PhD Winter School 2023:

ADVANCED METHODS for MATHEMATICAL IMAGE ANALYSIS

Computational Imaging Lab (A)

Alessandro Lanza,
Monica Pragliola

mail: alessandro.lanza2@unibo.it
monica.pragliola@unina.it

	WEDNESDAY 18/01	THURSDAY 19/01	FRIDAY 20/01	SATURDAY SUNDAY	MONDAY 23/01	TUESDAY 24/01	WEDNESDAY 25/01		MONDAY 06/02
08:00									
09:00									VIRTUAL PART
10:00									SEMINAR
11:00									EXAMS
12:00									
13:00	Lunch	Lunch	Lunch		Lunch	Lunch	Lunch		
14:30	Comp. Imaging Lab	EXERCISES	Comp.Imaging LAB		SEMINAR **Cultural Heritage**	LAB	SEMINAR **Energy Physics**		
15:30	LAB	LAB	EXERCISES		EXERCISES	Comp.Imaging LAB	Comp.Imaging LAB		
16:30	SEMINAR **Automotive**	SEMINAR **Industrial**	SEMINAR **Health**		EXERCISES	LAB	scientific coffee		
17:30									
	Prof. L. Calatroni	Social Dinner			Prof. J. Gondzio				Scientific committee UMI-MIVA
	Prof. O. Öktem				Prof. S. Siltanen				

A

B

PROVIDED MATERIAL:

- Slides (LEC_A1, LEC_A2, LEC_B)
- Matlab source codes
- Repository of relevant related articles

OUTLINE:

- Inverse Imaging Problems:
  - Examples and definitions; Image Restoration and Inpainting
- Variational Methods for Inverse Imaging Problems:
  - Some popular variational models; the large and important class of TV_p-L_q variational models for Image Restoration and Inpainting:
    - Specific models of interest (motivations and properties):
      - TIK-L_2 , TV-L_2 , TIK-L_1 , TV-L_1 (unconstrained) models
      - Numerical solution of the models:
        - TIK-L_2 model direct solution by Discrete Fourier Transform (DFT)
        - TV-L_2, TIK-L_1, TV-L_1 models iterative solution by ADMM
    - Automatic selection of the regularization parameter (discrepancy):
      - The constrained TV-L_2 model, numerically solved by ADMM
    - Experimentation (Matlab)

GOALS:

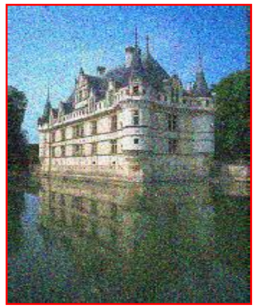
- Give students an idea on variational methods for the solution of inverse problems in imaging, with focus on the popular (and effective), non-differentiable TV regularizer

- Give students a preliminary idea of the popular Alternating Direction Method of Multipliers (ADMM) iterative optimization approach applied to the numerical solution of non-smooth variational models

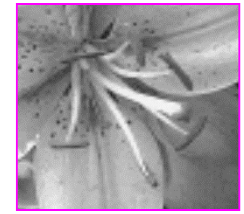
- Share with students a source code (Matlab) which makes it possible to rigorously compare (in a qualitative/quantitative way) the performance of different variational models for the solution of image restoration and inpainting inverse problems for different test images corrupted by different degradation factors

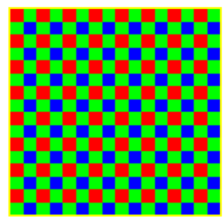
Some Inverse Problems in Imaging

- De-Noising
- De-Blurring
- In-Painting
- De-Mosaicing
- Computed Tomography
- Super-Resolution

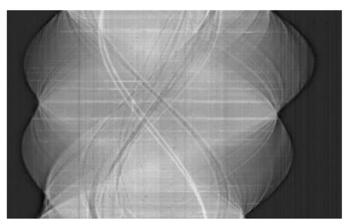













Some Inverse Problems in Imaging

- De-Noising
- De-Blurring
- In-Painting
- De-Mosaicing
- Computed Tomography
- Super-Resolution
- Segmentation
- and more …

Image Restoration Inverse Problem

Goal: **Restore *degraded (blurred and noisy)* images**

Rocca di Bertinoro
(Forlì – Cesena)

forward (degradation)

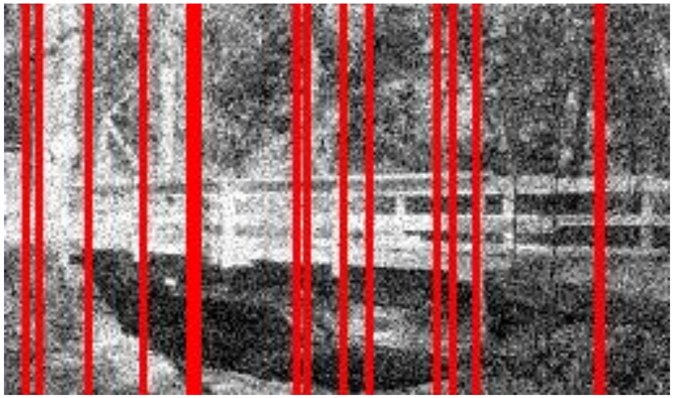
backward (estimation)

Image Inpainting Inverse Problem

Goal: Restore *degraded («masked» and noisy)* images

forward (degradation)

backward (estimation)



Inverse Imaging Problems: Definitions

General Image Formation Model

$$y = \mathcal{N}(\Phi(x)), \text{ where :}$$

- $\Phi(x) = \varphi(Ax)$ typically deterministic degradation operator, with:

  $\varphi(\cdot)$ identity or nonlinear operator, $A\,\cdot$ linear operator
- $\mathcal{N}(\cdot)$ typically randomic noise operator

Inverse Imaging Problems: Definitions

General Image Formation Model

$$y = \mathcal{N}(\Phi(x)), \text{ where :}$$

- $\Phi(x) = \varphi(Ax)$ typically deterministic degradation operator, with:

  $\varphi(\cdot)$ identity or nonlinear operator, $A\,\cdot$ linear operator
- $\mathcal{N}(\cdot)$ typically randomic noise operator

The inverse problem is called linear/nonlinear depending on φ(.)

Inverse Imaging Problems: Definitions

General Image Formation Model

$$y = \mathcal{N}(\Phi(x)), \text{ where :}$$

- $\Phi(x) = \varphi(Ax)$ typically deterministic degradation operator, with:

  $\varphi(\cdot)$ identity or nonlinear operator, $A\cdot$ linear operator

- $\mathcal{N}(\cdot)$ typically randomic noise operator

Example: **Image Denoising**

$\varphi(\cdot)$ and $A\cdot$ identity operators

$$\boxed{y = \mathcal{N}(x)}$$

LINEAR inverse problem

Inverse Imaging Problems: Definitions

General Image Formation Model

$$y = \mathcal{N}(\Phi(x)), \text{ where :}$$

- $\Phi(x) = \varphi(Ax)$ typically deterministic degradation operator, with:

  $\varphi(\cdot)$ identity or nonlinear operator, $A\,\cdot$ linear operator

- $\mathcal{N}(\cdot)$ typically randomic noise operator

Example: **Image Restoration (deblurring + denoising)**

$$\varphi(\cdot) \text{ identity operator, } A\,\cdot = K\,\cdot \text{ blurring operator}$$

$$\boxed{y = \mathcal{N}(Kx)}$$ LINEAR inverse problem

Inverse Imaging Problems: Definitions

General Image Formation Model

$$y = \mathcal{N}(\Phi(x)), \text{ where :}$$

- $\Phi(x) = \varphi(Ax)$ typically deterministic degradation operator, with:

