Pre-course: imaging and learning, part I

(images, noise, Bayesian/variational formulation, optimisation)

Luca Calatroni

MaLGa Center, Department of Computer Science, University of Genoa, Italy

Mathematics & Machine Learning for image analysis

ERASMUS+ International PhD Summer School

Bologna, June 3-11 2025

Co-funded by the European Union (ERC, MALIN, 10117133). Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the EU nor the granting authority can be held responsible for them.

Meet the instructors (Luca's) & program

Luca Calatroni
Part I (14:30-15:45)

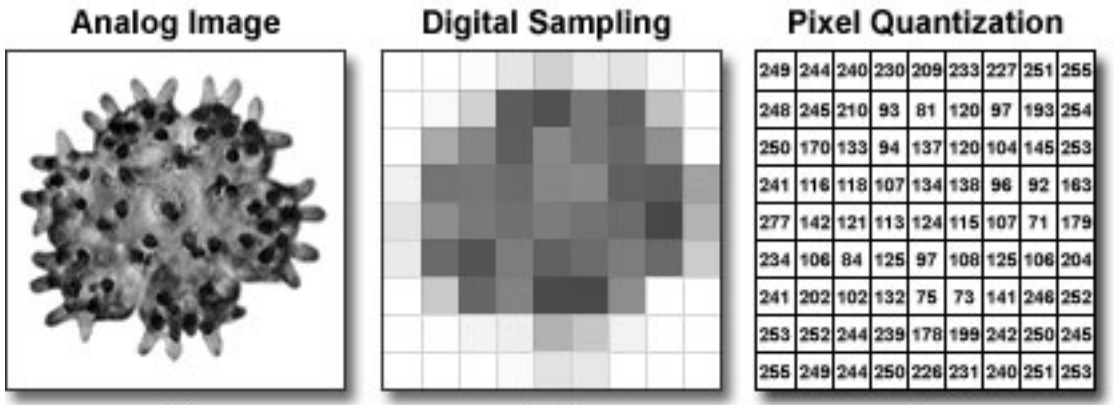
- Digital images
- Modelling noise
- Quality measures (MSE, PSNR, SSIM)
- Image denoising as a toy 'inverse' problem
- Bayesian formulation
- MAP estimators
- Regularisation
- Bits of optimisation: GD, proximal operator

Luca Ratti
Part II (16:00-17:15)

- Image denoising as a regression problem
- Supervised/unsupervised learning
- Neural networks
- Bias-variance tradeoff
- Over-parametrisation
- NNs for imaging: CNN, U-NET
- Training a NN: backpropagation, batches, optimisers..
- NumPy/Matplotlib/Pytorch (tomorrow)

Digital images

Digital images are discrete representations of the continuous world we live in.



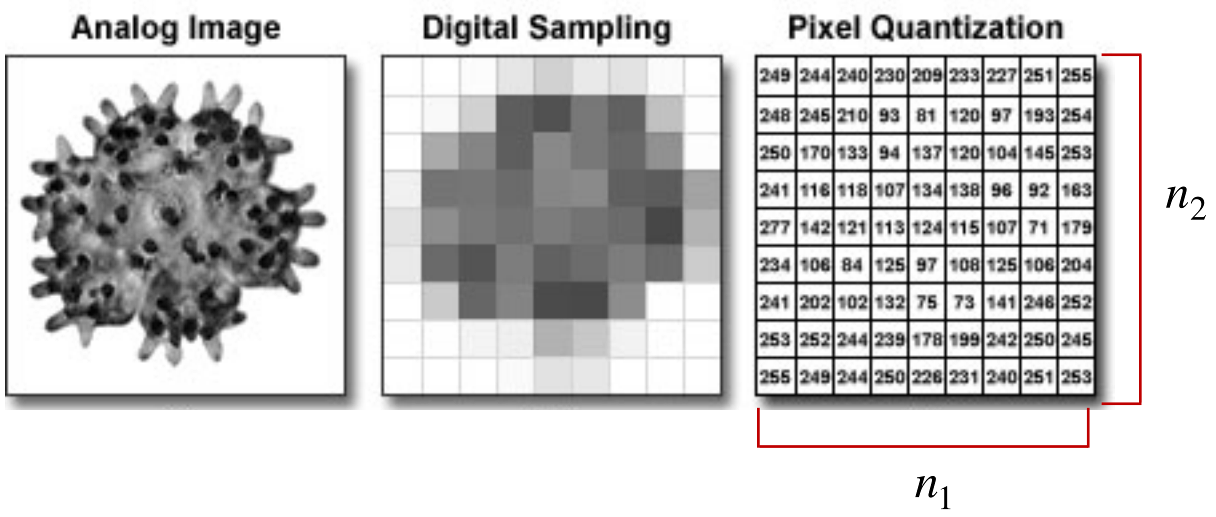
249	244	240	230	209	233	227	251	255
248	245	210	93	81	120	97	193	254
250	170	133	94	137	120	104	145	253
241	116	118	107	134	138	96	92	163
277	142	121	113	124	115	107	71	179
234	106	84	125	97	108	125	106	204
241	202	102	132	75	73	141	246	252
253	252	244	239	178	199	242	250	245
255	249	244	250	226	231	240	251	253

Sampling: allows to represent a continuous image into a finite (pixel) grid.

Quantisation: assigns a grey-level describing average brightness at each pixel.

Digital images

Digital images are discrete representations of the continuous world we live in.



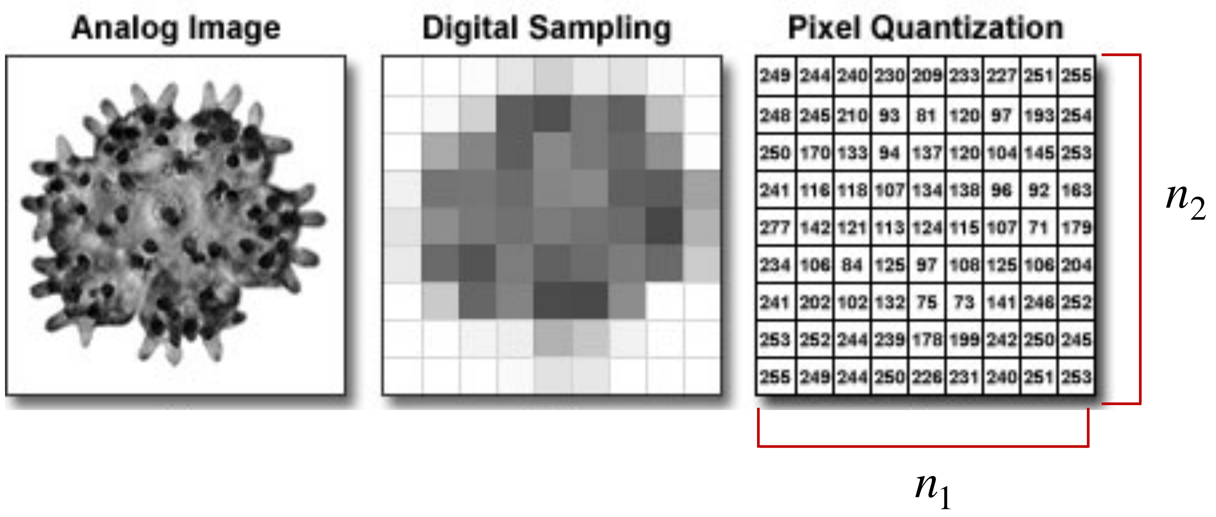
Sampling: allows to represent a continuous image into a finite (pixel) grid.

Quantisation: assigns a grey-level describing average brightness at each pixel.

$$\Omega = \{1,\ldots,n_1\} \times \{1,\ldots,n_2\} \longrightarrow \text{image domain}$$

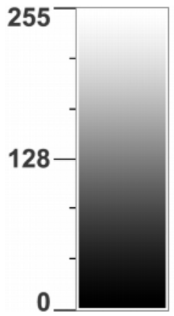
Digital images

Digital images are discrete representations of the continuous world we live in.



Sampling: allows to represent a continuous image into a finite (pixel) grid.

Quantisation: assigns a grey-level describing average brightness at each pixel.

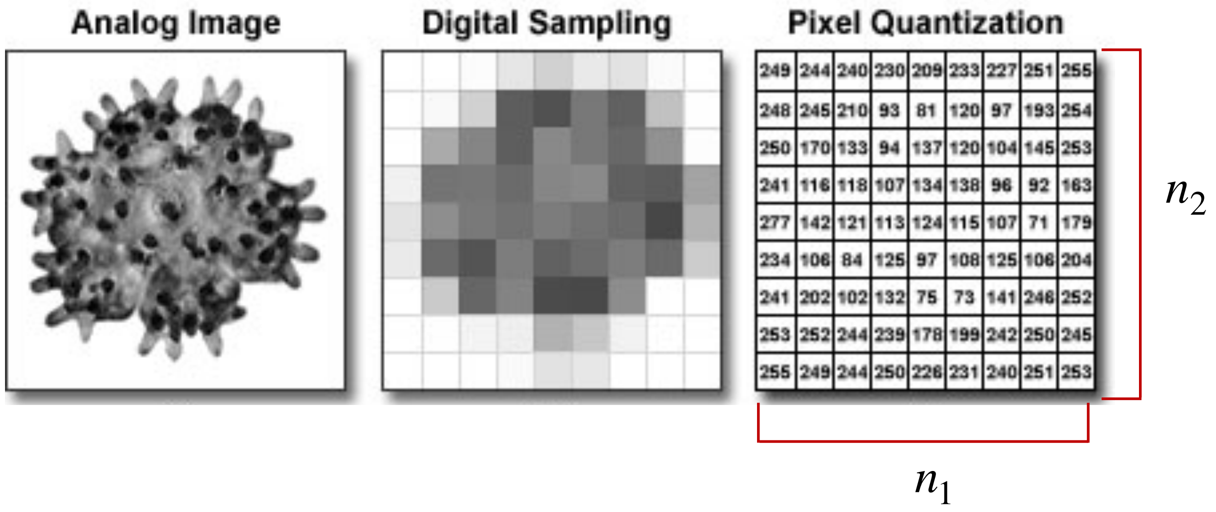


$$\Omega = \{1,\ldots,n_1\} \times \{1,\ldots,n_2\} \longrightarrow \text{image domain}$$

$$\mathbf{X} : \Omega \to \{0,\ldots,255\} \qquad \mathbf{X} = (x_{i,j}) \in \{0,\ldots,255\}^{n_1 \times n_2}$$

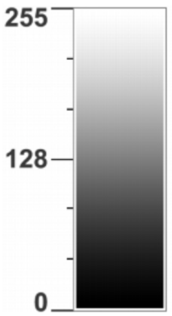
Digital images

Digital images are discrete representations of the continuous world we live in.



