Image restoration

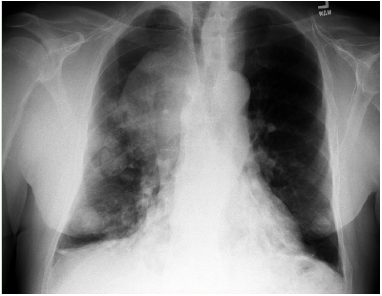
Julie Delon
Univ. Paris Descartes

Cours de Master M2

Première partie I

Image degradations

Image degradations - Noise



Noise Sources : *Shot noise* (discrete nature of light), readout noise, transmission errors, etc.

Image degradations - Blur

Motion blur. *Source : flicker*

Image degradations - Blur

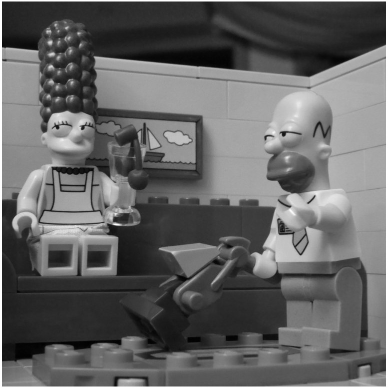
Optical blur. *Wikimedia Commons*

Image degradations - Missing pixels


Source : JD.

Image degradations

Original image



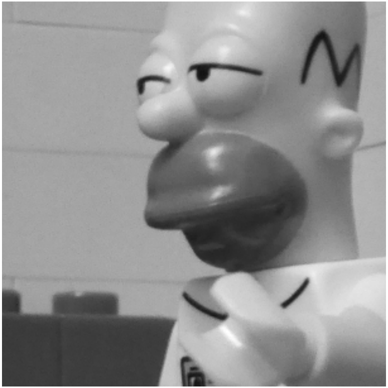
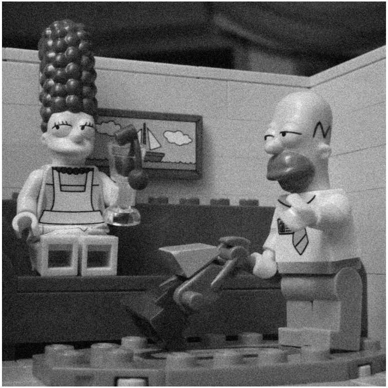


Image degradations

Additive Gaussian noise with $\sigma = 20$ (image range in $[0, 255]$)



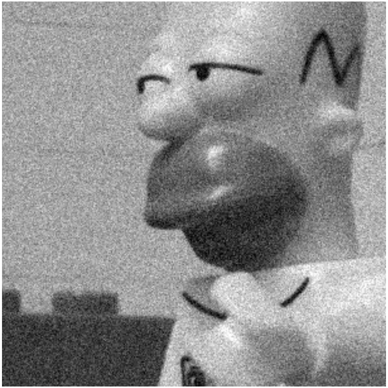
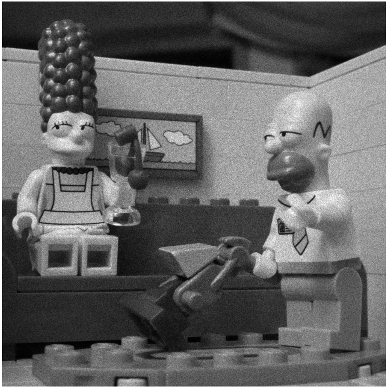