  $\varphi(\cdot)$ identity or nonlinear operator, $A\,\cdot$ linear operator

- $\mathcal{N}(\cdot)$ typically randomic noise operator

Example: **Image Inpainting**

$\varphi(\cdot)$ identity operator, $A \cdot = S \cdot$ selection (masking) operator

$$\boxed{y = \mathcal{N}(Sx)}$$

<u>LINEAR</u> inverse problem

The IMAGE RESTORATION and INPAINTING

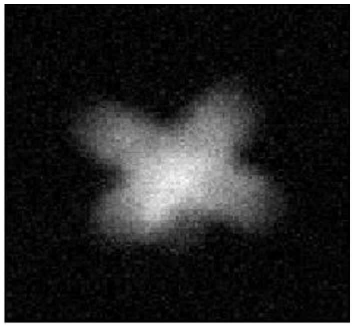
INVERSE PROBLEMS

Image Restoration

b = u $*$ k + n

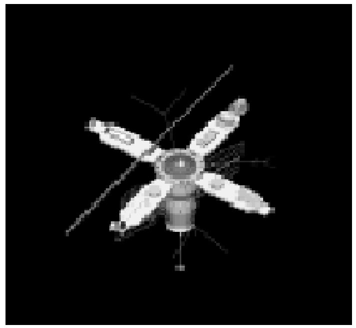
2D discrete convolution

additive noise



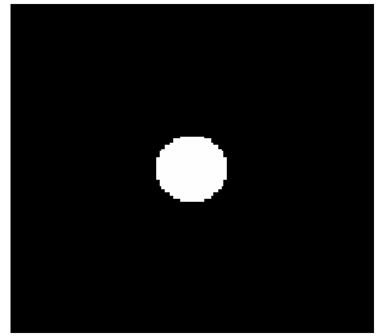
Observed image (known)

=



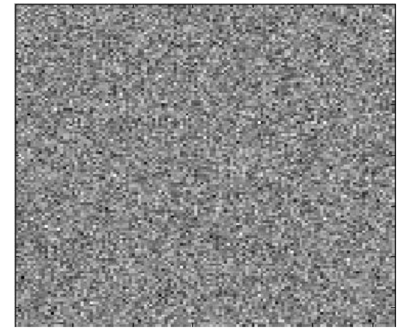
Original image (unknown)

$*$



Point Spread Function (known/unknown)

+



Realization of Random Noise (unknown)

Goal: Given b and k (non-blind), and some information on the noise distribution, recover u → **Inverse Problem**

Note: blur is not necessarily space-invariant (convolution operator) and noise is not necessarily additive (e.g. multiplicative, Poisson,...)

Image Restoration

Continuous degradation model:

compact (rectangular) domain

$$b(x) = (k * u)(x) + n(x) = \iint k(x-y)u(y)dy + n(x), \quad x, y \in \Omega \subset \mathbb{R}^2$$

Observed degraded image

Original image

Noise (additive) image

Blur kernel (PSF): linear space-invariant blur

Discrete degradation model (w x h image → d = w h pixels):

$$b = K u + n \qquad u, b, n \in \mathbb{R}^d, \; K \in \mathbb{R}^{d \times d}$$

Blur matrix

images are vectorized (d-entries column vectors)

Image Restoration

Continuous model:

$$\boxed{b(x) = (k * u)(x) + n(x)} \qquad x \in \Omega \subset \mathbb{R}^2$$

Discrete model:

$$\boxed{b = K u + n} \qquad u, b, n \in \mathbb{R}^d, \;\; K \in \mathbb{R}^{d \times d}$$

with blur matrix K: block Toeplitz with Toeplitz blocks,
block circulant with circulant blocks, ...
huge dimension, sparse

BCs

severely ill-conditioned $\longrightarrow$ regularization

Image Restoration

Continuous model:

$$\boxed{b(x) = (k * u)(x) + n(x)} \qquad x \in \Omega \subset \mathbb{R}^2$$

Discrete model:

$$\boxed{b = K\,u + n} \qquad u, b, n \in \mathbb{R}^d, \;\; K \in \mathbb{R}^{d \times d}$$

BLUR DEGRADATION: DETERMINISTIC NATURE

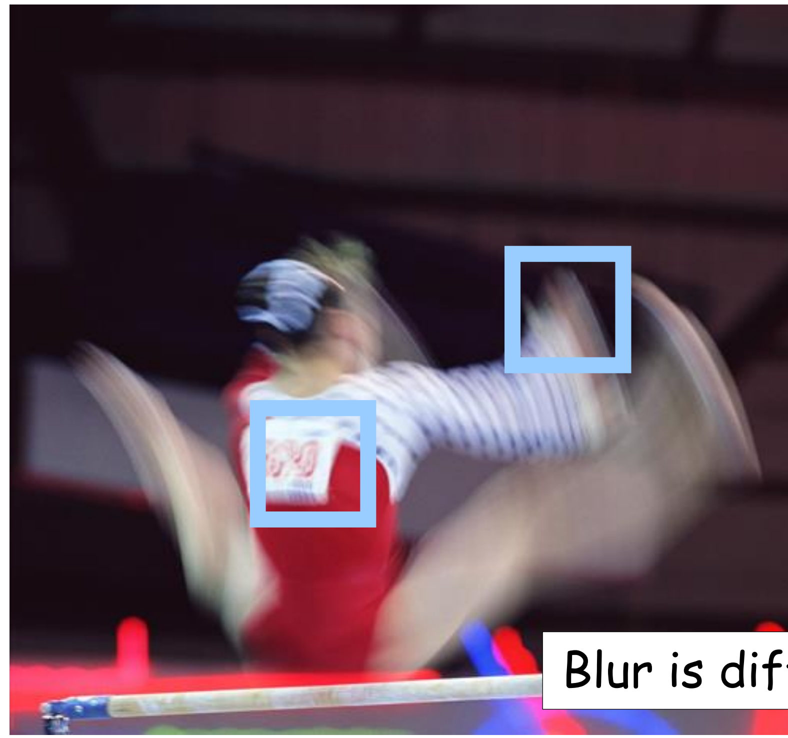
NOISE DEGRADATION: PROBABILISTIC NATURE

In practice, blur and noise models can be known a priori or inferred before restoration ...

Blur models: space-invariant/variant

Two causes for *motion* blur:

Camera motion



Scene motion

Blur can be the same

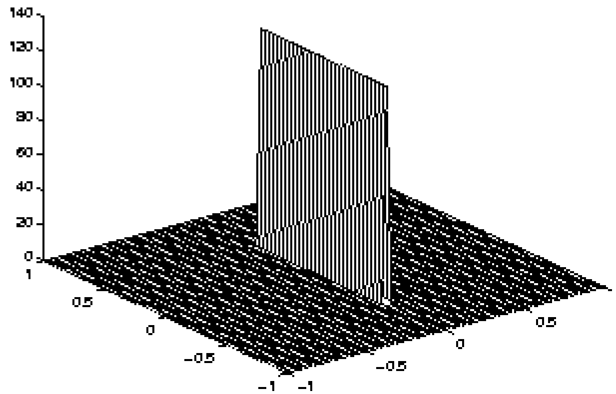
depending on 3d scene geometry

Blur is different

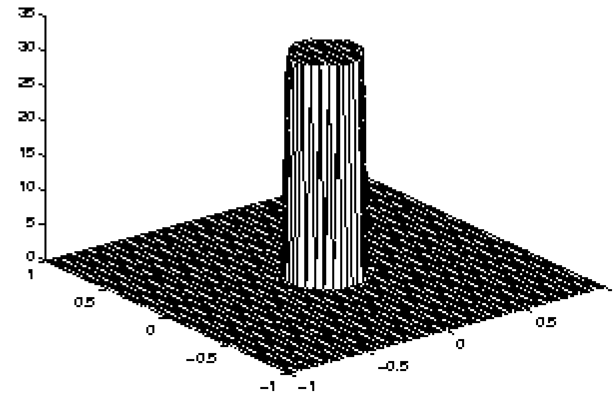
We will consider <u>space-invariant</u> blurs
(blur as a 2D Convolution)

Blur models: Point-Spread Functions

PSFs with sharp edges:

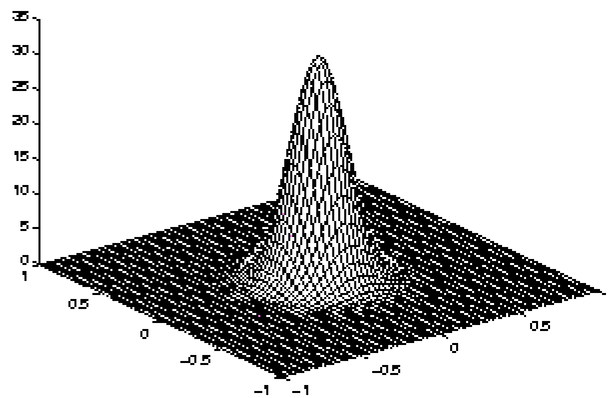


Motion Blur

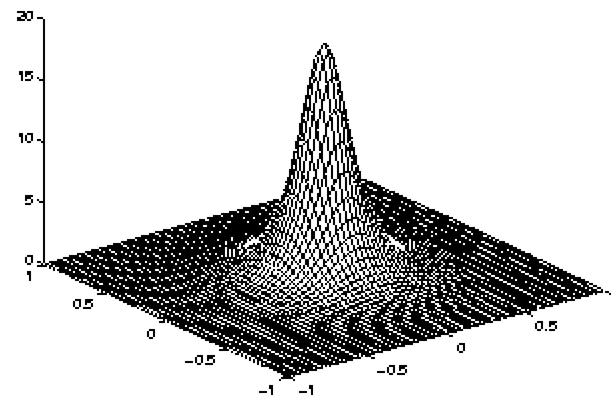


Out of Focus Blur

PSFs with smooth transitions:



Gaussian Blur



Scatter Blur

Boundary conditions

Zero-Dirichlet

Periodic

Reflective (Neumann hom.)

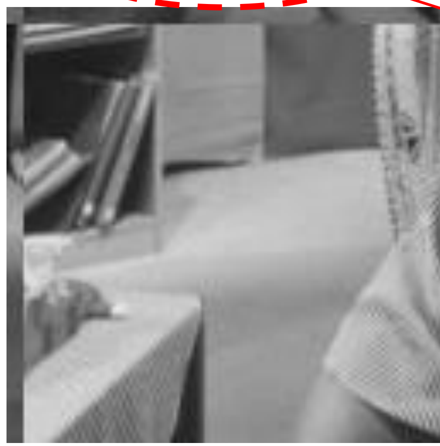
Anti-reflective
(S.Serra-Capizzano,
M. Donatelli, et al.)

Synthetic
(Y. Wai, J. Nagy)

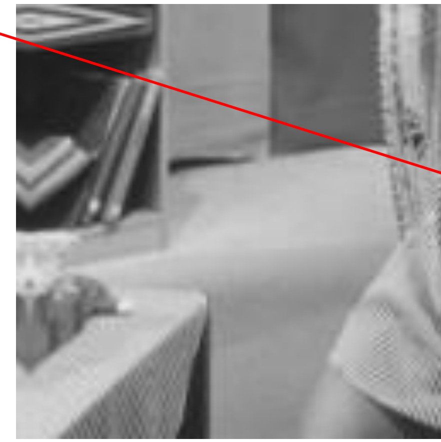
Boundary conditions

Zero-Dirichlet

Periodic



Reflective (Neumann hom.)



2D DFT
(by FFT)

Anti-reflective
(S.Serra-Capizzano,
M. Donatelli, et al.)

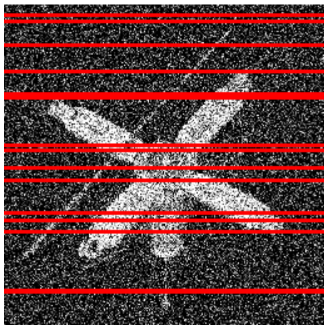
Synthetic
(Y. Wai, J. Nagy)

Image Inpainting

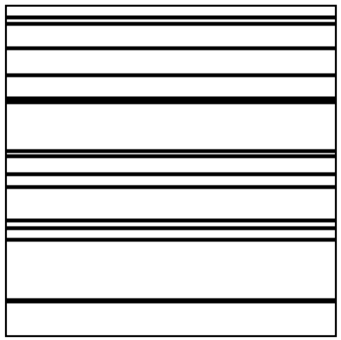
$$b = M \odot \left\{ u + \tilde{n} \right\}$$

Hadamard matrix product

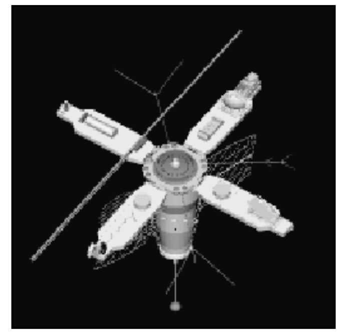
additive noise



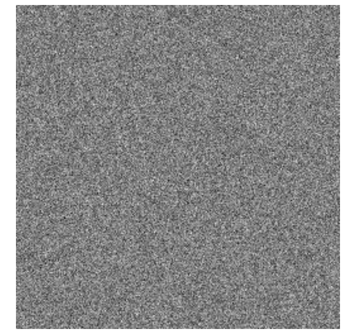
Observed image (known)



Binary Inpainting mask (known)



Original image (unknown)



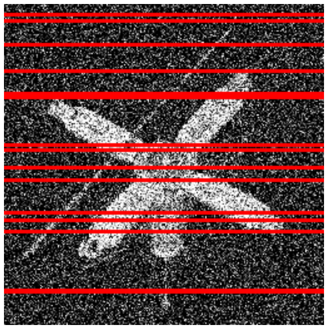
Realization of Random Noise (unknown)

Goal: Given b and M, and some information on the noise distribution, recover u → Inverse Problem

Note: noise is not necessarily additive (e.g. multiplicative, Poisson,...)

Image Inpainting

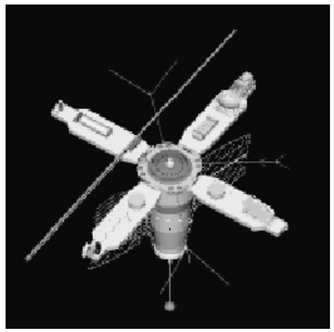
b



Observed image (known)

$=$

u

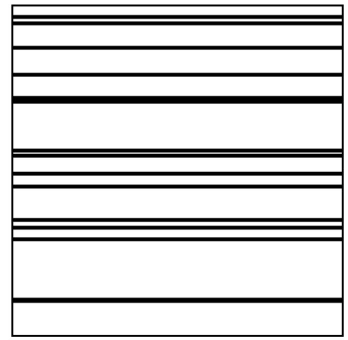


Original image (unknown)

Hadamard matrix product

$\odot$

M

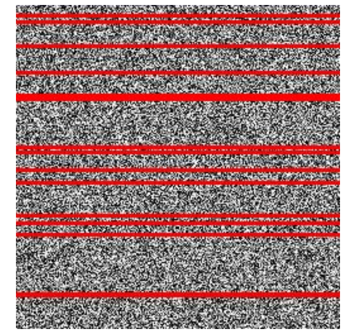


Binary Inpainting mask (known)

additive noise

$+$

$n = M \odot \tilde{n}$



Realization of Random Noise (unknown)

Goal: Given b and M, and some information on the noise distribution, recover u → **Inverse Problem**

Note: noise is not necessarily additive (e.g. multiplicative, Poisson,...)