Sampling: allows to represent a continuous image into a finite (pixel) grid.

Quantisation: assigns a grey-level describing average brightness at each pixel.

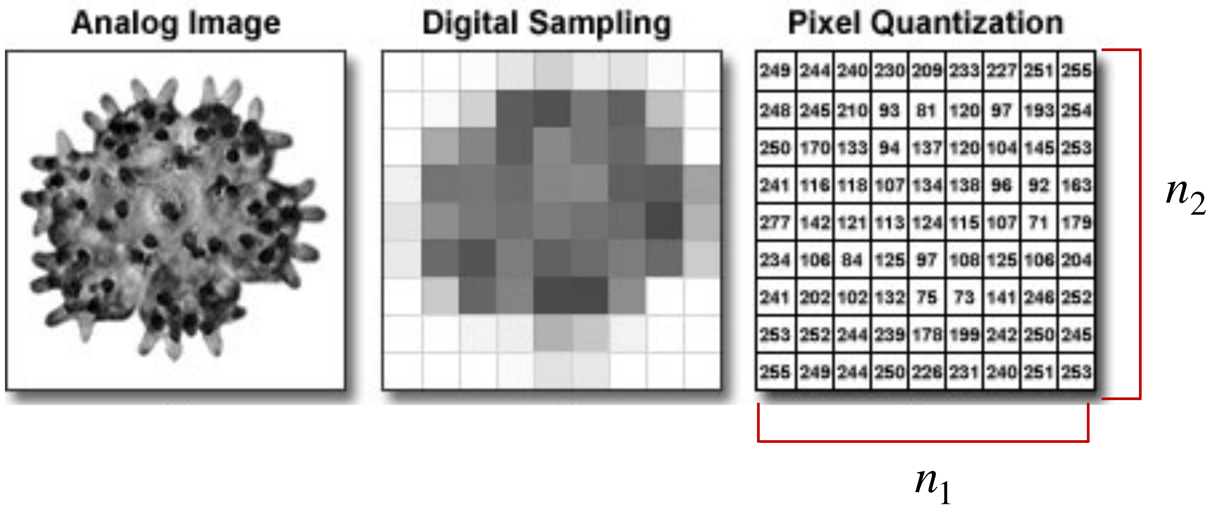


$\Omega = \{1,\ldots,n_1\} \times \{1,\ldots,n_2\}$ $\longrightarrow$ image domain

$\mathbf{X} : \Omega \to \{0,\ldots,255\}$ $\quad$ $\mathbf{X} = (x_{i,j}) \in \{0,\ldots,255\}^{n_1 \times n_2}$ $\xrightarrow[\text{normalisation, adjustments..}]{}$ $\mathbf{X} \in [0,1]^{n_1 \times n_2}$ $\quad$ $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$

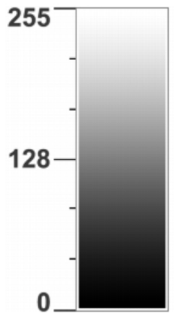
Digital images

Digital images are discrete representations of the continuous world we live in.



Sampling: allows to represent a continuous image into a finite (pixel) grid.

Quantisation: assigns a grey-level describing average brightness at each pixel.



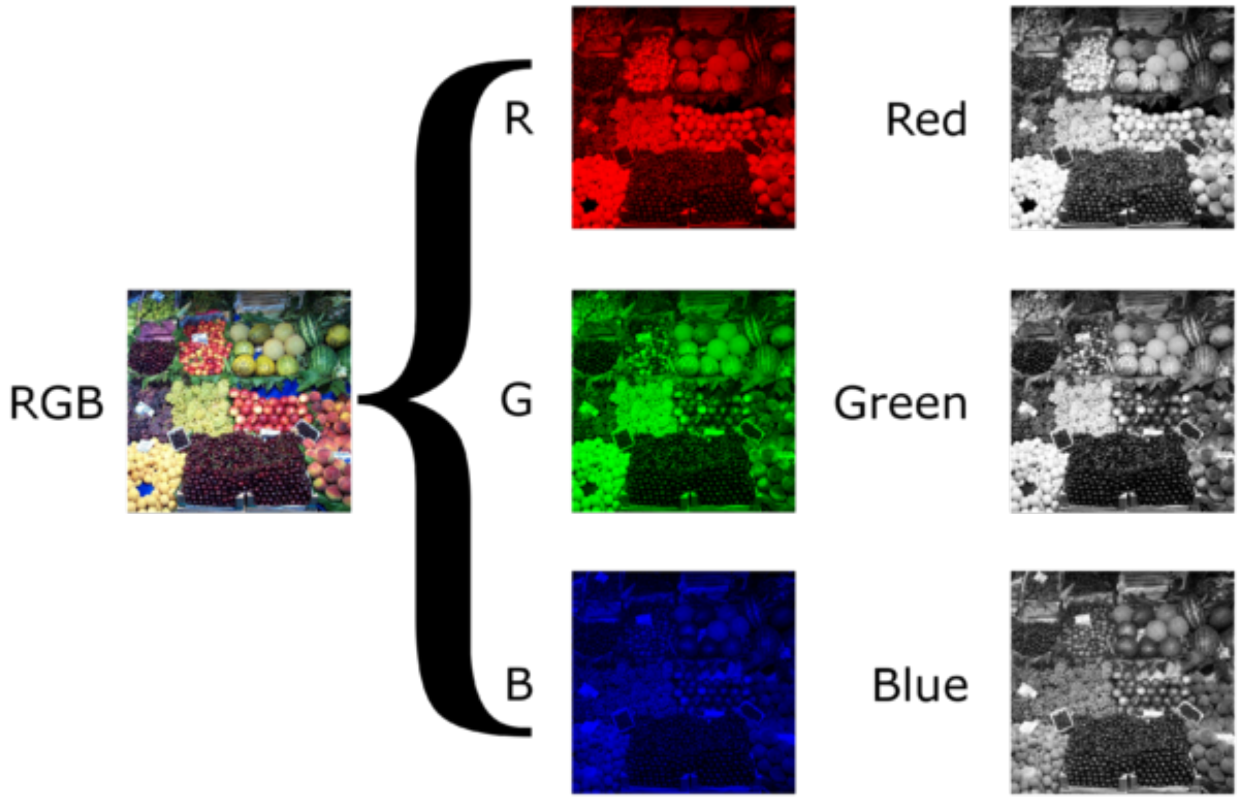
$\Omega = \{1,\ldots,n_1\} \times \{1,\ldots,n_2\}$ $\longrightarrow$ image domain

$\mathbf{X} : \Omega \to \{0,\ldots,255\}$ $\quad \mathbf{X} = (x_{i,j}) \in \{0,\ldots,255\}^{n_1 \times n_2}$ $\xrightarrow[\text{normalisation, adjustments..}]{}$ $\mathbf{X} \in [0,1]^{n_1 \times n_2}$ $\quad \mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$

Upon vectorisation of the 2D image $\mathbf{X}$, consider a vector $\mathbf{x} \in \mathbb{R}^n$, with $n = n_1 n_2$.

Grayscale/RGB images

Natural scenes are not grayscale. Color is a combination of Red-Green-Blue channels.



The higher the intensity in the individual channel, the more represented is the color.

$$x_{i,j} = (r_{i,j}, g_{i,j}, b_{i,j}) \in \mathbb{R}^3$$

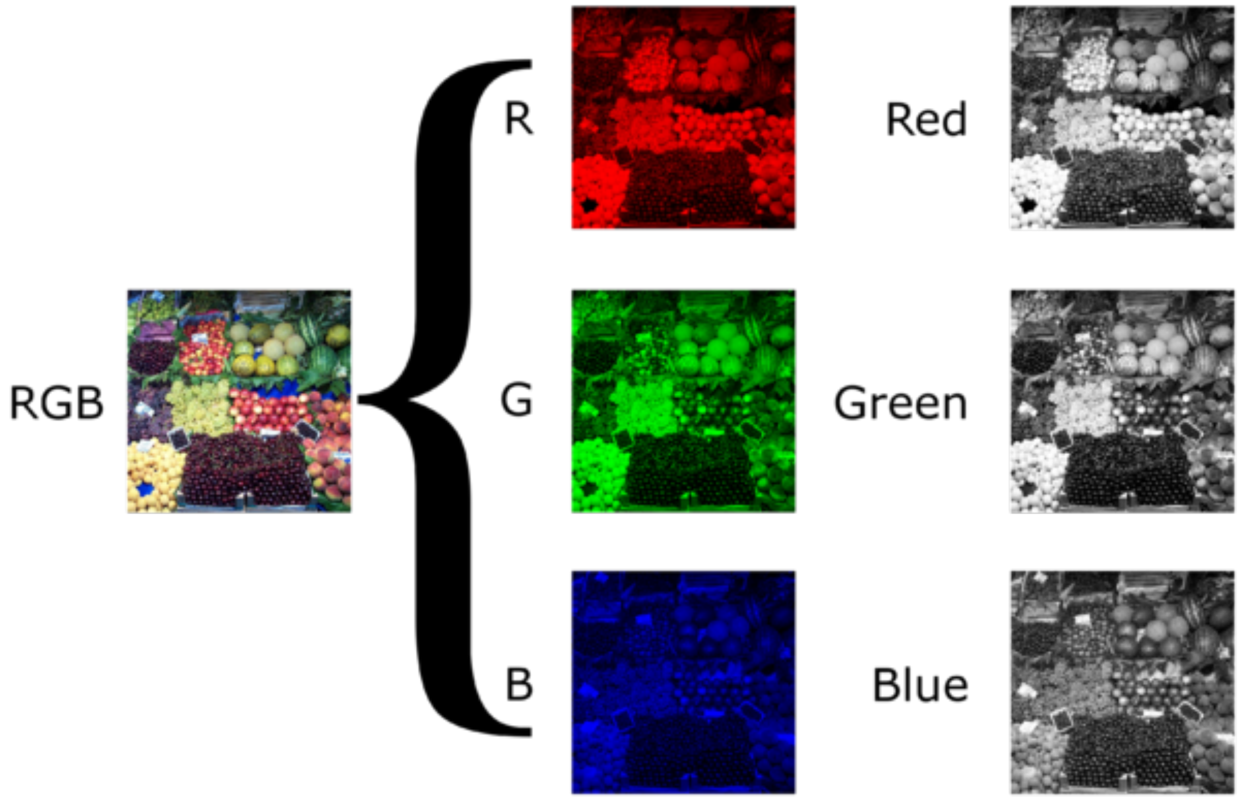
Hence, for RGB images:

$$\mathbf{x} \in \mathbb{R}^{n \times 3} \quad \text{or} \quad \mathbb{R}^{3n}$$

- Other color spaces are possible (CMYK, HSV..)
- Often, color channels are processed separately.

Grayscale/RGB images

Natural scenes are not grayscale. Color is a combination of Red-Green-Blue channels.



The higher the intensity in the individual channel, the more represented is the color.

$$x_{i,j} = (r_{i,j}, g_{i,j}, b_{i,j}) \in \mathbb{R}^3$$

Hence, for RGB images:

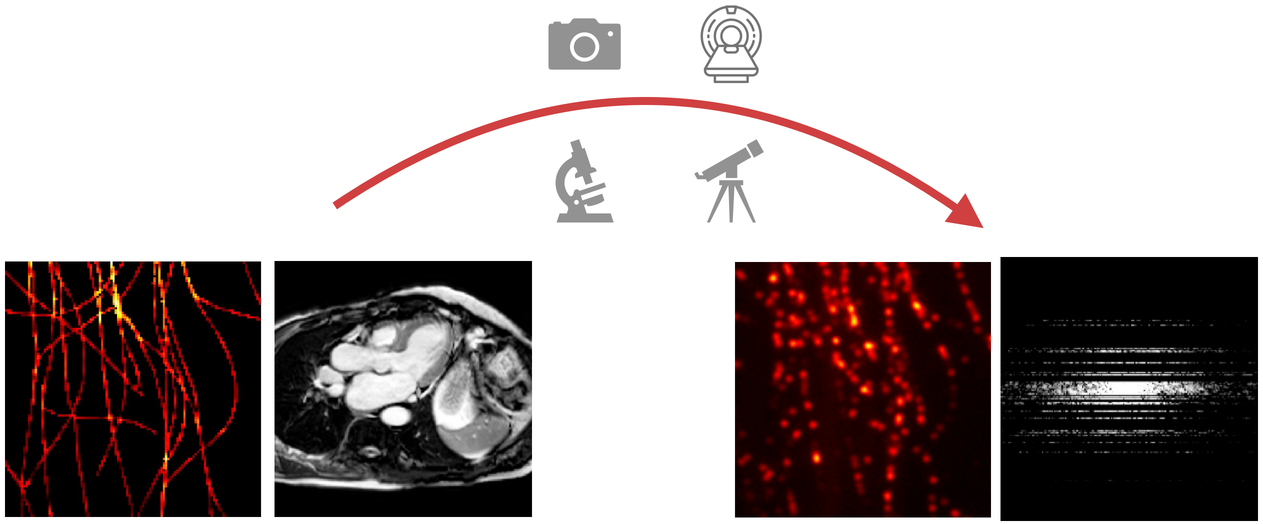
$$\mathbf{x} \in \mathbb{R}^{n \times 3} \quad \text{or} \quad \mathbb{R}^{3n}$$

- Other color spaces are possible (CMYK, HSV..)
- Often, color channels are processed separately.

Assume $\mathbf{x}$ is grayscale (extension is trivial)

Modelling degradation processes

Images are matrices/vectors. How to model acquisition processes?



$\mathbf{x} \in \mathbb{R}^n$
(unknown)

$\mathbf{y} \in \mathbb{R}^m$
(observed)

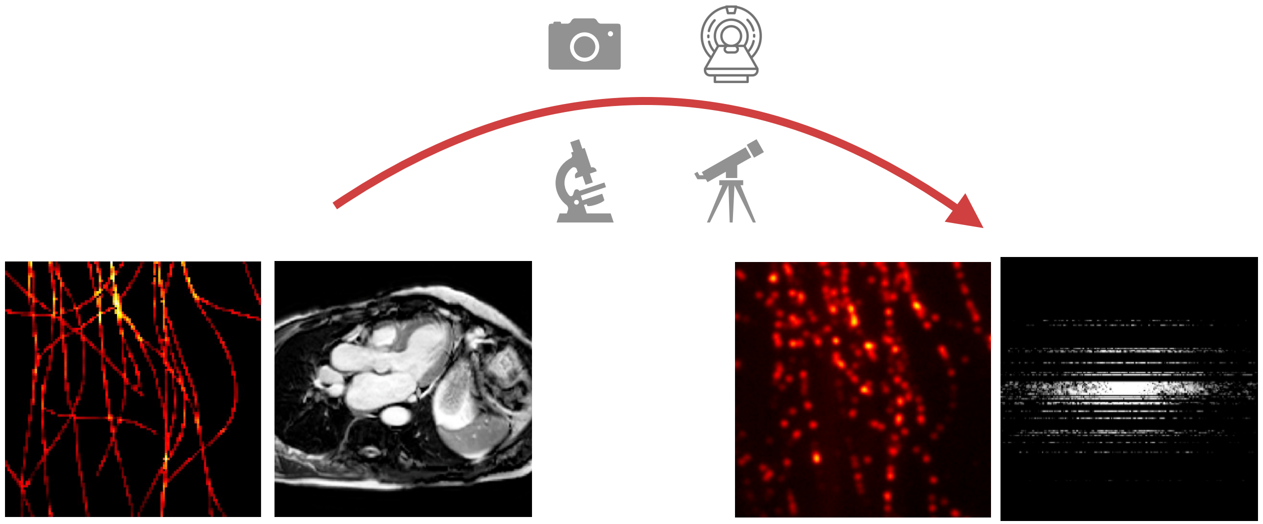
$$\boxed{\mathbf{y} = \mathbf{A}\mathbf{x}}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

linear input ($\mathbf{x}$)-output ($\mathbf{y}$) relation

Modelling degradation processes

Images are matrices/vectors. How to model acquisition processes?



$\mathbf{x} \in \mathbb{R}^n$

(unknown)

$\mathbf{y} \in \mathbb{R}^m$

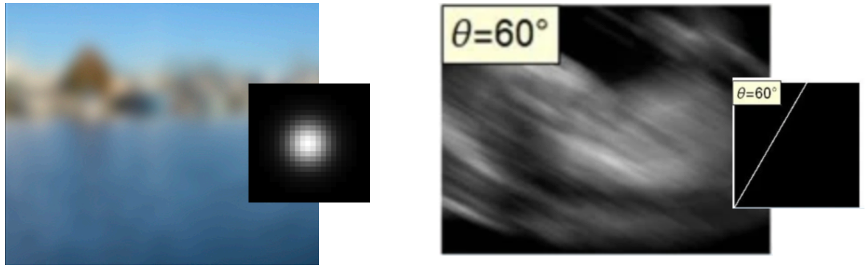
(observed)

$$\boxed{\mathbf{y} = \mathbf{A}\mathbf{x}}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

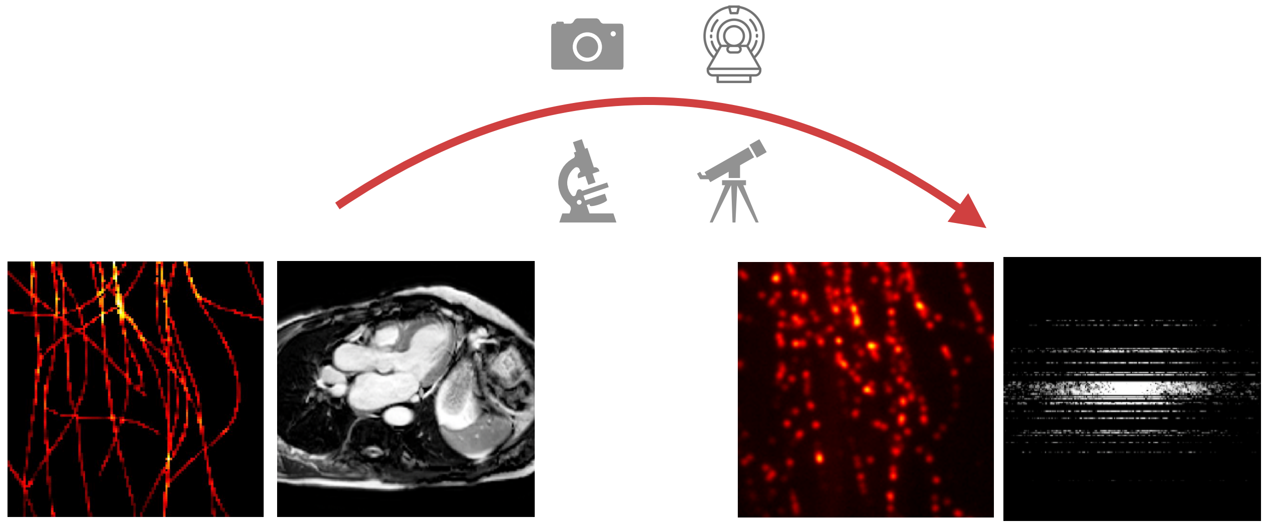
linear input ($\mathbf{x}$)-output ($\mathbf{y}$) relation

- **Convolution:** $\mathbf{A}\mathbf{x} \leftrightarrow h * \mathbf{X}$, h is a kernel



Modelling degradation processes

Images are matrices/vectors. How to model acquisition processes?



