convergence of gradient descent If ∇J is Lipschitz continuous with constant L (i.e. $||\nabla J(u) - \nabla J(v)|| \leq L||u - v||$, for all $u, v \in X$), then the gradient descent converges for all $\tau < 2/L$. For the Heat equation, $\nabla J_H = -\Delta$ is Lispchitz continuous. With the discretization of the Laplacian considered page 5, the constant L of Δ is 8, so that we can take $\tau < 1/8$. On the other hand, it is worth noting that ∇J_{PM} is not Lispchitz continuous so that the Perona-Malik PDE may diverge if not optimizing the time step at each iteration with line search methods.

Enhancing PDEs methods A previously underlined drawback of the presented PDEs is that the original data f is only used as an initialization and forgotten along the process. We now see how defining a convex function that will take into account this information so that its minimizers will have a stronger link with f.

[2] With the Neumann condition, the PDEs will converge to the constant image where the constant is the mean value of f. It corresponds to a homogeneous diffusion on the whole domain Ω of the initial temperature f.

1.3 Variational model with data fidelity term

As previously noticed, the functions J_H and J_{PM} have a smoothing effect. A simple idea to counter balance this regularization behavior is to consider an additional data fidelity term:

$$J_D(u) = \frac{1}{2}\sum_{x\in\Omega}||u(x) - f(x)||^2 = \frac{1}{2}||u - f||_X^2$$

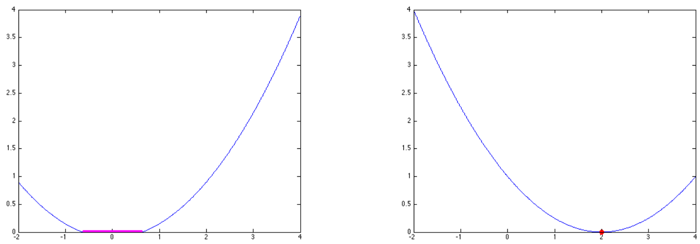
This function is strictly convex and its obvious minimizer is obtained for $u = f$. This model assume that the data is a degradation of the perfect unknown image u^* with a Gaussian noise ω, i.e. $f = u^* + \omega$. Notice that minimizing the $||.||_X^2$ norm corresponds, in the Bayesian framework, to the minimization of the likelihood of the image u with respect to the data noisy f.

We can now consider the following kind of models:

$$J_\lambda(u) = \lambda J_D(u) + J_R(u), \tag{1.5}$$

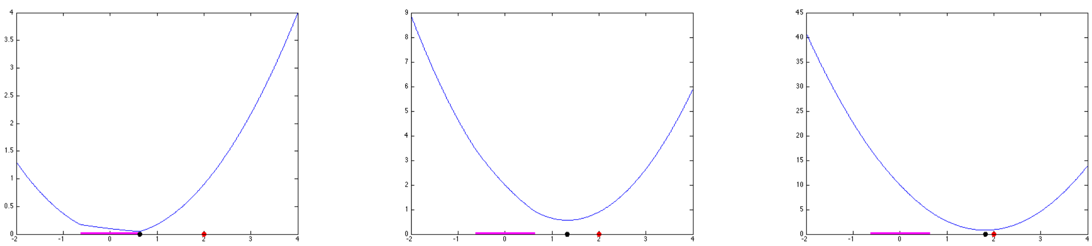
where $\lambda \geq 0$ is a parameter weighting the influence of the regularization with respect to the data and J_R is a regularization function that can be taken as $J_H(u) = \frac{1}{2}\sum_{x\in\Omega}||\nabla u(x)||^2$ or $J_{PM}(u) = \sum_{x\in\Omega}\sqrt{||\nabla u(x)||^2 + 1}$.