Image degradations

Poisson noise



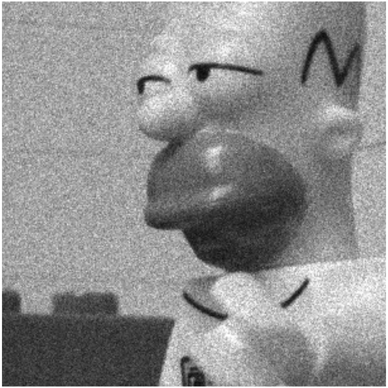


Image degradations

Impulse noise with $p = 0.3$

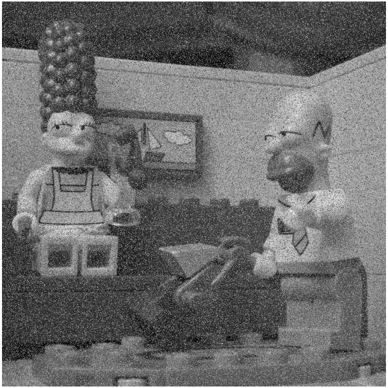


Image degradations

Gaussian blur

Image degradations

Motion blur

Image degradations

Random missing pixels

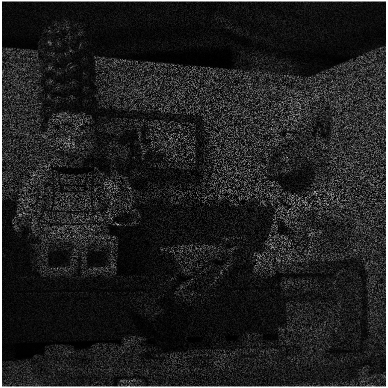
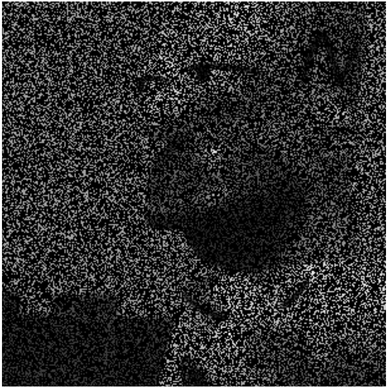
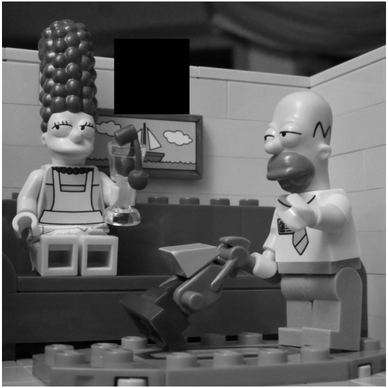
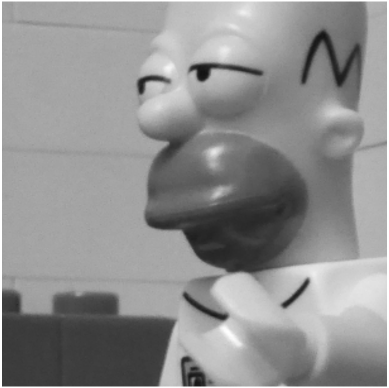



Image degradations

Missing region





Deuxième partie II

Global methods

Degradation model

$$\widetilde{U} = AU + N$$

- A linear degradation, known
- U original image $\in \mathbb{R}^{n \times m}$
- N noise

Degradation model

$$\widetilde{U} = AU + N$$

- A linear degradation, known : blur (deconvolution), missing pixels (inpainting or superresolution)
- U original image $\in \mathbb{R}^{n \times m}$
- N noise

Degradation model

$$\widetilde{U} = AU + N$$

- A linear degradation, known : blur (deconvolution), missing pixels (inpainting or superresolution)
- U original image $\in \mathbb{R}^{n\times m}$: an *a priori* on U is eventually known (regularity, content)
- N noise

Degradation model

$$\widetilde{U} = AU + N$$

- A linear degradation, known : blur (deconvolution), missing pixels (inpainting or superresolution)
- U original image $\in \mathbb{R}^{n \times m}$: an *a priori* on U is eventually known (regularity, content)
- N noise : $N = h(U)\varepsilon$, the function h and the distribution of ε are known.

Direct solution

$$\widehat{U} = \arg\min_{U} \|\widetilde{U} - AU\|^2$$
$$\Rightarrow$$
$$A^*\widetilde{U} - A^*A\widehat{U} = 0$$

where A^* is the adjoint of A.

The problem is **ill-posed** since A^*A is not always one-to-one and its eigenvalues may be small.