$\mathbf{x} \in \mathbb{R}^n$
(unknown)

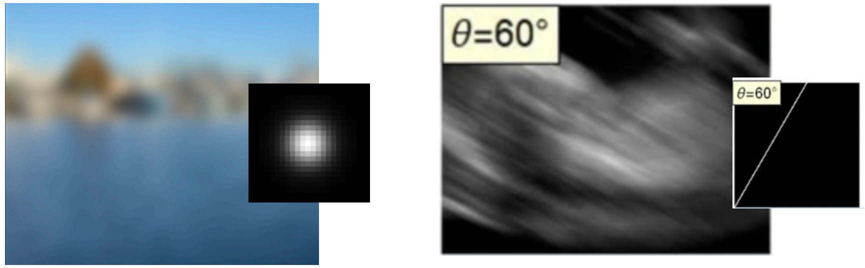
$\mathbf{y} \in \mathbb{R}^m$
(observed)

$$\boxed{\mathbf{y} = \mathbf{A}\mathbf{x}}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

linear input ($\mathbf{x}$)-output ($\mathbf{y}$) relation

- **Convolution:** $\mathbf{A}\mathbf{x} \leftrightarrow h * \mathbf{X}$, h is a kernel

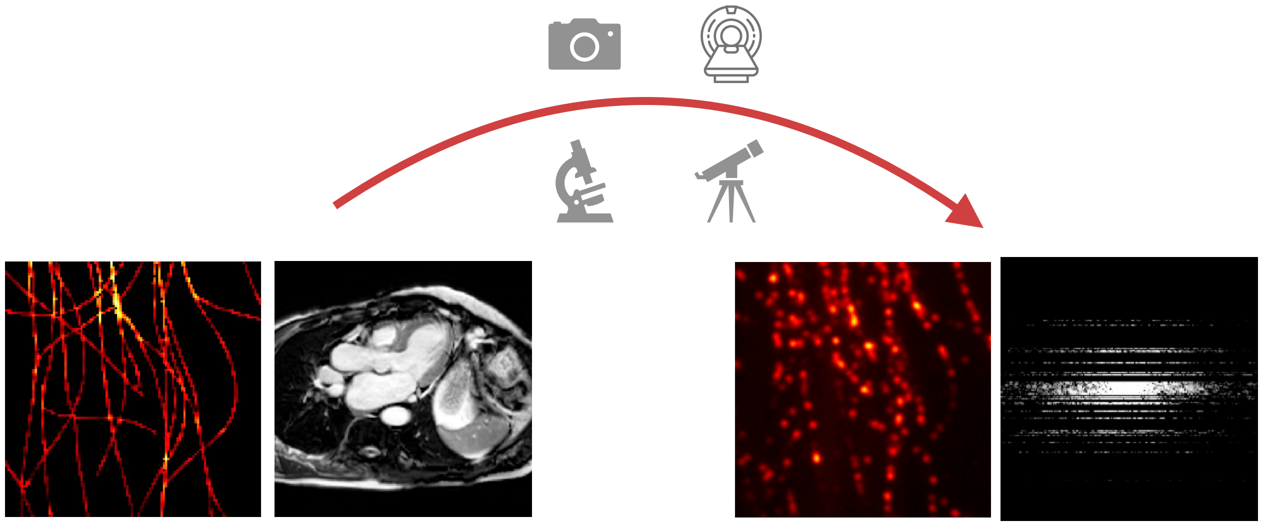


- **Masking:**

$$\mathbf{A} = \mathbf{M} \in \{0,1\}^{m \times n}$$

Modelling degradation processes

Images are matrices/vectors. How to model acquisition processes?



$\mathbf{x} \in \mathbb{R}^n$
(unknown)

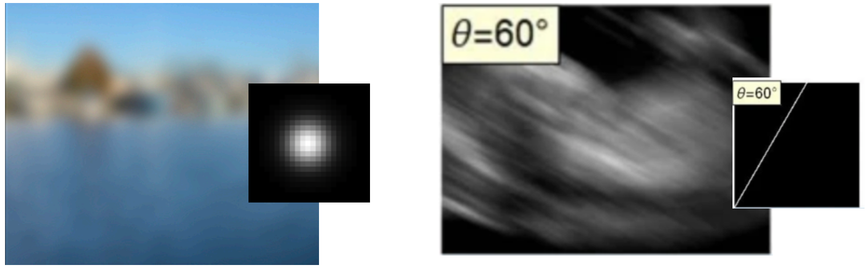
$\mathbf{y} \in \mathbb{R}^m$
(observed)

$$\boxed{\mathbf{y} = \mathbf{A}\mathbf{x}}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

linear input ($\mathbf{x}$)-output ($\mathbf{y}$) relation

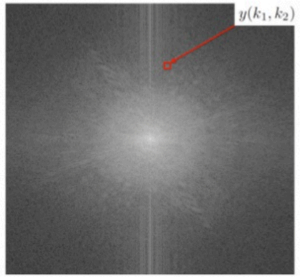
- **Convolution:** $\mathbf{A}\mathbf{x} \leftrightarrow h * \mathbf{X}$, h is a kernel



- **Masking:**

$$\mathbf{A} = \mathbf{M} \in \{0,1\}^{m \times n}$$

- **Fourier transform + Masking:**



$$\mathbf{A}\,\cdot = \mathbf{M}\mathscr{F}(\,\cdot\,)$$

Modelling noise

Acquisitions are never perfect. Interferences, errors, faults may happen.

$$\mathbf{y} = \text{Noise}(\mathbf{A}\mathbf{x})$$

Noise : $\mathbb{R}^m \to \mathbb{R}^m$ codifies instrumental errors.

In the following: $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{n \times n}$. How to model noise?

Modelling noise

Acquisitions are never perfect. Interferences, errors, faults may happen.

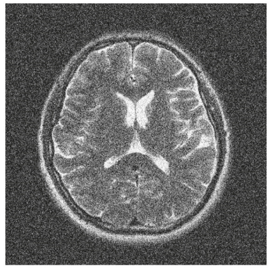
$$\mathbf{y} = \text{Noise}(\mathbf{A}\mathbf{x})$$

Noise : $\mathbb{R}^m \to \mathbb{R}^m$ codifies instrumental errors.

In the following: $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{n \times n}$. How to model noise?

Gaussian noise:

$$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$



Mostly used due to CLT.
Models signal-independent electronic noise

Modelling noise

Acquisitions are never perfect. Interferences, errors, faults may happen.

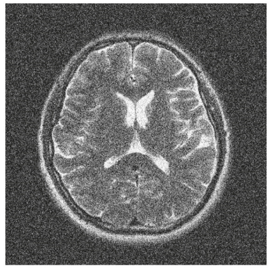
$$\mathbf{y} = \text{Noise}(\mathbf{A}\mathbf{x})$$

Noise : $\mathbb{R}^m \to \mathbb{R}^m$ codifies instrumental errors.

In the following: $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{n \times n}$. How to model noise?

Gaussian noise:

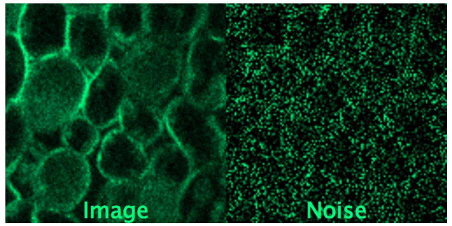
$$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$



Mostly used due to CLT.
Models signal-independent electronic noise

Poisson noise:

$$\text{Noise}(\mathbf{x}) = \text{Pois}(\mathbf{x}+\boldsymbol{\beta}), \quad \mathbf{x} \in \mathbb{R}^n_{\geq 0}, \ \boldsymbol{\beta} \in \mathbb{R}^n_{>0}$$



Used in low-photon imaging.
Astronomical, microscopy imaging.

Bertero, Boccacci, '98

Modelling noise

Acquisitions are never perfect. Interferences, errors, faults may happen.

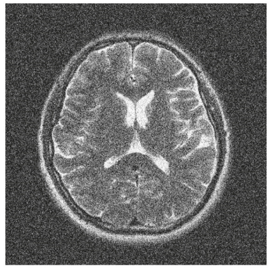
$$\mathbf{y} = \text{Noise}(\mathbf{A}\mathbf{x})$$

Noise : $\mathbb{R}^m \to \mathbb{R}^m$ codifies instrumental errors.

In the following: $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{n \times n}$. How to model noise?

Gaussian noise:

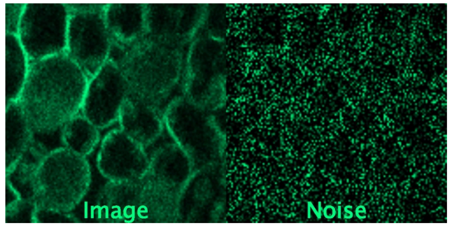
$$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$



Mostly used due to CLT.
Models signal-independent electronic noise

Poisson noise:

$$\text{Noise}(\mathbf{x}) = \text{Pois}(\mathbf{x}+\boldsymbol{\beta}), \quad \mathbf{x} \in \mathbb{R}^n_{\geq 0}, \ \boldsymbol{\beta} \in \mathbb{R}^n_{>0}$$



Used in low-photon imaging.
Astronomical, microscopy imaging.