It is worth noting that the function J_λ in (1.5) is strictly convex for $\lambda > 0$. We call u_λ a minimizer of (1.5) for a given λ. For $\lambda = 0$, we recover the previously studied models and u_λ is not unique (any $u \in C$ is a minimizer). For $\lambda \to 0^+$, the minimum is achieved for the constant image with the same mean as f. For $\lambda \to \infty$, the data term is prominent so we have $\lim_{\lambda\to\infty} u_\lambda = f$. The interesting values are thus in between. We next illustrate the influence of this parameter on the minimizer u_λ.



$J_R(u)$ $\qquad$ $J_D(u)$

The set C of global minimizers of J_R is in purple. The minimizer of J_D $(= f)$ is in red.



$J_\lambda(u)$, $\lambda = 0.1$ $\qquad$ $J_\lambda(u)$, $\lambda = 2$ $\qquad$ $J_\lambda(u)$, $\lambda = 10$

Illustration of J_λ and its minimizer u_λ (black points) for different values of λ.

Hence, we clearly observe that the minimizer of J_λ will be a compromise between the data f and an uniform image in C. Tuning the parameter λ adequately is in practice more simple than setting a number of iteration in PDEs approaches. The parameter can for instance been set with respect to the expected noise level of the image.

Variational models in image processing In image processing applications (denoising, segmentation, optical flow estimation...), it is classical to design algorithms based on the minimization of a function. The important point is therefore to ensure that the minimizers of the proposed function have good properties with respect to the tackled application. Here these properties are the following: the denoised image u should be smooth and close to f.

1.3.1 Minimization of J_λ

We now consider the gradient descent algorithm to solve problem (1.5):

$$u^{k+1} = u^k - \tau(\lambda \nabla J_D(u^k) + \nabla J_R(u^k)).$$

We can first observe that $\nabla J_D(u) = (u - f) \in X$. Next, we detail the algorithm for different regularizers J_R.

2 Check the strict convexity and the computation of the gradient of the function J_D.

Tikhonov regularization The so-called Tikhonov regularization is $J_R(u) = J_H(u) = ||\nabla u||_X^2$. This function enforces the image to be ***smooth***. The corresponding the gradient descent algorithm reads:

$$u^{k+1} = u^k + \tau(\lambda(f - u^k) + \Delta u^k), \tag{1.6}$$

where the time step can be taken as $\tau = 1/(\lambda + 4)$.

Smoothed Total Variation regularization The so-called smoothed Total Variation regularization is $J_R(u) = J_{TV}^\epsilon(u) := \sum_{x\in\Omega} \sqrt{||\nabla u(x)||^2 + \epsilon}$. This function enforces the image to be ***piecewise constant***. The gradient descent algorithm is in this case:

$$u^{k+1} = u^k + \tau\left(\lambda(f - u^k) + \text{div}\left(\frac{\nabla u^k}{\sqrt{||\nabla u^k||^2 + \epsilon}}\right)\right), \tag{1.7}$$

where the time step τ must be taken small enough to avoid numerical instabilities (see Remark page 13). With the previous notations, we have $J_{PM} = J_{TV}^1$. Taking $\epsilon = 1$ gives a good approximation of the Total Variation regularization if the gray values of f are within the range $[0; 255]$.

PDE point of view It is interesting to interpret the algorithms (1.6) and (1.7) as PDEs. For the Tikhonov regularization, one recovers:

$$\left\{\begin{array}{lll} \frac{\partial u(t,x)}{\partial t} & = \lambda(f(x) - u(t,x)) + \Delta u(t,x) & \text{for } t \geq 0 \text{ and } x \in \Omega \\ u(0,x) & = f(x) & \text{for } x \in \Omega \\ \frac{\partial u(t,x)}{\partial N} & = 0 & \text{for } t > 0 \text{ and } x \in \partial\Omega. \end{array}\right. \tag{1.8}$$

With respect to the Heat equation and the Perona-Malink one, the additional term $\lambda(f(x) - u(t,x))$ now enforces, the solution $u(t,x)$ to go into the direction of $f(x)$, with an influence given by λ.