Bayesian approach

The operator $D : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ which minimizes the risk

$$\mathbb{E}[\| U - D\widetilde{U}\|^2].$$

is the conditionnal expectation

$$D\widetilde{U} = \mathbb{E}[U/\widetilde{U}].$$

D is **not necessarily linear**, and is quite difficult to estimate in practice, even if one knows the laws of the noise N and the image U (*a priori* law).

Wiener

Assume that $\mathbb{E}[N] = 0$ (the noise is centered) and $\mathbb{E}[U] = 0$ (if not, replace U by $U - \mathbb{E}[U]$).

The **linear** operator D minimizing $\mathbb{E}[\|U - D\widetilde{U}\|^2]$ is the Wiener filter

$$D = \mathbb{E}[U\widetilde{U}^T].\mathbb{E}[\widetilde{U}\widetilde{U}^T]^{-1} = \Sigma_U.A^T(A\Sigma_U A^T + \Sigma_N)^{-1}.$$

Wiener deconvolution

Assume that

- A is a convolution with a discrete filter a
- the noise N is independent of the signal U.

We look for a convolution filter g which minimizes

$$\mathbb{E}[\|g \star \widetilde{U} - U\|^2].$$

Then, in the Fourier domain

$$\widehat{g} = \frac{\widehat{a}^*}{\frac{\mathbb{E}[|\widehat{N}|^2]}{\mathbb{E}[|\widehat{U}|^2]} + |\widehat{a}|^2},$$

where $\hat{a}$ is the discrete fourier transform (DFT) of a. In practice, we do not have access to $\mathbb{E}[|\widehat{U}|^2]$...

Wiener

Wiener

Maximum *a posteriori*

Replace $\mathbb{E}[U/\widetilde{U}]$ by

$$\begin{aligned} \arg\max_U \mathbb{P}[U/\widetilde{U}] &= \arg\max \ \mathbb{P}[\widetilde{U}/U]\ \mathbb{P}[U] \\ &= \arg\min_U -\log \mathbb{P}[\widetilde{U}/U] - \log \mathbb{P}[U] \\ &= \arg\min_U \ \underbrace{G(U,\widetilde{U})}_{\text{data fidelity term}} + \underbrace{F(U)}_{\text{a priori on U}} \ . \end{aligned}$$

Maximum *a posteriori*

Replace $\mathbb{E}[U/\widetilde{U}]$ by

$$\begin{aligned}\arg\max_U \mathbb{P}[U/\widetilde{U}] &= \arg\max\ \mathbb{P}[\widetilde{U}/U]\ \mathbb{P}[U] \\ &= \arg\min_U -\log \mathbb{P}[\widetilde{U}/U] - \log \mathbb{P}[U] \\ &= \arg\min_U \underbrace{G(U,\widetilde{U})}_{\text{data fidelity term}} + \underbrace{F(U)}_{\text{a priori on U}}.\end{aligned}$$

Example

$N \sim \mathcal{N}(0, \sigma^2 Id)$, $p_U \propto e^{-F(u)}$, then

$$\arg\min_U \|\widetilde{U} - AU\|_2^2 + F(U),$$

with $\|\|_2$ the euclidean norm on $\mathbb{R}^{n\times m}$.

Tykhonov regularization

[Tikhonov, Arsenin, 1977]

$$E(u) = \frac{1}{2}\|\tilde{u} - Au\|_2^2 + \lambda \int |\nabla u|^2,$$

where $|\nabla u|$ denotes the norm of ∇u in $\mathbb{R}^2$

Tykhonov regularization

[Tikhonov, Arsenin, 1977]

$$E(u) = \frac{1}{2}\|\tilde{u} - Au\|_2^2 + \lambda \int |\nabla u|^2,$$

where $|\nabla u|$ denotes the norm of ∇u in $\mathbb{R}^2$

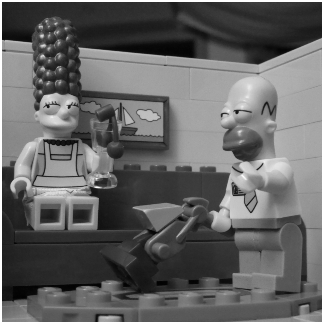
Minimization : convex, differentiable, **gradient descent** with