Bertero, Boccacci, '98

Impulsive noise:

$$\text{Noise}(\mathbf{x}) = (\mathbf{1} - \mathbf{s}) \odot \mathbf{x} + \mathbf{s} \odot \mathbf{c}$$

$$c_i = \mathscr{B}(1/2), \ s_i = \mathscr{B}(p), p \in [0,1]$$

Used to describe faulty detectors
and/or long time exposures under bad lighting.

Quality metrics: MSE, SNR, PSNR

How to assess image quality between two images using **pixel information**?

Quality metrics: MSE, SNR, PSNR

How to assess image quality between two images using **pixel information**?

$$\mathrm{MSE}(\mathbf{y}, \mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|^2 = \frac{1}{n_1}\frac{1}{n_2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} |y_{i_1,i_2} - x_{i_1,i_2}|^2$$

Quality metrics: MSE, SNR, PSNR

How to assess image quality between two images using **pixel information**?

$$\mathrm{MSE}(\mathbf{y}, \mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|^2 = \frac{1}{n_1}\frac{1}{n_2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} |y_{i_1,i_2} - x_{i_1,i_2}|^2$$

$$\mathrm{SNR}(\mathbf{y}, \mathbf{x}) = 10\log_{10}\left(\frac{\|\mathbf{x}\|^2}{\|\mathbf{x} - \mathbf{y}\|^2}\right)$$

noise $\boldsymbol{\varepsilon}$

Quality metrics: MSE, SNR, PSNR

How to assess image quality between two images using **pixel information**?

$$\mathrm{MSE}(\mathbf{y}, \mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|^2 = \frac{1}{n_1}\frac{1}{n_2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} |y_{i_1,i_2} - x_{i_1,i_2}|^2$$

$$\mathrm{SNR}(\mathbf{y}, \mathbf{x}) = 10\log_{10}\left(\frac{\|\mathbf{x}\|^2}{\|\mathbf{x}-\mathbf{y}\|^2}\right)$$

noise $\boldsymbol{\varepsilon}$

$$\mathrm{PSNR}(\mathbf{y}, \mathbf{x}) = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}(\mathbf{y}, \mathbf{x})}\right)$$

where MAX is the highest possible value (e.g., 255 or 1)

Quality metrics: MSE, SNR, PSNR

How to assess image quality between two images using **pixel information**?

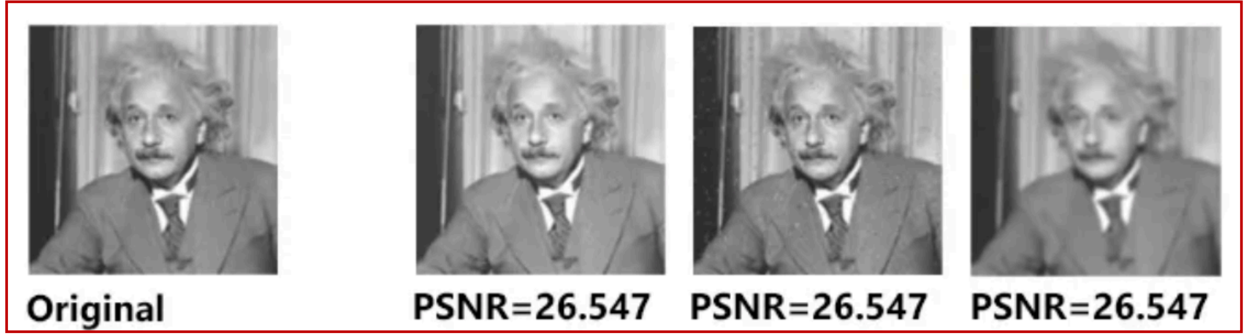
$$\text{MSE}(\mathbf{y}, \mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|^2 = \frac{1}{n_1}\frac{1}{n_2}\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} |y_{i_1,i_2} - x_{i_1,i_2}|^2$$

$$\text{SNR}(\mathbf{y}, \mathbf{x}) = 10\log_{10}\left(\frac{\|\mathbf{x}\|^2}{\|\mathbf{x}-\mathbf{y}\|^2}\right)$$

noise $\boldsymbol{\varepsilon}$

$$\text{PSNR}(\mathbf{y}, \mathbf{x}) = 10\log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}(\mathbf{y}, \mathbf{x})}\right)$$

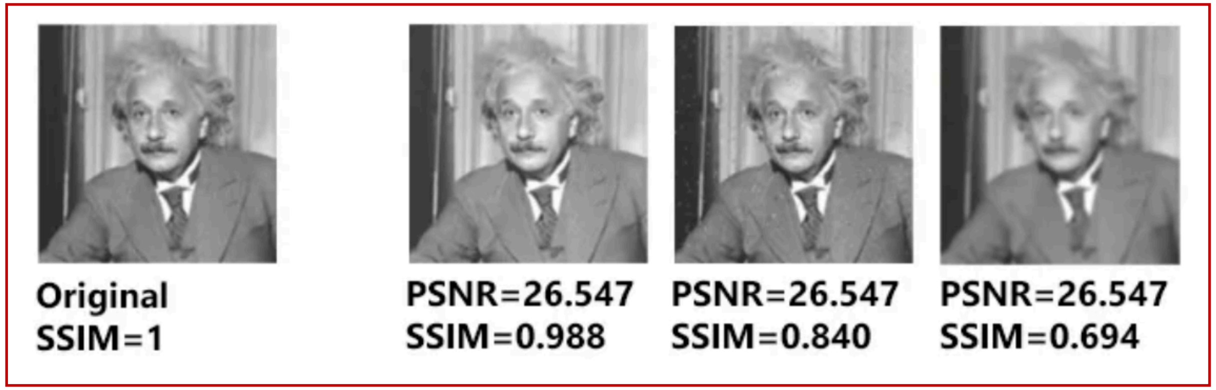
where MAX is the highest possible value (e.g., 255 or 1)



Sensitive to **pixel variations**
Is it a good quality metric for natural images?!

Quality metrics: SSIM

$$\text{SSIM}(x,y) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2)} \in [0,1]$$



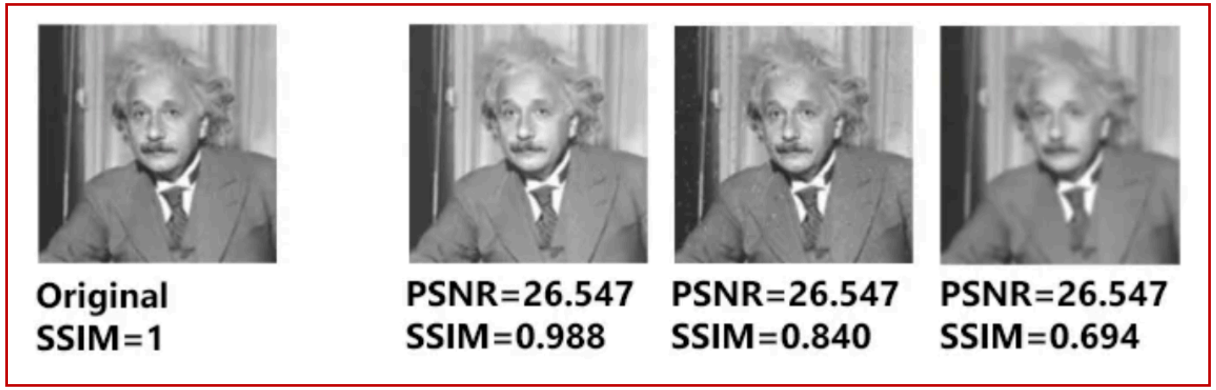
- Based on image statistics: mean, variance, covariance
+ constant C_1, C_2 stabilising the division.

- Typically performed on small image patches + averaging

Wang, Bovik, Sheikh, Simoncelli, ‘04

Quality metrics: SSIM

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2)} \in [0,1]$$



- Based on image statistics: mean, variance, covariance
+ constant C_1, C_2 stabilising the division.

- Typically performed on small image patches + averaging

	RANGE	IDEAL	GOOD
MSE	$[0,\mathrm{MAX}^2]$	0	< 100 (for images in [0,255])
SNR	$(-\infty, +\infty)$	$+\infty$	> 30
PSNR	$[0, +\infty)$	$+\infty$	> 30
SSIM	$[0,1]$	1	> 0.9

All such metrics are **supervised**.
They depend on ground-truth $\mathbf{x}$.

Wang, Bovik, Sheikh, Simoncelli, '04

Image denoising as a ‘toy’ inverse problem

Given $\mathbf{y} \in \mathbb{R}^n$, find $\mathbf{x} \in \mathbb{R}^n$ such that:

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

- “Inverse problem” (operator to invert is $\mathbf{A} = \mathbf{I}$)
- Still challenging: the noise realisation is unknown
- If noise is high (σ is big), image content can be lost

$\sigma = 0.002$ $\sigma = 0.05$ $\sigma = 0.3$

$\mathbf{x} \in [0,1]^{3n}$

How to model this problem in mathematical terms?

MSE

PSNR/SSIM

Bayesian formulation

Idea: model also $\mathbf{x}$ and $\mathbf{y}$ as realisation of random variables $\mathscr{X} \sim \pi_{\mathscr{X}}, \mathscr{Y} \sim \pi_{\mathscr{Y}}$ + conditional laws.

Bayesian formulation

Idea: model also $\mathbf{x}$ and $\mathbf{y}$ as realisation of random variables $\mathscr{X} \sim \pi_{\mathscr{X}}, \mathscr{Y} \sim \pi_{\mathscr{Y}}$ + conditional laws.

$\boxed{\pi_{\mathscr{X}|\mathscr{Y}}(\mathbf{x}\,|\,\mathbf{y})}$ **Posterior distribution:** what we would like to maximise.