1.4 Back to work

We can now solve the problem (1.5) for the different regularizers.

3 Write a function *Denoise_ Tikhonov* that

- takes as argument the noisy image f, a time step τ (that can be automatically set to $1/(\lambda+4)$), a number of iterations K and a parameter λ
- realizes the gradient descent algorithm (1.6) for iterations $k = 1 \cdots K$
- returns u^K.

4 Write a function *Denoise_ TV* that

- takes as argument the noisy image f, a time step τ, a number of iterations K and parameters λ and ϵ
- realizes the gradient descent algorithm (1.7) for iterations $k = 1 \cdots K$
- returns u^K.

5 Write a script that test these functions on a noisy image f for different parameters λ.

What is the influence of λ ?

How does λ_{opt} evolves with respect to the amount of noise ?

Remark: The parameter K is a maximum number of iteration. The algorithm can be stopped if a convergence criteria is met. A standard criteria is to measure the normalized root-mean-square error (RMSE) between successive iterations: $||u^{k+1} - u^k||/||u^k||$ and stop the algorithm when it is small enough (for instance $< 10^{-5}$ for Tikhonov and $< 10^{-4}$ for smoothed Total Variation). Also notice that since the Tikhonov regularization is quadratic and the Total Variation one in piecewise linear, good values of λ are not in the same range for the 2 regularizations. An example of the obtained results is given below.

1.5 Solving Tikhonov regularization with Fourrier Transform

The Tikhonov regularization corresponds to solve the problem:

$$\min_u J_\lambda(u) = \lambda J_D(u) + J_H(u) = \frac{\lambda}{2}||u-f||_X^2 + \frac{1}{2}||\nabla u||_Y^2. \tag{1.9}$$

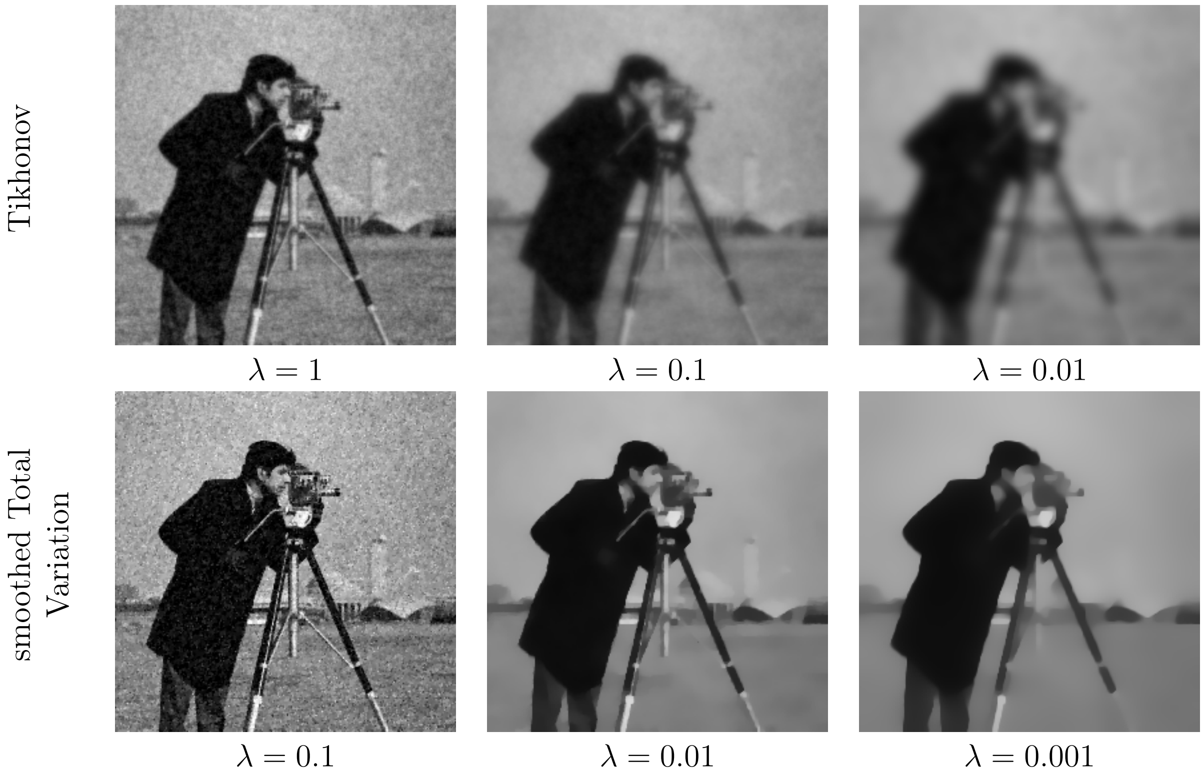
The minimizer u of this convex function is then characterized by $\nabla J_\lambda(u) = 0$. Computing the Euler-Lagrange equation of (1.9), it gives as optimality condition:

$$\lambda(u-f) - \Delta u = 0. \tag{1.10}$$

As for the Heat equation, whose solution $u(t,x)$ can be explicitly obtained through a convolution with an adequate kernel depending on t, the solution of the Tikhonov regularization problem can be explicitly computed (i.e. without minimizing J_λ iteratively).

The solution can indeed be exhibited by considering Discrete Fourier Transform (DFT) of the optimality condition (1.10).

We recall that the DFT of a $m \times n$ discrete image $f(k,l)$ ($0 \leq k \leq m-1$ et $0 \leq l \leq n-1$) is given, for $0 \leq p \leq m-1$ and $0 \leq q \leq n-1$, by:



Denoising results for different values of λ. First line: Minimizer u_λ of J_λ with Tikhonov regularization. Second line: Minimizer u_λ with smoothed Total Variation regularization.

$$\mathcal{F}(f)(p,q) = F(p,q) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} f(k,l)e^{-j(2\pi/m)pk}e^{-j(2\pi/n)ql} \tag{1.11}$$

and the inverse transform is:

$$f(k,l) = \frac{1}{mn}\sum_{p=0}^{m-1}\sum_{q=0}^{n-1} F(p,q)e^{j(2\pi/m)pk}e^{j(2\pi/n)ql} \tag{1.12}$$

One can show that, for the centered discretization of the Laplacian operator and periodic conditions, we have

$$\mathcal{F}(\Delta f)(p,q) = -4\mathcal{F}(f)(p,q)\left(\sin^2\left(\frac{\pi p}{m}\right) + \sin^2\left(\frac{\pi q}{n}\right)\right) \tag{1.13}$$

from which we can deduce that the minimizer u of (1.9) checks:

$$\mathcal{F}(u)(p,q) = \frac{\lambda\mathcal{F}(f)(p,q)}{\lambda + 4\left(\sin^2\left(\frac{\pi p}{m}\right) + \sin^2\left(\frac{\pi q}{n}\right)\right)} \tag{1.14}$$

7 Implement relation (1.14) and find the minimizer using functions *fft2* and *ifft2*. Compare the solution with the one obtained with the *Denoise_ Tikhonov* function for the same λ. Comments ?

7' *Bonus question:* Show relations (1.13) and (1.14). Recall: $2\sin^2(a) = 1 - \cos(2a)$.

Remark: The above DFT method assumes periodic conditions, while Neumann conditions were previously considered. As a consequence, differences between both approaches should be mainly visible on the image boundaries.