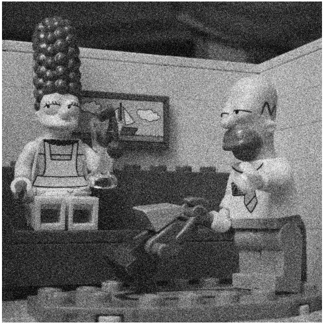
$$\frac{\partial E}{\partial u} = A^*(Au - \tilde{u}) - 2\lambda\Delta u.$$

Or solve $A^*(Au - \tilde{u}) - 2\lambda\Delta u = 0$ using the discrete Fourier transform.

Tykhonov for denoising

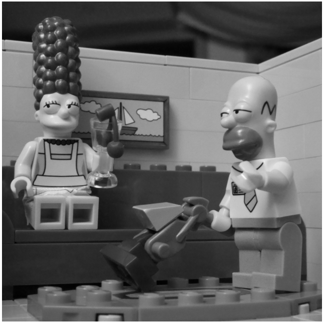
$A = Id$, $N \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.2$ (image range $= [0, 1]$).

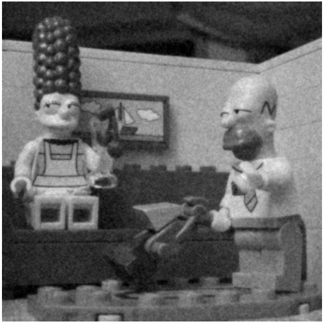




Tykhonov for denoising

Tykhonov regularization with $\lambda = 5$.





TV-L2

$$E(u) = \frac{1}{2}\|\tilde{u} - Au\|_2^2 + \lambda TV(u),$$

with $TV(u) = \sum_{(i,j)} |(\nabla u)_{i,j}|$ the discrete version of the total variation.

Minimization : Convex, but not differentiable.

$$\frac{\partial E}{\partial u} = 2A^*(Au - \tilde{u}) - \lambda \mathrm{div} \frac{\nabla u}{\|\nabla u\|}.$$

Approximation of $TV(u)$ by $\int \sqrt{\|\nabla u\|^2 + \varepsilon^2}$, with a small ε and gradient descent.

TV-L2 for denoising

[Ruding-Osher-Fatemi, 1992] : denoising case, $A = Id$.

$$E(u) = \frac{1}{2}\|\tilde{u} - u\|_2^2 + \lambda TV(u).$$

TV-L2 for denoising

[Ruding-Osher-Fatemi, 1992] : denoising case, $A = Id$.

$$E(u) = \frac{1}{2}\|\tilde{u} - u\|_2^2 + \lambda TV(u).$$

Projection method using duality : [Chambolle 2004].

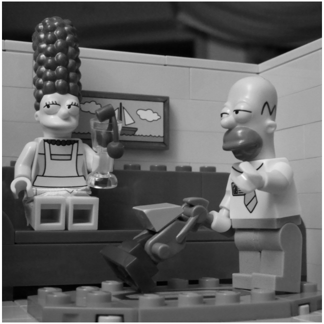
$$\begin{aligned} \hat{u} = \arg\min_u E(u) \quad &\Leftrightarrow \quad 0 \in \hat{u} - \tilde{u} + \lambda\partial TV(\hat{u}) \\ &\Leftrightarrow \quad \ldots \text{ duality arguments} \\ &\Leftrightarrow \quad \frac{\tilde{u} - \hat{u}}{\lambda} = \Pi_\kappa(\frac{\tilde{u}}{\lambda}), \end{aligned}$$

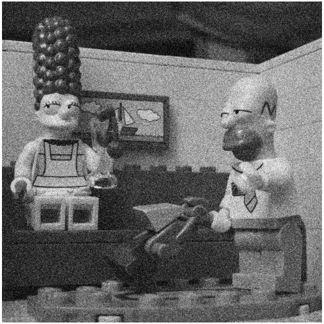
with

$$\kappa = \{\operatorname{div} p \mid \max_{x\in\Omega} |p(x)| \leq 1\}.$$

TV L2 for denoising

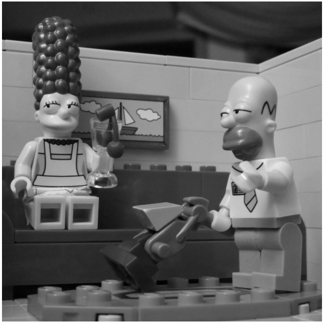
$A = Id$, $N \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.2$ (image range $= [0, 1]$).