$\boxed{\pi_{\mathscr{Y}|\mathscr{X}}(\mathbf{y}\,|\,\mathbf{x})}$ **Likelihood:** function describing the probability of observing the data given a choice of $\mathbf{x}$

$\boxed{\pi_{\mathscr{Y}}(\mathbf{y})}$ **Evidence term:** normally neglected, does not depend on $\mathbf{x}$

$\boxed{\pi_{\mathscr{X}}(\mathbf{x})}$ **Prior:** prior assumptions on the unknown quantity $\mathbf{x}$

Bayesian formulation

Idea: model also $\mathbf{x}$ and $\mathbf{y}$ as realisation of random variables $\mathcal{X} \sim \pi_{\mathcal{X}}, \mathcal{Y} \sim \pi_{\mathcal{Y}}$ + conditional laws.

$\boxed{\pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y})}$ **Posterior distribution:** what we would like to maximise.

$\boxed{\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})}$ **Likelihood:** function describing the probability of observing the data given a choice of $\mathbf{x}$

$\boxed{\pi_{\mathcal{Y}}(\mathbf{y})}$ **Evidence term:** normally neglected, does not depend on $\mathbf{x}$

$\boxed{\pi_{\mathcal{X}}(\mathbf{x})}$ **Prior:** prior assumptions on the unknown quantity $\mathbf{x}$

$$\boxed{\pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y}) = \frac{\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})}{\pi_{\mathcal{Y}}(\mathbf{y})}}$$

Bayes' Theorem

Under Gaussian noise assumption:

$$\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}) = \pi_E(\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{x})$$

Bayesian formulation: priors, score functions

$$\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x}} \pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y}) = \operatorname{argmax}_{\mathbf{x}} \frac{\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})}{\pi_{\mathcal{Y}}(\mathbf{y})} = \operatorname{argmax}_{\mathbf{x}} \pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})$$

By taking the negative logarithm:

$$\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}} -\ln\left(\pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y})\right) = \operatorname{argmin}_{\mathbf{x}} -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})\right) = \operatorname{argmin}_{\mathbf{x}} \underbrace{-\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right)}_{\text{max. log-likelihood}} \underbrace{-\ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}_{\text{prior}}$$

Bayesian formulation: priors, score functions

$$\mathbf{x}^* \in \text{argmax}_{\mathbf{x}}\ \pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y}) = \text{argmax}_{\mathbf{x}}\ \frac{\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})}{\pi_{\mathcal{Y}}(\mathbf{y})} = \text{argmax}_{\mathbf{x}}\ \pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})$$

By taking the negative logarithm:

$$\mathbf{x}^* \in \text{argmin}_{\mathbf{x}}\ -\ln\left(\pi_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}\,|\,\mathbf{y})\right) = \text{argmin}_{\mathbf{x}}\ -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\pi_{\mathcal{X}}(\mathbf{x})\right) = \text{argmin}_{\mathbf{x}}\ \underbrace{-\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right)}_{\text{max. log-likelihood}}\ \underbrace{-\ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}_{\text{prior}}$$

Example: $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}) = \pi_E(\mathbf{y}-\mathbf{x}), \quad \pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$

$$\mathbf{x}^* \in \text{argmin}_{\mathbf{x}}\ -\ln\left(\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}}\exp\left(-\frac{|y_i - x_i|^2}{2\sigma_\varepsilon^2}\right)\right) - \ln\left(\frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_x|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_x)^\top\boldsymbol{\Sigma}_x^{-1}(\mathbf{x}-\boldsymbol{\mu}_x)\right)\right)$$

$$= \text{argmin}_{\mathbf{x}}\ \frac{1}{2\sigma_\varepsilon^2}\|\mathbf{y}-\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{x}-\boldsymbol{\mu}_x\|^2_{\boldsymbol{\Sigma}_x^{-1}}$$

+ neglecting constants

—-> towards optimisation problem!

From a statistical to a variational perspective: optimisation

$$\boxed{\operatorname{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right) - \ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}$$

Whenever likelihood + prior functionals belong to a *log-concave* exponential family we end up with ***convex*** optimisation problems for a functional $J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$

From a statistical to a variational perspective: optimisation

$$\boxed{\operatorname{argmin}_{\mathbf{x}} J(\mathbf{x}) := -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right) - \ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}$$

Whenever likelihood + prior functionals belong to a *log-concave* exponential family we end up with ***convex*** optimisation problems for a functional $J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$

We are going to see a very simple algorithm for minimising J based on gradients. Note:

$$\nabla J(\mathbf{x}^*) = -\nabla \ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}^*)\right) - \nabla \ln\left(\pi_{\mathcal{X}}(\mathbf{x}^*)\right) = \mathbf{0}$$

when looking for MAP estimators.

From a statistical to a variational perspective: optimisation

$$\boxed{\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right) - \ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}$$

Whenever likelihood + prior functionals belong to a *log-concave* exponential family we end up with ***convex*** optimisation problems for a functional $J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$

We are going to see a very simple algorithm for minimising J based on gradients. Note:

$$\nabla J(\mathbf{x}^*) = -\nabla \ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}^*)\right) - \nabla \ln\left(\pi_{\mathcal{X}}(\mathbf{x}^*)\right) = \mathbf{0}$$

when looking for MAP estimators.

	Bayesian model	**Optimisation approach**
Noise information	Likelihood	Data-term
A-priori information	Prior	Regularisation
Parameters	Hyperparameters	Regularisation parameters

From a statistical to a variational perspective: optimisation

$$\boxed{\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := -\ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})\right) - \ln\left(\pi_{\mathcal{X}}(\mathbf{x})\right)}$$

Whenever likelihood + prior functionals belong to a *log-concave* exponential family we end up with ***convex*** optimisation problems for a functional $J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$

We are going to see a very simple algorithm for minimising J based on gradients. Note:

$$\nabla J(\mathbf{x}^*) = -\nabla \ln\left(\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}^*)\right) - \nabla \ln\left(\pi_{\mathcal{X}}(\mathbf{x}^*)\right) = \mathbf{0}$$

when looking for MAP estimators.

	Bayesian model	**Optimisation approach**
Noise information	Likelihood	Data-term
A-priori information	Prior	Regularisation
Parameters	Hyperparameters	Regularisation parameters

How to choose good noise and image models?

Data terms

$\mathbf{y} = \text{Noise}(\mathbf{x})$ $\dashrightarrow$ $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x})$ $\xrightarrow[\text{+ constants}]{-\ln}$ $D(\mathbf{y}, \mathbf{x})$

Tailored distance function with observation related to noise assumptions?

Data terms

$\mathbf{y} = \text{Noise}(\mathbf{x})$ $\dashrightarrow$ $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y} \,|\, \mathbf{x})$ $\xrightarrow[\text{+ constants}]{-\ln}$ $D(\mathbf{y}, \mathbf{x})$

Tailored distance function with observation related to noise assumptions?

Gaussian noise:

$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbf{I})$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y} - \mathbf{x}) = \dfrac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \|\mathbf{y} - \mathbf{x}\|_2^2$

Data terms

$\mathbf{y} = \text{Noise}(\mathbf{x})$ $\dashrightarrow$ $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y} \,|\, \mathbf{x})$ $\xrightarrow[\text{+ constants}]{-\ln}$ $D(\mathbf{y}, \mathbf{x})$

Tailored distance function with observation related to noise assumptions?

Gaussian noise:

$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbf{I})$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y} - \mathbf{x}) = \dfrac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \|\mathbf{y} - \mathbf{x}\|_2^2$

Poisson noise:

$\text{Noise}(\mathbf{x}) = \text{Pois}(\mathbf{x} + \boldsymbol{\beta}), \quad \mathbf{x} \in \mathbb{R}^n_{\geq 0}, \boldsymbol{\beta} \in \mathbb{R}^n_{>0}$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y}, \mathbf{x} + \boldsymbol{\beta}) = \text{KL}(\mathbf{y}, \mathbf{x} + \boldsymbol{\beta}) = \sum_{i=1}^{n} x_i + \beta_i - y_i \ln\left(x_i + \beta_i\right)$

Data terms

$\mathbf{y} = \text{Noise}(\mathbf{x})$ $\dashrightarrow$ $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y} \,|\, \mathbf{x})$ $\xrightarrow[\text{+ constants}]{-\ln}$ $D(\mathbf{y}, \mathbf{x})$

Tailored distance function with observation related to noise assumptions?

Gaussian noise:

$\text{Noise}(\mathbf{x}) = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbf{I})$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y} - \mathbf{x}) = \dfrac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \|\mathbf{y} - \mathbf{x}\|_2^2$

Poisson noise:

$\text{Noise}(\mathbf{x}) = \text{Pois}(\mathbf{x} + \boldsymbol{\beta}), \quad \mathbf{x} \in \mathbb{R}^n_{\geq 0}, \boldsymbol{\beta} \in \mathbb{R}^n_{>0}$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y}, \mathbf{x} + \boldsymbol{\beta}) = \text{KL}(\mathbf{y}, \mathbf{x} + \boldsymbol{\beta}) = \sum_{i=1}^{n} x_i + \beta_i - y_i \ln\left(x_i + \beta_i\right)$

Impulsive noise:

$\text{Noise}(\mathbf{x}) = (\mathbf{1} - \mathbf{s}) \odot \mathbf{x} + \mathbf{s} \odot \mathbf{c}$ $\dashrightarrow$ $D(\mathbf{y}, \mathbf{x}) = D(\mathbf{y} - \mathbf{x}) = \dfrac{1}{\tau_{\boldsymbol{\varepsilon}}} \|\mathbf{y} - \mathbf{x}\|_1$

$c_i = \mathcal{B}(1/2), \ s_i = \mathcal{B}(p), p \in [0,1]$

$\approx \mathbf{x} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{L}(\mathbf{0}, \tau_{\boldsymbol{\varepsilon}} \mathbf{I})$

noise "sparsity" (the residual is 0 only in few pixels)

Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[\text{+ constants}]{-\ln} R(\mathbf{x})$$

codify *a-priori* information

Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[+\text{ constants}]{-\ln} R(\mathbf{x})$$

codify *a-priori* information

Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \boldsymbol{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\boldsymbol{\Sigma}_x^{-1}}$$

Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[+\text{ constants}]{-\ln} R(\mathbf{x})$$

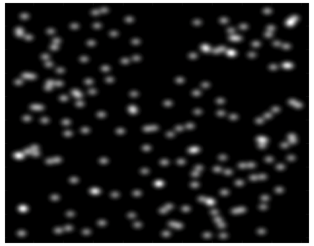
codify *a-priori* information

Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \boldsymbol{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\boldsymbol{\Sigma}_x^{-1}}$$

Sparsity:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\|\mathbf{x}\|_1$$



Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[+\text{ constants}]{-\ln} R(\mathbf{x})$$

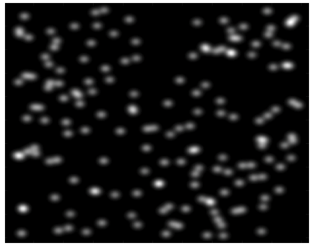
codify *a-priori* information

Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \mathbf{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\mathbf{\Sigma}_x{}^{-1}}$$

Sparsity:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\|\mathbf{x}\|_1$$



$$q_i = \|(\nabla \mathbf{x})_i\|_2$$

Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[\text{+ constants}]{-\ln} R(\mathbf{x})$$

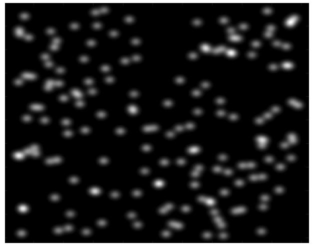
codify *a-priori* information

Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \boldsymbol{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\boldsymbol{\Sigma}_x^{-1}}$$

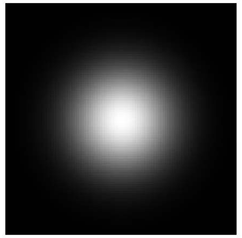
Sparsity:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\|\mathbf{x}\|_1$$



Smoothness:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \pi_{\mathcal{Q}}(\mathbf{q}) = \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{q}}^2\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{2\sigma_{\mathbf{q}}^2}\|\nabla\mathbf{x}\|_2^2$$



$q_i = \|(\nabla \mathbf{x})_i\|_2$

Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[+\text{ constants}]{-\ln} R(\mathbf{x})$$

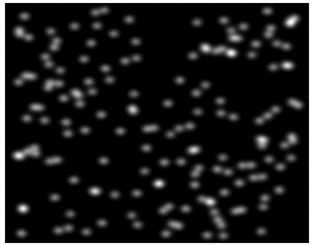
codify *a-priori* information

Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \boldsymbol{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\boldsymbol{\Sigma}_x^{-1}}$$

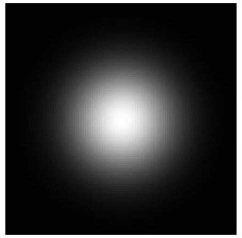
Sparsity:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\|\mathbf{x}\|_1$$



Smoothness:

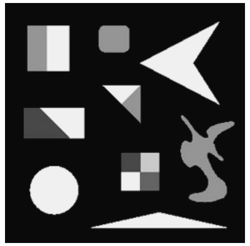
$$\pi_{\mathcal{X}}(\mathbf{x}) = \pi_{\mathcal{Q}}(\mathbf{q}) = \mathcal{N}(\mathbf{0}, \sigma^2_{\mathbf{q}}\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{2\sigma^2_{\mathbf{q}}}\|\nabla\mathbf{x}\|^2_2$$

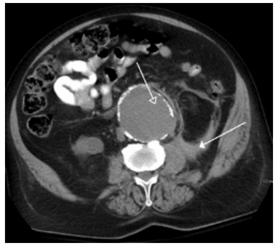


$q_i = \|(\nabla\mathbf{x})_i\|_2$

Piece-wise constancy:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \pi_{\mathcal{Q}}(\mathbf{q}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\mathrm{TV}(\mathbf{x}) = \frac{1}{\tau}\|\nabla\mathbf{x}\|_{2,1}$$





Regularisation terms

$$\pi_{\mathcal{X}}(\mathbf{x}) \xrightarrow[+\text{ constants}]{-\ln} R(\mathbf{x})$$

codify *a-priori* information

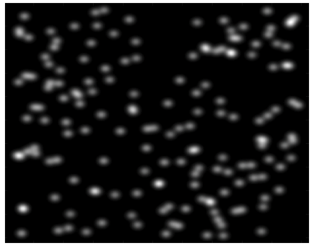
Regularity around the mean:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mu_{\mathbf{x}}, \mathbf{\Sigma}_x) \dashrightarrow R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mu_{\mathbf{x}}\|^2_{\mathbf{\Sigma}_x^{-1}}$$

Tikhonov

Sparsity:

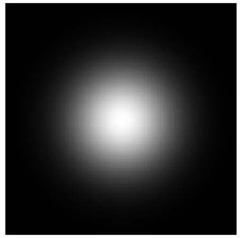
$$\pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\|\mathbf{x}\|_1$$



Sparse reg./
Compressed sensing

Smoothness:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \pi_{\mathcal{Q}}(\mathbf{q}) = \mathcal{N}(\mathbf{0}, \sigma^2_{\mathbf{q}}\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{2\sigma^2_{\mathbf{q}}}\|\nabla\mathbf{x}\|^2_2$$

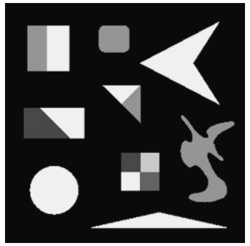


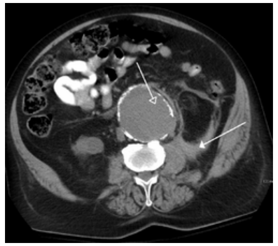
Tikhonov/
Sobolev

$q_i = \|(\nabla\mathbf{x})_i\|_2$

Piece-wise constancy:

$$\pi_{\mathcal{X}}(\mathbf{x}) = \pi_{\mathcal{Q}}(\mathbf{q}) = \mathcal{L}(\mathbf{0}, \tau\mathbf{I}) \dashrightarrow R(\mathbf{x}) = \frac{1}{\tau}\mathrm{TV}(\mathbf{x}) = \frac{1}{\tau}\|\nabla\mathbf{x}\|_{2,1}$$





Total Variation

The regularisation parameter

Example: $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}) = \pi_E(\mathbf{y}-\mathbf{x}), \quad \pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{x}}^2\mathbf{I})$

$$\operatorname{argmin}_{\mathbf{x}} \frac{1}{2\sigma_\varepsilon^2}\|\mathbf{y}-\mathbf{x}\|^2 + \frac{1}{2\sigma_x^2}\|\mathbf{x}\|_2^2 = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{y}-\mathbf{x}\|^2 + \frac{\sigma_\varepsilon^2}{\sigma_x^2}\|\mathbf{x}\|^2 = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{y}-\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|^2$$

Ratio between noise level (*can be estimated*) and image statistical features (*hard to estimate*).

The regularisation parameter

Example: $\pi_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}\,|\,\mathbf{x}) = \pi_E(\mathbf{y}-\mathbf{x}), \quad \pi_{\mathcal{X}}(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{x}}^2\mathbf{I})$

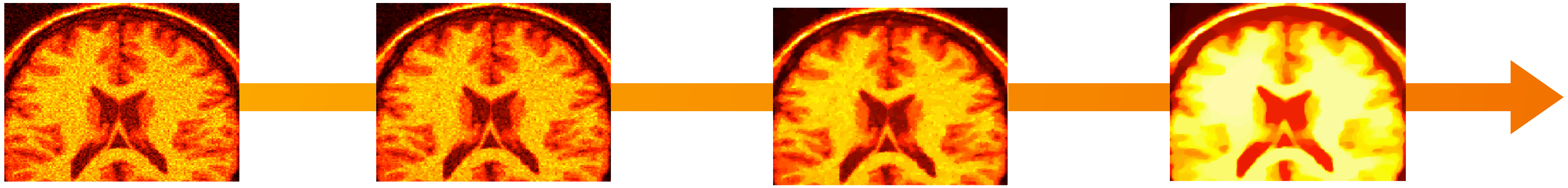
$$\text{argmin}_{\mathbf{x}} \ \frac{1}{2\sigma_\varepsilon^2}\|\mathbf{y}-\mathbf{x}\|^2 + \frac{1}{2\sigma_x^2}\|\mathbf{x}\|_2^2 = \text{argmin}_{\mathbf{x}} \ \|\mathbf{y}-\mathbf{x}\|^2 + \frac{\sigma_\varepsilon^2}{\sigma_x^2}\|\mathbf{x}\|^2 = \text{argmin}_{\mathbf{x}} \ \|\mathbf{y}-\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|^2$$

Ratio between noise level (*can be estimated*) and image statistical features (*hard to estimate*).