Image Inpainting

Continuous degradation model:

compact (rectangular) domain

$$b(x) = m(x)\big(u(x) + \tilde{n}(x)\big) = m(x)u(x) + n(x), \qquad x \in \Omega \subset \mathbb{R}^2$$

Observed degraded image

Original image

Noise (additive) image

Binary Inpainting mask function (characteristic function)

Discrete degradation model (w x h image → d = w h pixels):

$$b = S(u + \tilde{n}) = S\,u + n \qquad u, b, n, \tilde{n} \in \mathbb{R}^d,\ S \in \mathbb{R}^{d \times d}$$

images are vectorized (d-entries column vectors)

Binary Inpainting mask (selection) matrix

Image Inpainting

Continuous degradation model:

compact (rectangular) domain

$$b(x) = m(x)\big(u(x) + \tilde{n}(x)\big) = m(x)u(x) + n(x), \qquad x \in \Omega \subset \mathbb{R}^2$$

Observed degraded image

Original image

Noise (additive) image

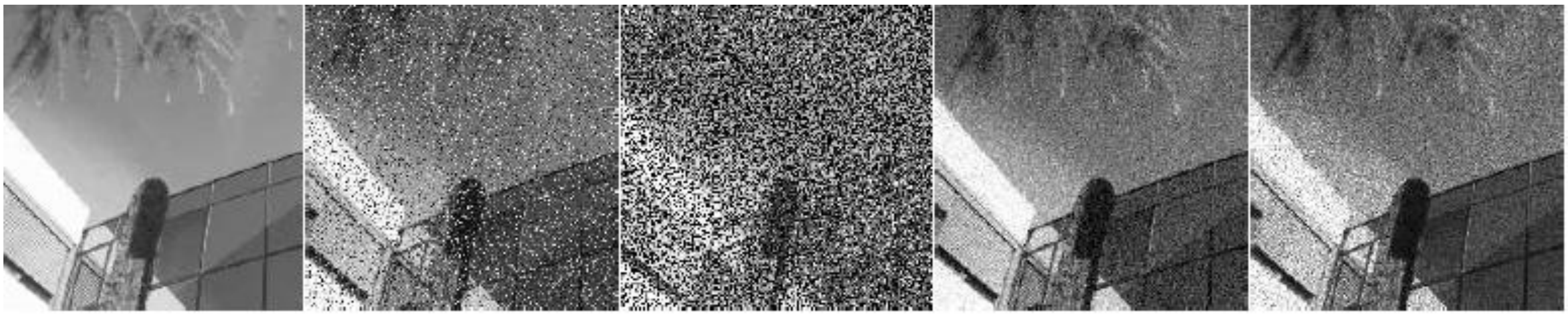
Binary Inpainting mask function (characteristic function)

Discrete degradation model (w x h image → d = w h pixels):

$$b = S(u + \tilde{n}) = S\,u + n \qquad u, b, n, \tilde{n} \in \mathbb{R}^d,\ S \in \mathbb{R}^{d\times d}$$

$S = diag(s_1, \ldots, s_d)$ with $s_i = 0$ if the i-th pixel belongs to the "inpainting set", 1 otherwise

Noise models



Original salt and pepper speckle Gaussian add. Gaussian mult.

Some important noise models

$Au \in \mathbb{R}^s$ vectorized image, $(Au)_i \in \mathbb{R}$ i-th entry

Additive noises: $b_i = \left(Au\right)_i + n_i \,, \quad i \in \Omega \triangleq \left\{1, 2, \ldots, s\right\}$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0, \sigma)$

$$p(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(x-\mu\right)^2}{2\sigma^2} \right)$$

$\mu \in \mathbb{R}$ mean

$\sigma \in \mathbb{R}_{++}$ standard deviation

σ is also called **noise level**

White noise: the covariance matrix for n is a scaled identity, Cov(n) = $\sigma^2 I$.
This is true if all the elements n_i of n are uncorrelated and from the same distribution.

Some important noise models

Additive noises: $$b_i = (Au)_i + n_i\,, \quad i \in \Omega \triangleq \{1,2,\ldots,s\}$$

Additive White Gaussian Noise (AWGN): $$N_i \sim G(0,\sigma)$$

Additive White Uniform Noise (AWUN) $$N_i \sim U(0,\sigma)$$

$$p(x\,|\,\mu,\sigma) = \begin{cases} \dfrac{1}{2\sqrt{3}\sigma} & \text{for } x \in \left[\mu-\sqrt{3}\sigma, \mu+\sqrt{3}\sigma\right] \\ 0 & \text{for } x \notin \left[\mu-\sqrt{3}\sigma, \mu+\sqrt{3}\sigma\right] \end{cases}$$

$\mu \in \mathbb{R}$ mean

$\sigma \in \mathbb{R}_{++}$ standard deviation

Some important noise models

Additive noises: $$b_i = (Au)_i + n_i\,, \quad i \in \Omega \triangleq \{1,2,\ldots,s\}$$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0,\sigma)$

Additive White Uniform Noise (AWUN) $N_i \sim U(0,\sigma)$

Additive White Laplacian Noise (AWLN) $N_i \sim L(0,\sigma)$

$$p(x \mid \mu,\sigma) = \frac{1}{\sigma\sqrt{2}} \exp\left(-\sqrt{2}\,\frac{|x-\mu|}{\sigma} \right)$$

$\mu \in \mathbb{R}$ mean

$\sigma \in \mathbb{R}_{++}$ standard deviation

Some important noise models

Additive noises: $b_i = (Au)_i + n_i\,, \quad i \in \Omega \triangleq \{1, 2, \ldots, s\}$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0, \sigma)$

Additive White Uniform Noise (AWUN) $N_i \sim U(0, \sigma)$

Additive White Laplacian Noise (AWLN) $N_i \sim L(0, \sigma)$

Additive White GG Noise (AWGGN) $N_i \sim GG(0, \sigma, \beta)$

Generalized Gaussian

Some important noise models

Additive noises: $b_i = (Au)_i + n_i\,, \quad i \in \Omega \triangleq \{1, 2, \ldots, s\}$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0, \sigma)$

Additive White Uniform Noise (AWUN) $N_i \sim U(0, \sigma)$

Additive White Laplacian Noise (AWLN) $N_i \sim L(0, \sigma)$ or α

Additive White GG Noise (AWGGN) $N_i \sim GG(0, \sigma, \beta)$

$$p(x \mid \mu, \sigma, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\frac{|x-\mu|^{\beta}}{\alpha^{\beta}} \right), \; \alpha = \sigma \sqrt{\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}}$$

$\mu \in \mathbb{R}$ mean

$\alpha \in \mathbb{R}_{++}$ scale parameter

$\beta \in \mathbb{R}_{++}$ shape parameter

Some important noise models

Additive noises: $$b_i = \left(Au\right)_i + n_i\,,\quad i \in \Omega \triangleq \{1,2,\ldots,s\}$$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0,\sigma)$

Additive White Uniform Noise (AWUN) $N_i \sim U(0,\sigma)$

Additive White Laplacian Noise (AWLN) $N_i \sim L(0,\sigma)$

$\boxed{\text{Additive White GG Noise (AWGGN)} \qquad N_i \sim GG(0,\sigma,\beta)}$

$$p(x\,|\,\mu,\sigma,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)}\exp\left(-\frac{|x-\mu|^{\beta}}{\alpha^{\beta}}\right),\ \alpha = \sigma\sqrt{\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}}$$

Γ Gamma function:

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}e^{-t}\,dt,\quad x \in \mathbb{R}_{++}$$

Some important noise models

Additive noises: $b_i = \left(Au\right)_i + n_i\,,\quad i \in \Omega \triangleq \{1, 2, \ldots, s\}$