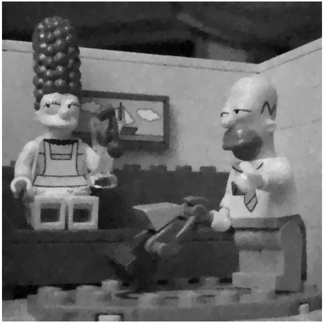




TV L2 for denoising

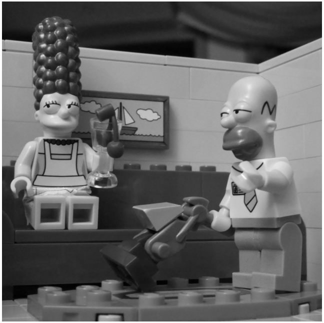
ROF, small λ.

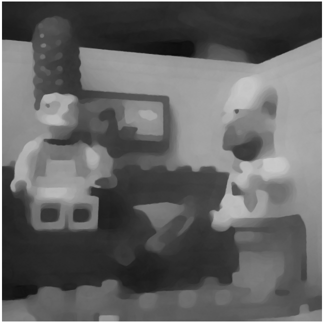




TV L2 for denoising

ROF, large λ.





TV-L1

$$E(u) = \frac{1}{2}\|\tilde{u} - Au\|_1 + \lambda TV(u).$$

- Contrast invariant
- Much more robust to outliers than TV-L2 (impulse noise, salt&pepper noise) .

Minimization ?

TV-L1

Let $X = \mathbb{R}^{n \times m}$, it happens that

$$TV(u) = \max_{p \in X \times X} < \nabla u, p >_X - \iota_\kappa(p),$$

where

$$\iota_\kappa(p) = \begin{cases} 0 & \text{if } p \in \kappa \\ +\infty & \text{if } p \notin \kappa. \end{cases}$$

TV-L1

Let $X = \mathbb{R}^{n\times m}$, it happens that

$$TV(u) = \max_{p\in X\times X} < \nabla u, p >_X - \iota_\kappa(p),$$

where

$$\iota_\kappa(p) = \begin{cases} 0 & \text{if } p \in \kappa \\ +\infty & \text{if } p \notin \kappa. \end{cases}$$

Thus,

$$\arg\min_{u\in X} \quad \frac{1}{2}\|\tilde{u} - Au\|_1 + \lambda TV(u) =$$

$$\arg\min_{u\in X}\max_{p\in X\times X} \underbrace{\frac{1}{2}\|\tilde{u} - Au\|_1}_{G(u)} + \underbrace{\lambda < \nabla u, p >_X}_{<Ku,p>} - \underbrace{\iota_\kappa(p)}_{F^*(p)}.$$

Chambolle-Pock

[Chambolle-Pock, 2012] Primal-dual algorithm for (smooth or non smooth) convex problems of the form :

$$\min_{x \in X} \max_{y \in Y} < Kx, y > + G(x) - F^*(y) \tag{1}$$

with

- X and Y finite-dimensional vector spaces
- $K \ : \ X \rightarrow \ Y$ linear
- $G : X \rightarrow \mathbb{R}^+$ and $F^* : X \rightarrow \mathbb{R}^+$ convex, proper, lower-semicontinuous.