More in general, the regularisation parameter $\lambda > 0$ weights data fit against regularisation.

$$\text{argmin}_{\mathbf{x}} \ D(\mathbf{y},\mathbf{x}) + \lambda R(\mathbf{x})$$

- Small λ: low regularisation, trust in the data, noise overfit
- High λ: high regularisation, need to regularise the data, artefacts induced by R



$\lambda \to +\infty$

Examples

$\mathbf{y} = \text{Noise}(\mathbf{x})$

Gaussian noise + piece-wise constant image: reference model for many applications.

$$\text{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \text{TV}(\mathbf{x})$$

Poisson noise + sparse signal: used in microscopy/astronomical imaging

$$\text{argmin}_{\mathbf{x}\geq 0} \text{KL}(\mathbf{y}, \mathbf{x}+\boldsymbol{\beta}) + \lambda\|\mathbf{x}\|_1$$

Gaussian noise + Tikhonov-type regularisation: often used when $\mathbf{A} \neq \mathbf{I}$ for applications

$$\text{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \frac{\lambda}{2}\|\mathbf{L}\mathbf{x}\|_2^2 \qquad \mathbf{L} \in \mathbb{R}^{d\times n}$$

Solving the problem: nods on optimisation

$\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := D(\mathbf{y}, \mathbf{x}) + \lambda R(\mathbf{x})$ variational formulation of the image denoising problem

$J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is proper

$$\text{dom}(J) = \{\mathbf{x} : J(\mathbf{x}) < +\infty\} \neq \emptyset$$

J is L-smooth, i.e. has L-Lipschitz (Gâteaux) gradient:

$$\exists L > 0 : \quad \|\nabla J(\mathbf{x}_1) - \nabla J(\mathbf{x}_2)\|_2 \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|_2$$

J is convex:

$$(\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n), \quad (\forall \alpha \in [0,1]) : \quad J(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha J(\mathbf{x}_1) + (1-\alpha)J(\mathbf{x}_2)$$

Solving the problem: nods on optimisation

$\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := D(\mathbf{y}, \mathbf{x}) + \lambda R(\mathbf{x})$ variational formulation of the image denoising problem

$J : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is proper

$$\text{dom}(J) = \{\mathbf{x} : J(\mathbf{x}) < +\infty\} \neq \varnothing$$

J is L-smooth, i.e. has L-Lipschitz (Gâteaux) gradient:

$$\exists L > 0 : \quad \|\nabla J(\mathbf{x}_1) - \nabla J(\mathbf{x}_2)\|_2 \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|_2$$

$$\Longleftrightarrow$$

$$J(\mathbf{x}_1) \leq J(\mathbf{x}_2) + \langle \nabla J(\mathbf{x}_2), \mathbf{x}_2 - \mathbf{x}_1 \rangle + \frac{L}{2}\|\mathbf{x}_2 - \mathbf{x}_1\|_2^2$$

under convexity

J is convex:

$$(\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n), \quad (\forall \alpha \in [0,1]) : \quad J(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha J(\mathbf{x}_1) + (1-\alpha)J(\mathbf{x}_2)$$

Solving the problem: nods on optimisation

$\text{argmin}_{\mathbf{x}} \; J(\mathbf{x}) := D(\mathbf{y}, \mathbf{x}) + \lambda R(\mathbf{x})$

J is proper, convex, L-smooth and coercive. $\lim_{\|\mathbf{x}\| \to +\infty} J(\mathbf{x}) = +\infty$

Theorem

There exists a minimiser for J. All local minimisers are global minimisers.

For all $\mathbf{x}^* \in \text{Argmin}_{\mathbf{x}} \; J(\mathbf{x})$, there holds $\nabla J(\mathbf{x}^*) = \mathbf{0}$.

Solving the problem: nods on optimisation

$\operatorname{argmin}_{\mathbf{x}} J(\mathbf{x}) := D(\mathbf{y}, \mathbf{x}) + \lambda R(\mathbf{x})$

J is proper, convex, L-smooth and coercive. $\lim_{\|\mathbf{x}\| \to +\infty} J(\mathbf{x}) = +\infty$

Theorem

There exists a minimiser for J. All local minimisers are global minimisers.

For all $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x}} J(\mathbf{x})$, there holds $\nabla J(\mathbf{x}^*) = \mathbf{0}$.

Algorithm (gradient descent): for $\mathbf{x}_0 \in \operatorname{dom}(J)$, $\tau \in \left(0, \frac{2}{L}\right)$, $k \geq 0$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau \nabla J(\mathbf{x}_k)$$

Theorem

There holds $\mathbf{x}_k \to \mathbf{x}^*$ and for the function values: $J(\mathbf{x}_k) - J(\mathbf{x}^*) \leq \dfrac{\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{2\tau k}$.

Example on Tikhonov regularisation

$$\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{L}\mathbf{x}\|_2^2, \qquad \mathbf{L} \in \mathbb{R}^{d\times n}$$

Examples: $\mathbf{L} \in \{\mathbf{I}, \nabla, \nabla^2\}$

Remark: the problem is quadratic, it can be solved by looking at the optimality condition:

$$(\mathbf{x}^* - \mathbf{y}) + \lambda \mathbf{L}^T\mathbf{L}\mathbf{x}^* = \mathbf{0} \quad \Rightarrow \quad (\mathbf{I} + \lambda\mathbf{L}^T\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

and solving the linear system, e.g., using DFT. Also, faster iterative methods exploiting further regularity (strong convexity, C^2) can be employed.

Hansen, Nagy, O'leary, '06, Nesterov, '83

Example on Tikhonov regularisation

$$\text{argmin}_{\mathbf{x}}\ J(\mathbf{x}) := \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{L}\mathbf{x}\|_2^2, \qquad \mathbf{L} \in \mathbb{R}^{d\times n}$$

Examples: $\mathbf{L} \in \left\{\mathbf{I}, \nabla, \nabla^2\right\}$

Remark: the problem is quadratic, it can be solved by looking at the optimality condition:

$$(\mathbf{x}^* - \mathbf{y}) + \lambda \mathbf{L}^T\mathbf{L}\mathbf{x}^* = \mathbf{0} \quad \Rightarrow \quad (\mathbf{I} + \lambda \mathbf{L}^T\mathbf{L})\mathbf{x}^* = \mathbf{y}$$

and solving the linear system, e.g., using DFT. Also, faster iterative methods exploiting further regularity (strong convexity, C^2) can be employed.

Hansen, Nagy, O'leary, '06, Nesterov, '83

$$\nabla J(\mathbf{x}) = (\mathbf{x}-\mathbf{y}) + \lambda\mathbf{L}^T\mathbf{L}\mathbf{x}, \quad L = 1 + \lambda\|\mathbf{L}^T\mathbf{L}\|_*, \quad \mathbf{x}_0 \in \mathbb{R}^n, \quad \tau \in (0, 2/L)$$

```
while not converging
```

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau\left((\mathbf{x}_k - \mathbf{y}) + \lambda\mathbf{L}^T\mathbf{L}\mathbf{x}_k\right)$$

```
end
```

Image denoisers and proximal operators

For general (possibly non-smooth) regularisation functionals $R : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$, note that:

$$\mathrm{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 + \lambda R(\mathbf{x}) = \mathrm{prox}_{\lambda R}(\mathbf{y})$$

where $\mathrm{prox}_{\lambda R} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is single-valued if R is convex and multi-valued (multiple minimisers) otherwise.

Image denoisers and proximal operators

For general (possibly non-smooth) regularisation functionals $R : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$, note that:

$$\text{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 + \lambda R(\mathbf{x}) = \text{prox}_{\lambda R}(\mathbf{y})$$

where $\text{prox}_{\lambda R} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is single-valued if R is convex and multi-valued (multiple minimisers) otherwise.

Examples:

$R(\mathbf{x}) = \iota_C(\mathbf{x})$, C is convex and closed. $\text{prox}_{\iota_C}(\mathbf{y}) = P_C(\mathbf{y})$

$R(\mathbf{x}) = \|\mathbf{x}\|_1$ $\text{prox}_{\lambda\|\cdot\|_1}(\mathbf{y}) = \text{ST}(\mathbf{y}; \lambda)$

$R(\mathbf{x}) = \text{TV}(\mathbf{x})$ $\text{prox}_{\lambda\, TV(\cdot)}(\mathbf{y})?$

Non-smooth regularisation functionals

∇R not defined.

Image denoisers and proximal operators

For general (possibly non-smooth) regularisation functionals $R : \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$, note that:

$$\mathrm{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2 + \lambda R(\mathbf{x}) = \mathrm{prox}_{\lambda R}(\mathbf{y})$$

where $\mathrm{prox}_{\lambda R} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is single-valued if R is convex and multi-valued (multiple minimisers) otherwise.

Examples:

$R(\mathbf{x}) = \iota_C(\mathbf{x})$, C is convex and closed. $\quad \mathrm{prox}_{\iota_C}(\mathbf{y}) = P_C(\mathbf{y})$

$R(\mathbf{x}) = \|\mathbf{x}\|_1 \quad \mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{y}) = \mathrm{ST}(\mathbf{y};\lambda)$

$R(\mathbf{x}) = \mathrm{TV}(\mathbf{x}) \quad \mathrm{prox}_{\lambda\, TV(\cdot)}(\mathbf{y})?$

Non-smooth regularisation functionals

∇R not defined.

Proximal operators are widely used as implicit variants of gradients for non-smooth optimisation:

$$\mathbf{x}_{k+1} = \mathrm{prox}_{\tau\lambda R}\left(\mathbf{x}_k - \tau\nabla_{\mathbf{x}} D(\mathbf{y}, \mathbf{A}\mathbf{x}_k)\right)$$

Gradient-descent on data term + denoising

This observation stands at the very basis of Plug & Play approaches where $\cancel{\mathrm{prox}_{\tau\lambda R}} \longrightarrow \mathscr{D}_\varsigma$

Kamilov, Bouman, Buzzard, Wuhlberg, '23

References

J. Kaipio, E. Somersalo, *Statistical and computational inverse problems*, Springer, 2005.

Tony F. Chan, J. Shen, *Image Processing and Analysis: variational, PDE, wavelet and stochastic methods*, SIAM, 2005.

M. Pragliola, L. Calatroni, A. Lanza, F. Sgallari, *On and beyond Total Variation in imaging: the role of space variance*, SIAM Review, 65 (3), (2023).

S. Arridge, P. Maas, O. Öktem, C.B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numerica, 2019

A. Chambolle, T. Pock, *An introduction to continuous optimization for imaging*, Acta Numerica, 2016

A. Beck, *First-order methods in optimization*, Volume 25, MOS-SIAM series on Optimization, 2017.

Thanks for your attention!

luca.calatroni@unige.it

UniGe

MaLGa