Additive White Gaussian Noise (AWGN): $N_i \sim G(0, \sigma)$

Additive White Uniform Noise (AWUN) $N_i \sim U(0, \sigma)$

Additive White Laplacian Noise (AWLN) $N_i \sim L(0, \sigma)$

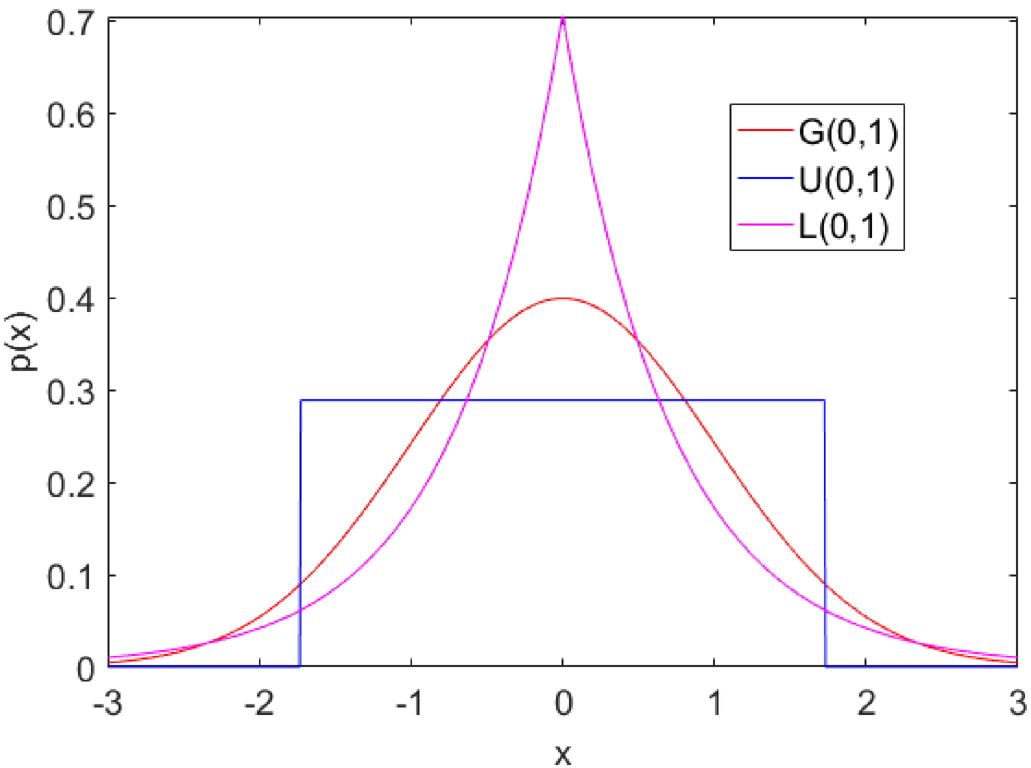
Additive White GG Noise (AWGGN) $N_i \sim GG(0, \sigma, \beta)$

$$p(x \mid \mu, \sigma, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\frac{|x-\mu|^{\beta}}{\alpha^{\beta}} \right), \; \alpha = \sigma\sqrt{\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}}$$

$$\beta = 2: \quad \mathrm{G}(\mu, \sigma)$$
$$\beta \to +\infty: \quad \mathrm{U}(\mu, \sigma)$$
$$\beta = 1: \quad \mathrm{L}(\mu, \sigma)$$

Some important noise models

Additive noises: $b_i = (Au)_i + n_i\,,\quad i \in \Omega \triangleq \{1,2,\dots,s\}$



$$N_i \sim G(0,\sigma) = GG(0,\sigma,2)$$

$$N_i \sim U(0,\sigma) = GG(0,\sigma,+\infty)$$

$$N_i \sim L(0,\sigma) = GG(0,\sigma,1)$$

Some important noise models

Multiplicative noises: $b_i = (Au)_i \times n_i\,, \quad i \in \Omega \triangleq \{1,2,\ldots,s\}$

Multiplicative White Gaussian Noise (MWGN): $N_i \sim G(1,\sigma)$

Multiplicative White Uniform Noise (MWUN) $N_i \sim U(1,\sigma)$

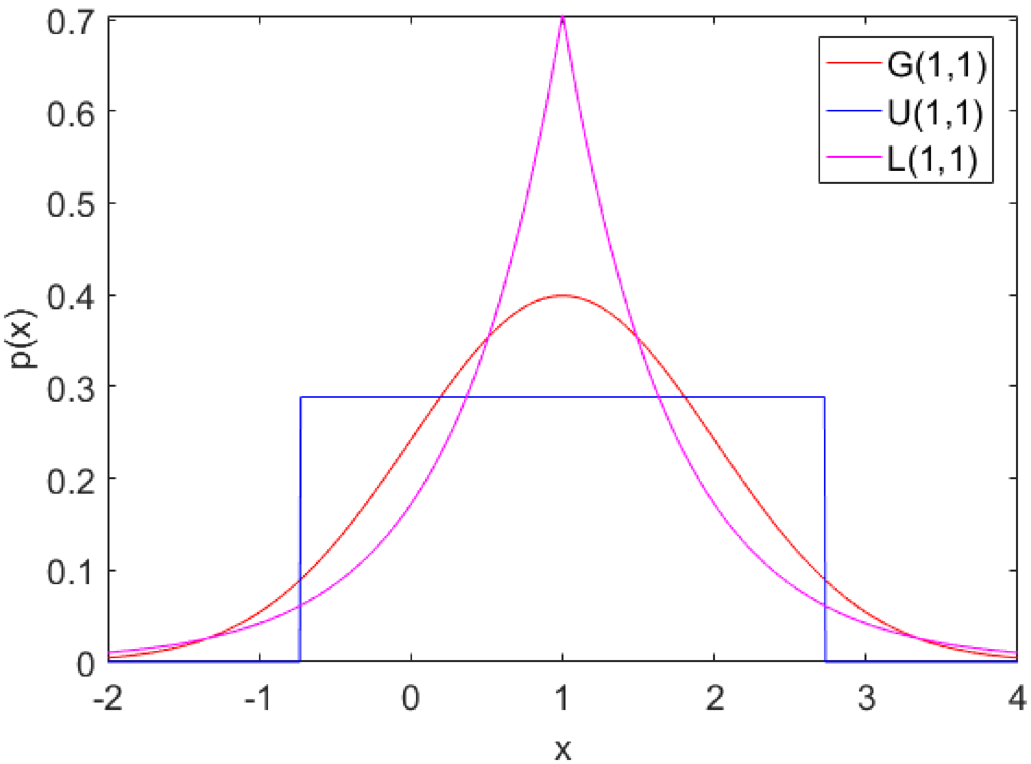
Multiplicative White Laplacian Noise (MWLN) $N_i \sim L(1,\sigma)$

or α

Multiplicative White GG Noise (MWGGN) $N_i \sim GG(1,\sigma,\beta)$

Some important noise models

Multiplicative noises: $b_i = (Au)_i \times n_i, \quad i \in \Omega \triangleq \{1, 2, \ldots, s\}$



$$N_i \sim G(1,\sigma) = GG(1,\sigma,2)$$

$$N_i \sim U(1,\sigma) = GG(1,\sigma,+\infty)$$

$$N_i \sim L(1,\sigma) = GG(1,\sigma,1)$$

Some important noise models

Impulsive noises:

$$b_i = \begin{cases} n_i & \text{if } \; i \in \Omega_0 \subseteq \Omega \\ (Au)_i & \text{if } \; i \in \Omega \setminus \Omega_0 \end{cases}$$

Impulsive Salt and Pepper Noise (ISPN):

$$n_i \in \{0, 255\}, \qquad \mathrm{P}(n_i = 0) = \mathrm{P}(n_i = 255) = 0.5$$

Impulsive Random-Valued (IRVN):

$N_i \sim U(127.5, 127.5/\sqrt{3})$ uniformly distributed in [0,255]

VARIATIONAL METHODS

for INVERSE PROBLEMS

Variational models for inverse problems

Solution image by minimization of an energy function(al) (inverse problem casted as an optimization problem):

$$\boxed{u^* \in \underset{u \in R^d \text{ or } u \in V}{\arg\min} \left\{ J(u;\mu) = R(u) + \mu F(u;b,A) \right\}}$$

vector space: discrete setting

function space: continuous setting

Regularization term: a priori information on the original image (regularity, sparsity, ...)

Fidelity term: a priori information on the data acquisition model, in particular noise (additive or multiplicative, pdf, spectrum, ...)

Regularization parameter: positive scalar, which allows to set the (desired) tradeoff between regularization and fidelity to the abserved data

Variational models for inverse problems

Solution image by minimization of an energy function(al) (inverse problem casted as an optimization problem):

$$u^* \in \underset{u \in R^d \text{ or } u \in V}{\arg\min} \left\{ J(u;\mu) = \mu R(u) + F(u;b,K) \right\}$$

vector space: discrete setting

function space: continuous setting

Regularization term: a priori information on the original image (regularity, sparsity, ...)

Fidelity term: a priori information on the data acquisition model, in particular noise (additive or multiplicative, pdf, spectrum, ...)