2 Extensions to deconvolution and inpainting

We now extend the previous variational model to a more general one dealing with other applications than denoising. To that end, instead of the data function J_D, we define a new data function J_A that will be minimized jointly with the two studied regularization functions. This data function reads:

$$J_A(u) = \frac{1}{2}\sum_{x\in\Omega} ||(Au)(x) - f(x)||^2 = \frac{1}{2}||Au - f||_X^2, \tag{2.15}$$

where A is a linear operator from X to X that can be represented as $mn \times mn$ matrix in the discrete setting. This data function can model different interesting applications:

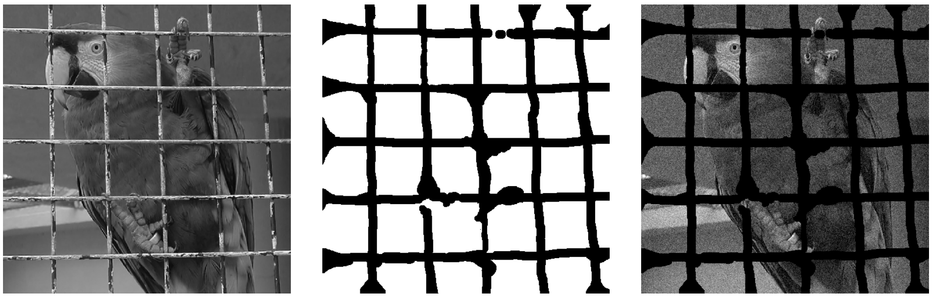
- **Deconvolution**: We assume that the available image f is obtained as $G * u^* + \omega$, where u^* is the unknown ground truth image to recover, $G(x)$ is a known $2D$ convolution filter and ω is an additional Gaussian noise. In this case, the data function $||G*u-f||_X^2$ tries to estimate an image u which convolution with the kernel G is close to the one of u^*. With adequate change of indexes, the discrete convolution of u with a kernel K can be seen as a matrix vector multiplication, by considering the $mn \times mn$ matrix $G_{kl} = G(x_l - x_k)$.

u^* $\quad$ $G * u^*$ $\quad$ $f = G * u^* + \omega$

Example of data f in case of deconvolution.

- **Inpainting**: We assume that the available image f is a partial observation of a perturbation of the ground truth image u^* to recover. More precisely, we only have access to observations of some pixels x that belong to a known region $\mathcal{M} \subset \Omega$. In this case, the data term can be taken as $\sum_{x\in\mathcal{M}} ||u(x) - f(x)||^2$. It corresponds to the framework of (2.15) by introducing the $mn \times mn$ matrix $M(k,l) = 1$ if $k = l$ and pixel $x_k \in \mathcal{M}$ and 0 otherwise.

Remark: Notice that J_A is strictly convex only if A is a positive definite matrix and thus an invertible matrix. In this case, one can equivalently consider as data term $||u - A^{-1}f||_X^2$ which enters in the framework of previous section. Hence, the interesting case to look at is when A has zero eigenvalues, for instance when A is positive semi-definite.



u^* $\mathcal{M}$ $f = u^*_{|\mathcal{M}}$

Example of data f in case of inpainting.

2.1 Gradient of J_A

The gradient of function (2.15) reads $\nabla J_A(u) = A^T(A(u-f)$. In the aforementioned applications one has:

- **Deconvolution**: With an isotropic kernel G, the gradient of $J_G(u) = \frac{1}{2}||G * u - f||^2_X$ is

$$\nabla J_G(u) = G * (G * u - f). \tag{2.16}$$

- **Inpainting**: The gradient of $J_{\mathcal{M}}(u) = \frac{1}{2}\sum_{x\in\mathcal{M}} ||u(x) - f(x)||^2$ is

$$\nabla J_{\mathcal{M}}(u)(x) = \begin{cases} u(x) - f(x) & if\, x \in \mathcal{M} \\ 0 & otherwise. \end{cases} \tag{2.17}$$

Considering the mask function $M(x) = 1$ if $x \in \mathcal{M}$ and 0 otherwise, we also have: $\nabla J_{\mathcal{M}}(u) = (u - f)M$.