Chambolle-Pock primal-dual algorithm

Algorithm

1. *Initialization : choose* τ, $\sigma > 0$, $(x^0, y^0) \in X \times Y$
2. *Iterations for* $n \geq 0$ *:*

$$\begin{aligned} y^{n+1} &= \underbrace{\mathrm{prox}_{\sigma F^*}}_{\textit{backward step}} \underbrace{(y^n + \sigma K x^n)}_{\textit{forward step}} \\ x^{n+1} &= \mathrm{prox}_{\tau G}(x^n - \tau K^* y^{n+1}). \end{aligned}$$

where prox_f is the proximity operator of a convex, l.s.c function f, defined as

$$\mathrm{prox}_f(x) = \arg\min_z f(z) + \frac{1}{2}\|x - z\|^2.$$

Makes sense if $\mathrm{prox}_{\sigma F}$ and $\mathrm{prox}_{\tau G}$ are easy to compute (closed-forms).

Chambolle-Pock algorithm for denoising with TV-L1

- $G(u) = \frac{1}{2}\|\tilde{u} - u\|_1$
- $F^*(p) = \iota_\kappa(p)$.
- $< Ku, p >= \lambda < \nabla u, p >$ (i.e. $K = \lambda\nabla$ and $K^* = -\lambda\mathrm{div}$)

The proximal operators of G and F^* are

$$(\mathrm{prox}_{\tau G}(u))_{i,j} = \begin{cases} u_{i,j} - \tau & \text{if } u_{i,j} - g_{i,j} > \tau \\ u_{i,j} + \tau & \text{if } u_{i,j} - g_{i,j} < \tau \\ g_{i,j} & \text{if } |u_{i,j} - g_{i,j}| \leq \tau \end{cases}$$

$$(\mathrm{prox}_{\sigma F^*}(p))_{i,j} = (\pi_\kappa(p))_{i,j} = \frac{p_{i,j}}{\max(1, |p_{i,j}|)}.$$

Chambolle-Pock algorithm for denoising with TV-L1

- $G(u) = \frac{1}{2}\|\tilde{u} - u\|_1$
- $F^*(p) = \iota_\kappa(p)$.
- $< Ku, p >= \lambda < \nabla u, p >$ (i.e. $K = \lambda\nabla$ and $K^* = -\lambda \mathrm{div}$)

The proximal operators of G and F^* are

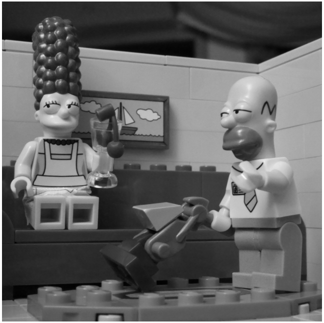
$$(\mathrm{prox}_{\tau G}(u))_{i,j} = \begin{cases} u_{i,j} - \tau & \text{if } u_{i,j} - g_{i,j} > \tau \\ u_{i,j} + \tau & \text{if } u_{i,j} - g_{i,j} < \tau \\ g_{i,j} & \text{if } |u_{i,j} - g_{i,j}| \leq \tau \end{cases}$$

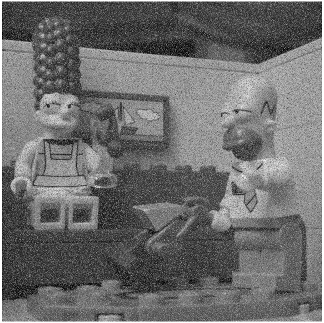
$$(\mathrm{prox}_{\sigma F^*}(p))_{i,j} = (\pi_\kappa(p))_{i,j} = \frac{p_{i,j}}{\max(1, |p_{i,j}|)}.$$

- If $G(u) = \frac{1}{2}\|\tilde{u} - u\|_2^2$, then $(\mathrm{prox}_{\tau G}(u))_{i,j} = \frac{u_{i,j} + \tau \tilde{u}_{i,j}}{1+\tau}$, so the same algorithm can be applied to minimize TV-L2.
- If $G(u) = \frac{1}{2}\|\tilde{u} - Au\|_2^2$ with A a convolution matrix, $\mathrm{prox}_{\tau G}(u)$ can also be computed explicitely.