Regularization parameter: positive scalar, which allows to set the (desired) tradeoff between regularization and fidelity to the abserved data

Popular Variational Models

Popular regularization functions/terms:

$$R(u) = \sum_{i=1}^{d} \Phi\left(g_i(u) := \left\|(\nabla u)_i\right\|_2\right)$$

$\Phi(s_i) = g_i^2$ **Tikhonov**

$\Phi(s_i) = g_i$ **TV**

$\Phi(s_i) = g_i^p$ **TV_p**

$\Phi(s_i) = \rho^2 \ln(1 + g_i^2 / \rho^2)$ **Perona - Malik**

Popular fidelity functions/terms:

$$F(u; b, A) = \frac{1}{q}\left\|A u - b\right\|_q^q$$

q=1: Additive White Laplacian noise (AWLN)

q=2: Additive White Gaussian noise (AWGN)

Popular TV-L_2 (**ROF**) variational model (nonsmooth, convex) for AWGN:

$$u^* = \arg\min_{u \in R^d} \left\{ J(u) = \mu \underbrace{\sum_{i=1}^{d} \left\|(\nabla u)_i\right\|_2}_{TV(u)} + \frac{1}{2}\left\|K u - b\right\|_2^2 \right\}$$

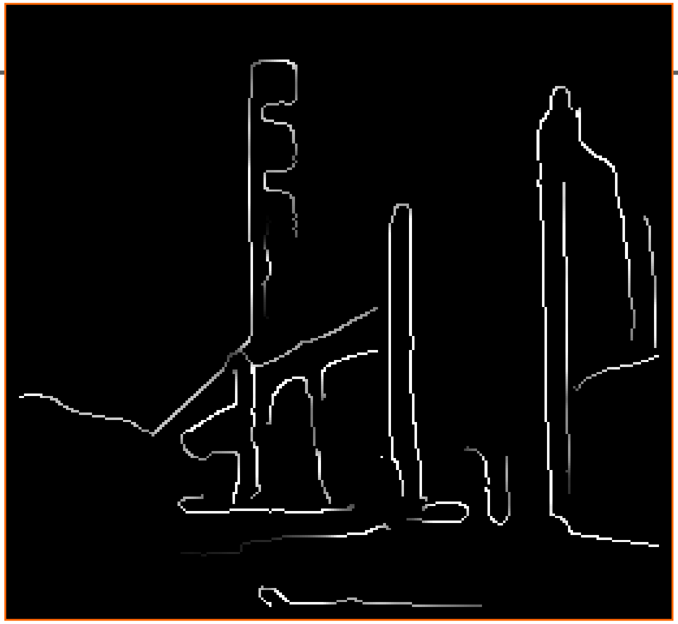
semi-norm

Edge-enhanched regularization

Tikhonov regularization

TV regularization

Edge-enhanched regularization



Tikhonov regularization

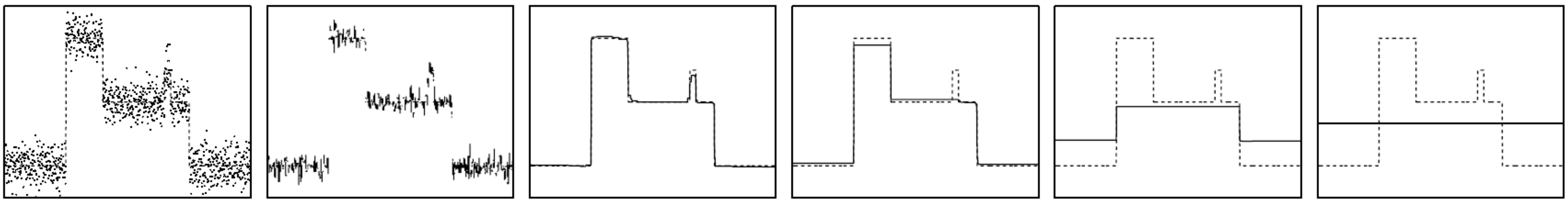
TV regularization

TV-L_2 (ROF) model: regularization parameter

$$u^* = \underset{u \in \mathbb{R}^d}{\arg\min} \left\{ J(u;\mu) = \mu \overbrace{\sum_{i=1}^{d} \left\| (\nabla u)_i \right\|_2}^{TV(u)} + \frac{1}{2} \left\| K u - b \right\|_2^2 \right\}$$

Experiment for the case of only denoising (no blur, K is the identity matrix) for 1D signals:

Noisy (b, μ=0)	μ = 0.0001	μ = 0.001	μ = 0.01	μ = 0.1	μ = 1

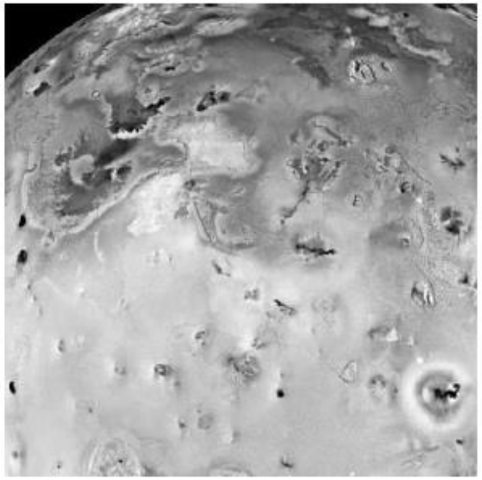


restoration residual $r := K\,u - b$ (or $r := b - K\,u$)

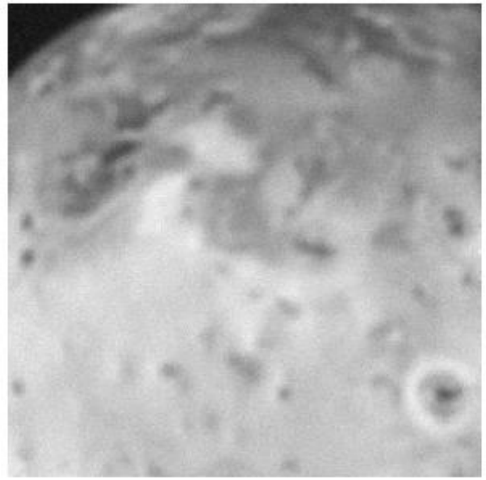
regularization parameter increases → restoration residual norm increases

TV-L_2 (ROF)

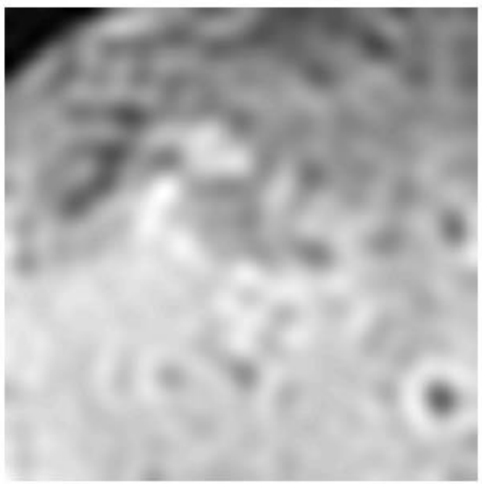
Exact



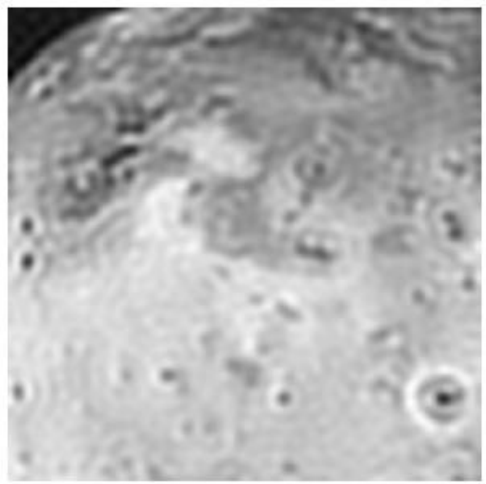
Blurred



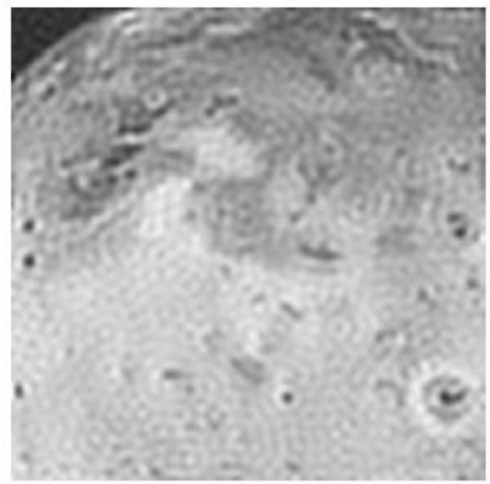
$\boldsymbol{\mu}$ too large



$\boldsymbol{\mu} \approx$ ok



$\boldsymbol{\mu}$ too small



WE ARE INTERESTED

in TWO SPECIFIC LINEAR INVERSE PROBLEMS:

IMAGE RESTORATION and INPAINTING,

and in a SPECIFIC CLASS of NOISES:

ADDITIVE, I.I.D, GG-DISTRIBUTED,

with particular focus on Gaussian and Laplacian noises

. . .

TV_p- L_q VARIATIONAL MODELS for IMAGE REST. / INP.

$$u^* \in \arg\min_{u \in \mathbb{R}^d} J(u;\mu), \quad \text{with:}$$

$$J(u;\mu) = \frac{1}{p}\sum_{i=1}^{d} \left\| (\nabla u)_i \right\|_2^p + \frac{\mu}{q}\left\| Au - b \right\|_q^q , \quad \mu, p, q > 0$$

We are interested in the **image rest./inp.** inverse problems (**A** square blur or selection matrix) and (mainly) interested in cases **q = 2, q = 1** for the fidelity term, corresponding to additive i.i.d. **Gaussian** and **Laplacian** noises, respectively, and in cases **p = 2**, **p = 1** for the regularization term, yielding the so-called **Tikhonov (TIK)** and **Total Variation (TV)** regularizers. Namely, we consider the four following (popular, convex) variational models for image restoration:

TIK-$\mathrm{L}_2/\mathrm{L}_1$ variational models

$$u^* \in \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \underbrace{\frac{1}{2}\sum_{i=1}^{d} \left\| (\nabla u)_i \right\|_2^2}_{\text{TIK}} + \underbrace{\frac{\mu}{2 \text{ or } 1}\left\| Au - b \right\|_{2 \text{ or } 1}^{2 \text{ or } 1}}_{\mathrm{L}_{2/1}} \right\}$$

TV-$\mathrm{L}_2/\mathrm{L}_1$ variational models

$$u^* \in \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \underbrace{\sum_{i=1}^{d} \left\| (\nabla u)_i \right\|_2}_{\text{TV}} + \underbrace{\frac{\mu}{2 \text{ or } 1}\left\| Au - b \right\|_{2 \text{ or } 1}^{2 \text{ or } 1}}_{\mathrm{L}_{2/1}} \right\}$$

TIK/TV - L_2/L_1 MODELS: discrete gradient operator

Introducing matrix $D := \begin{pmatrix} D_h \\ D_v \end{pmatrix} \in \mathbb{R}^{2d\times d}$, with $D_h, D_v \in \mathbb{R}^{d\times d}$ coefficient matrices of linear finite difference operators discretizing horizontal and vertical partial derivates of a given (w x h = d)-pixels image, respectively, then, the models can be equivalently written as:

TIK-L_2/L_1 variational models

$$\boxed{u^* \in \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2}\| Du \|_2^2 + \frac{\mu}{2 \text{ or } 1}\| Au - b \|_{2 \text{ or } 1}^{2 \text{ or } 1} \right\}}$$

TIK TV $L_{2/1}$

TV-L_2/L_1 variational models

$$\boxed{u^* \in \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \sum_{i=1}^{d} \sqrt{\left(D^{(h)}u\right)_i^2 + \left(D^{(v)}u\right)_i^2} + \frac{\mu}{2 \text{ or } 1}\| Au - b \|_{2 \text{ or } 1}^{2 \text{ or } 1} \right\}}$$

Standard (and, in practice, reasonable) assumption: $\boxed{\ker(D) \cap \ker(A) = \{0_d\}}$

TIK/TV - L_2/L_1 MODELS: solution existence/uniqueness

- all functions J continuous, coercive, convex → local minimizers are global, minimizers exist
- TIK-L_2, TIK-L_1, TV-L_2 admit a unique solution (global minimizer) → adjust notations
- TV-L_1 admits a compact convex set of solutions (global minimizers)

TIK - L_2

$$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2}\| Du \|_2^2 + \frac{\mu}{2}\| Au-b \|_2^2 \right\}$$

TIK - L_1

$$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2}\| Du \|_2^2 + \mu\| Au-b \|_1 \right\}$$

TV - L_2

$$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \sum_{i=1}^{d}\sqrt{(D_h u)_i^2 + (D_v u)_i^2} + \frac{\mu}{2}\| Au-b \|_2^2 \right\}$$

TV - L_1

$$u^* \in \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \sum_{i=1}^{d}\sqrt{(D_h u)_i^2 + (D_v u)_i^2} + \mu\| Au-b \|_1 \right\}$$

It guarantees coercivity of all functions J

where: $\mu \in \mathbb{R}_{++}, \quad b \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d\times d}, \quad D = \begin{pmatrix} D_h \\ D_v \end{pmatrix} \in \mathbb{R}^{2d\times d}, \quad \boxed{\ker(D) \cap \ker(A) = \{0_d\}}$

TIK/TV - L_2/L_1 MODELS: **regularity (differentiability)**

For all models but the TIK-L_2 the cost function J is non-differentiable, due to the regularizer and/or to the fidelity term: —— differentiable —— non-differentiable

TIK - L_2 $$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ \boxed{J(u;\mu)} = \boxed{\frac{1}{2}\| Du \|_2^2} + \boxed{\frac{\mu}{2}\| Au-b \|_2^2} \right\}$$

TIK - L_1 $$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ \boxed{J(u;\mu)} = \boxed{\frac{1}{2}\| Du \|_2^2} + \boxed{\mu\| Au-b \|_1} \right\}$$

TV - L_2 $$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ \boxed{J(u;\mu)} = \boxed{\sum_{i=1}^{d}\sqrt{(D_h u)_i^2 + (D_v u)_i^2}} + \boxed{\frac{\mu}{2}\| Au-b \|_2^2} \right\}$$

TV - L_1 $$u^* \in \arg\min_{u\in\mathbb{R}^d} \left\{ \boxed{J(u;\mu)} = \boxed{\sum_{i=1}^{d}\sqrt{(D_h u)_i^2 + (D_v u)_i^2}} + \boxed{\mu\| Au-b \|_1} \right\}$$

where: $$\mu \in \mathbb{R}_{++}, \quad b \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d\times d}, \quad D = \begin{pmatrix} D_h \\ D_v \end{pmatrix} \in \mathbb{R}^{2d\times d}, \quad \boxed{\ker(D) \cap \ker(A) = \{0_d\}}$$

TIK/TV - L_2/L_1 MODELS: numerical solution

We are going to see how to solve numerically the models of interest by means of two state-of-the-art effective iterative methods:

- **ADMM** (**A**lternating **D**irection **M**ethod of **M**ultipliers)
- **QMM** (**Q**uadratic **M**ajorization-**M**inimization approach)

ADMM solves directly the original models, also if they aren't differentiable, whereas QMM requires differentiability $\rightarrow$ the original models will be slightly changed (or «smoothed») into very similar differentiable models that, then, can be solved by QMM

Solutions by ADMM and QMM of the simplest among the four models, i.e. the TIK-L_2 quadratic model, coincide, in the sense that they reduce to solving the same linear system

Outline:

- Direct solution of TIK-L_2 model by **DFT or CG / other direct methods**
- Iterative solution of TV-L_2, TIK-L_1, TV-L_1 models by **ADMM**
- Iterative solution of TV-L_2, TIK-L_1, TV-L_1 models by **QMM**

“DIRECT”, EFFICIENT

- by using the 2D Discrete Fourier Transform (DFT), implemented by the 2D Fast Fourier Transform (FFT), for image restoration, the iterative (Preconditioned) Conjugate Gradient (CG) method or some direct method for image inpainting -

numerical solution of the

(unconstrained)

TIK-L_2 (quadratic) model

Direct solution of the TIK-L_2 model

$$u^* = \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2}\| D_h u \|_2^2 + \frac{1}{2}\| D_v u \|_2^2 + \frac{\mu}{2}\| Au - b \|_2^2 \right\}$$

The cost function J is **convex quadratic**, and takes the equivalent standard form:

$$J(u;\mu) = \frac{1}{2} u^T H\, u - v^T u + c, \quad \text{with}: \quad H = D_h^T D_h + D_v^T D_v + \mu\, A^T A \in \mathbb{R}^{d \times d}$$

$$v = \mu\, A^T b \in \mathbb{R}^d, \qquad c = \frac{\mu}{2}\|b\|_2^2 \in \mathbb{R}_+$$

can you prove it?