8 Check the computation of $\nabla J_A(u)$.
Compute G^T and M^T.

2.2 Variational models and minimization

We can now use the new data functions together with the regularization ones, in order to formalize and solve the deconvolution and inpainting problems.

- **Deconvolution**: The function to minimize reads:

$$\frac{\lambda}{2}||G * u - f||^2_X + \sum_{x\in\Omega} \sqrt{||\nabla u(x)||^2 + \epsilon^2}.$$

Minimizing this function corresponds to find a piecewise constant image u whose convolution with G is close to f. As the data f is assumed to be blurred, considering a Tikhonov regularization is in this case not very smart, since it would not produce a sharp enough image. The gradient descent algorithm applied to this problem gives:

$$u^{k+1} = u^k + \tau\left(\lambda G * (f - G * u^k) + \operatorname{div}\left(\frac{\nabla u^k}{\sqrt{||\nabla u^k||^2 + \epsilon^2}}\right)\right), \tag{2.18}$$

- **Inpainting**: The function to minimize reads:

$$\frac{\lambda}{2}\sum_{x\in\mathcal{M}}||u(x)-f(x)||^2+\sum_{x\in\Omega}\sqrt{||\nabla u(x)||^2+\epsilon^2}.$$

The idea is to diffuse the known information into the region $\mathcal{M}$ to the masked regions $\Omega\backslash\mathcal{M}$ with the regularization function. We thus obtain:

$$u^{k+1}=u^k+\tau\left(\lambda(f-u^k)M+\operatorname{div}\left(\frac{\nabla u^k}{\sqrt{||\nabla u^k||^2+\epsilon^2}}\right)\right), \tag{2.19}$$

The Tikhonov regularization is of interest in this application, namely in almost uniform image regions (sky, water...), as it will realize an isotropic diffusion of the known information. The corresponding gradient descent algorithm is:

$$u^{k+1}=u^k+\tau\left(\lambda(f-u^k)M+\Delta u^k\right). \tag{2.20}$$

9 Write a function *Deconvolution_TV* that

- takes as argument an image f, a kernel G, a time step τ, a parameter ϵ, a number of iterations K and a parameter λ and returns u^K (see Page 7 for image convolution in Matlab)
- realizes the gradient descent algorithm (2.18) for iterations $k=1\cdots K$

10 Write a function *Inpainting_TV* that

- takes as argument an image f, a mask image M, a time step τ, a parameter ϵ, a number of iterations K and parameter λ and returns u^K
- realizes the gradient descent algorithm (2.19) f for iterations $k=1\cdots K$

11 Implement the function *Inpainting_Tichonov* that

- takes as argument an image f, a mask image M, a time step τ, a number of iterations K and parameter λ and returns u^K
- realizes the gradient descent algorithm (2.20) f for iterations $k=1\cdots K$

12 Write a script that test, for different parameters λ, these functions on images f obtained as follows

- Deconvolution: convolve the image of your choice with a Kernel G (for instance *G= fspecial('gaussian',[7 7],5);*) and add noise. Give the same G to your function *Deconvolution_TV*
- Inpainting: Take a large value for K and use the images and masks available here: Image 1, Mask 1, Image 2, Mask 2, Image 3, Mask 3.

13 Comments ? What are the properties of each models ? What are the properties of the parameters ?

First column: Deconvolution with smoothed Total Variation. Second column: Inpainting with smoothed total variation. Third column: Inpainting with Tikhonov. Last column: Inpainting

14 Write functions *Denoise_g1* and *Denoise_g2* in the same way as function *Denoise_TV*, but where the function ϕ of the lecture is replaced by $\phi_1(\xi) = \frac{\xi^2}{1+\xi^2}$. or $\phi_2(\xi) = \log(1+\xi^2)$.

Notice that in the rest of the practical session we have used $\phi(\xi) = \sqrt{|\xi|^2+1}$.

Write the optimality condition in each case.

Compare the results of *Denoise_g1*, *Denoise_g2*, and *Denoise_TV*. Comments ?