TV-L1 and TV-L2 for impulse noise

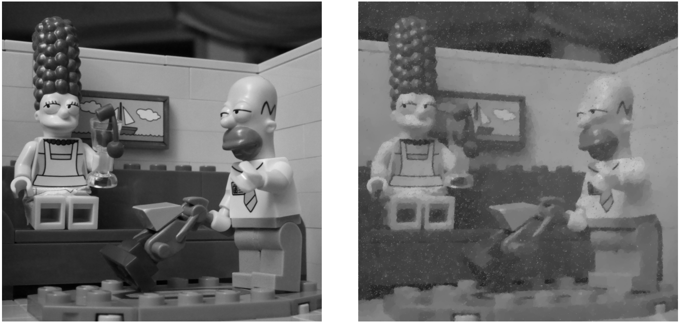
Impulse noise with $p = 0.3$





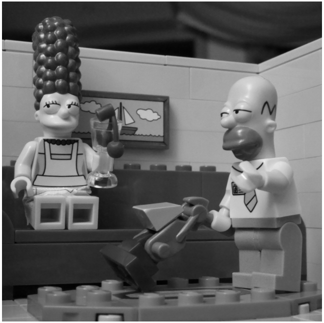
TV-L1 and TV-L2 for impulse noise

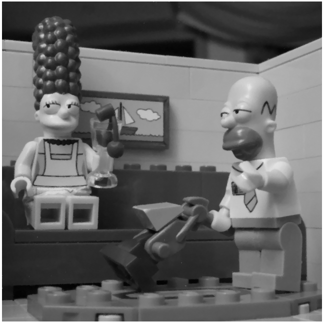
TV-L2



TV-L1 and TV-L2 for impulse noise

TV-L1





TV-L2 for deconvolution

[Chambolle-Pock,2012] with A motion blur.

(a) Original image

(b) Degraded image

(c) Wiener filter

(d) TV-deconvolution

TV for inpainting

$$\min_{u_{|mask}=\tilde{u}_{|mask}} TV(u)$$

TV for inpainting

$$\min_{u_{|mask} = \tilde{u}_{|mask}} TV(u)$$

$$= \min_u TV(u) + \iota_{\mathcal{C}}(u) \text{ with } \mathcal{C} = \{u;\ u_{|mask} = \tilde{u}_{|mask}\}$$

TV for inpainting

$$\min_{u_{|mask}=\tilde{u}_{|mask}} TV(u)$$

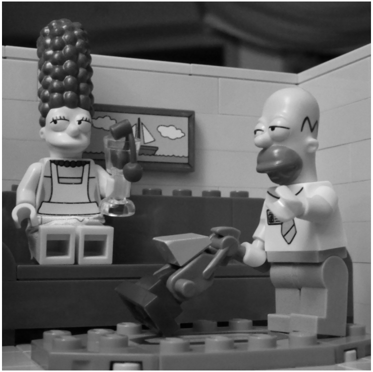
$$= \min_u TV(u) + \iota_{\mathcal{C}}(u) \;\text{ with }\; \mathcal{C} = \{u;\; u_{|mask} = \tilde{u}_{|mask}\}$$

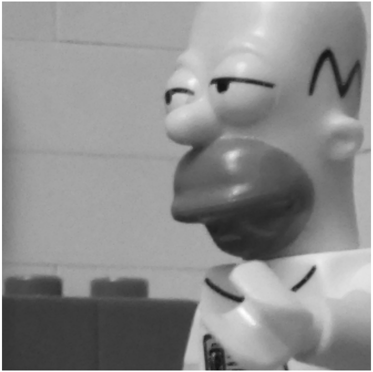

50% random missing pixels

TV for inpainting

$$\min_{u_{|mask} = \tilde{u}_{|mask}} TV(u)$$

$$= \min_u TV(u) + \iota_{\mathcal{C}}(u) \quad \text{with} \quad \mathcal{C} = \{u;\ u_{|mask} = \tilde{u}_{|mask}\}$$