The matrix H, which represents the (constant) Hessian matrix of J, **is positive definite**, hence J is strongly convex and admits a unique global minimizer which can be obtained by imposing the first-order optimality conditions for J, namely:

$$u^* \text{ solution of}: \;\; \nabla J(u) = 0_d \;\Leftrightarrow\; H\, u = v \;\Leftrightarrow\; \left(D_h^T D_h + D_v^T D_v + \mu\, A^T A \right) u = \mu\, A^T b$$

Since H is positive definite, it has full rank → the linear system **admits a unique solution** u^*. However, the linear system is commonly of huge size (d x d, with d number of pixels). Can we solve it efficiently, possibly avoiding to form (and store) explicitly the matrix H ?

YES, IN BOTH THE RESTORATION AND INPAINTING CASES ...

Direct solution of the TIK-L_2 model: restoration

$$u^* = \arg\min_{u\in\mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2}\| D_h u \|_2^2 + \frac{1}{2}\| D_v u \|_2^2 + \frac{\mu}{2}\| K u - b \|_2^2 \right\}$$

$$u^* \text{ unique solution of : } \quad \left(D_h^T D_h + D_v^T D_v + \mu K^T K \right) u = \mu K^T b$$

$D_h, D_v, K \in \mathbb{R}^{d\times d}$ are convolution matrices, and we are assuming periodic boundary conditions for image *u*, thus they can be diagonalized (in $\mathbb{C}$) by the 2D Discrete Fourier Transform (DFT):

$$D_h = F^* \tilde{D}_h F, \quad D_v = F^* \tilde{D}_v F, \quad K = F^* \tilde{K} F, \quad \text{with : } \quad F,\, \tilde{D}_h,\, \tilde{D}_v,\, \tilde{K} \in \mathbb{C}^{d\times d},$$

F^* conjugate transpose of F, $\quad F^* F = F F^* = I_d$ (F unitary), $\quad F, F^*$ direct, inverse 2D DFT matrices,

and $\quad \tilde{D}_h = \mathrm{diag}\left(\tilde{D}_{h,1},\ldots,\tilde{D}_{h,d}\right), \quad \tilde{D}_v = \mathrm{diag}\left(\tilde{D}_{v,1},\ldots,\tilde{D}_{v,d}\right), \quad \tilde{K} = \mathrm{diag}\left(\tilde{K}_1,\ldots,\tilde{K}_d\right)$.

Substituting factorizations above in the linear system:

real, diagonal, full rank matrix

$$\left(\tilde{D}_h^* \tilde{D}_h + \tilde{D}_v^* \tilde{D}_v + \mu \tilde{K}^* \tilde{K} \right) F u = \mu \tilde{K}^* F b \;\Rightarrow\; \left(\left|\tilde{D}_h\right|^2 + \left|\tilde{D}_v\right|^2 + \mu \left|\tilde{K}\right|^2 \right) F u = \mu \tilde{K}^* F b$$

$$\Rightarrow\; F u = \mu \left(\left|\tilde{D}_h\right|^2 + \left|\tilde{D}_v\right|^2 + \mu \left|\tilde{K}\right|^2 \right)^{-1} \tilde{K}^* F b \;\Rightarrow\; u = \mu F^* \left(\left|\tilde{D}_h\right|^2 + \left|\tilde{D}_v\right|^2 + \mu \left|\tilde{K}\right|^2 \right)^{-1} \tilde{K}^* F b$$

Direct solution of the TIK-L_2 model: restoration

$$u^* = \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2} \| D_h u \|_2^2 + \frac{1}{2} \| D_v u \|_2^2 + \frac{\mu}{2} \| K u - b \|_2^2 \right\}$$

u^* unique solution of : $\quad \left(D_h^T D_h + D_v^T D_v + \mu K^T K \right) u = \mu K^T b$

$$\Downarrow$$

$$u^* = \mu \boxed{F^*} \left(\left(\left|\tilde{D}_h\right|^2 + \left|\tilde{D}_v\right|^2 + \mu \left|\tilde{K}\right|^2 \right)^{-1} \tilde{K}^* \boxed{F b} \right)$$

$$u^* = \mu * \boxed{\text{real}\Big(\text{ifft2}} \left(\left(\tilde{K}^* .* \boxed{\text{fft2(b)}} \right) ./ \ \text{diag}\left(\left|\tilde{D}_h\right|^2 + \left|\tilde{D}_v\right|^2 + \mu \left|\tilde{K}\right|^2 \right) \right) \Big)$$

How can we compute these terms in Matlab? We will see ...

Direct solution of the TIK-L$_2$ model: restoration

$$u^* = \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2} \| D_h u \|_2^2 + \frac{1}{2} \| D_v u \|_2^2 + \frac{\mu}{2} \| K u - b \|_2^2 \right\}$$

u^* unique solution of : $\left(D_h^T D_h + D_v^T D_v + \mu K^T K \right) u = \mu K^T b$

$$\Downarrow$$

$$u^* = \mu \boxed{F^*} \left(\left(|\tilde{D}_h|^2 + |\tilde{D}_v|^2 + \mu |\tilde{K}|^2 \right)^{-1} \tilde{K}^* \boxed{F b} \right)$$

$$u^* = \mu * \boxed{\mathrm{real}\Big(\mathrm{ifft2}}\Big(\left(\tilde{K}^* .* \boxed{\mathrm{fft2(b)}} \right) ./ \ \mathrm{diag}\left(|\tilde{D}_h|^2 + |\tilde{D}_v|^2 + \mu |\tilde{K}|^2 \right) \Big)\Big)$$

How can we compute these terms in Matlab? We will see ...

BY USING 2D FFT, COMPUTATIONAL COMPLEXITY is O(d log d) ... VERY FAST !!

Direct solution of the TIK-L_2 model: inpainting

$$u^* = \arg\min_{u \in \mathbb{R}^d} \left\{ J(u;\mu) = \frac{1}{2} \| D_h u \|_2^2 + \frac{1}{2} \| D_v u \|_2^2 + \frac{\mu}{2} \| S u - b \|_2^2 \right\}$$

$$u^* \text{ unique solution of : } \quad \left(D_h^T D_h + D_v^T D_v + \mu S^T S \right) u = \mu S^T b$$

S is a diagonal binary matrix, hence $S = S^T = S^T S$; moreover, $S^T b = b$; the problem thus reads

$$\left(D_h^T D_h + D_v^T D_v + \mu S \right) u = \mu b$$

The finite difference matrices D_h, D_v discretizing the horizontal/vertical partial first-order derivatives are very sparse, then the coefficient matrix of the linear system above is very sparse, as well as symmetric and positive definite.

Moreover, D_h^T, D_h, D_v^T, D_v, S, can be very efficiently implemented as operators without forming explicitly the matrices.

It is thus convenient to solve the system by using the Conjugate Gradient method, eventually preconditioned, or some direct method for sparse systems.