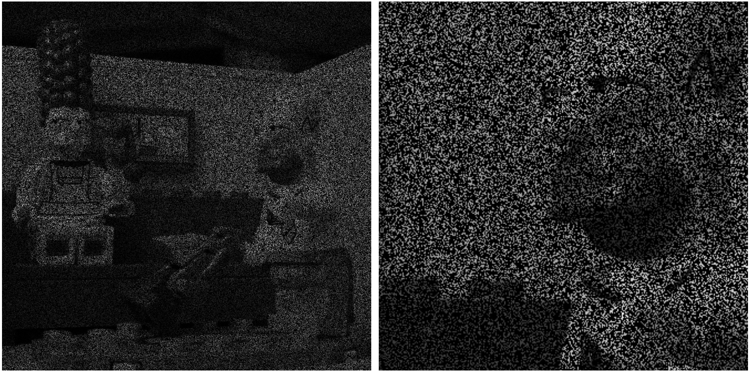


TV inpainting

TV for inpainting

$$\min_{u_{|mask}=\tilde{u}_{|mask}} TV(u)$$

$$= \min_u TV(u) + \iota_{\mathcal{C}}(u) \ \text{ with } \ \mathcal{C} = \{u;\ u_{|mask} = \tilde{u}_{|mask}\}$$

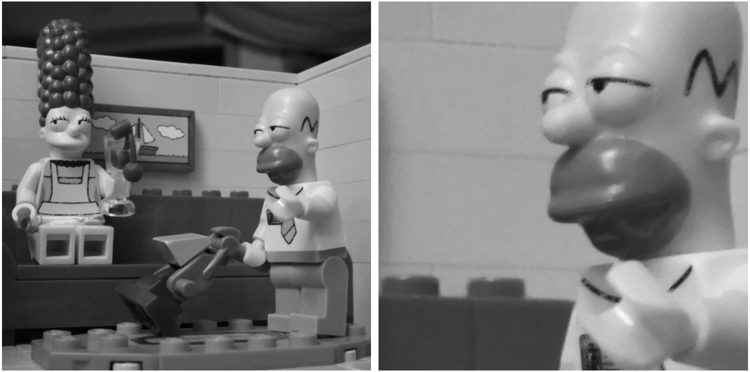


70% random missing pixels

TV for inpainting

$$\min_{u_{|mask}=\tilde{u}_{|mask}} TV(u)$$

$$= \min_u TV(u) + \iota_{\mathcal{C}}(u) \quad \text{with} \quad \mathcal{C} = \{u;\ u_{|mask} = \tilde{u}_{|mask}\}$$

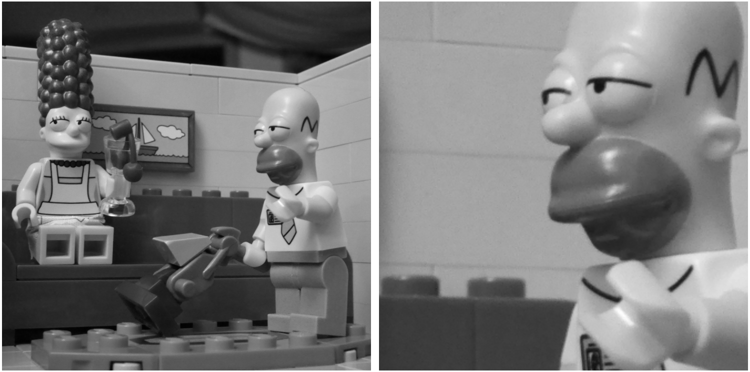


TV inpainting

TV for inpainting

$$\min_{u_{|mask} = \tilde{u}_{|mask}} TV(u)$$

$$= \min_{u} TV(u) + \iota_{\mathcal{C}}(u) \quad \text{with} \quad \mathcal{C} = \{u;\ u_{|mask} = \tilde{u}_{|mask}\}$$



Original image

Troisième partie III

Multi-image restoration

A word on multi-image restoration

Using several shots to increase image quality has become a common challenge in digital photography, movie post-production and remote sensing imaging.

A word on multi-image restoration

1. denoising (*burst denoising*) in low light (avoid motion blur) ;
2. dynamic range increasing (**HDR**) ;
3. panoramas creation ;
4. superresolution, 4k standard
5. color harmonization, style transfert

Multi-image for denoising

Take a burst of images $U_1, U_2, \ldots, U_n$ with short-exposure times and average them after registration.

Law of large numbers

Assume an i.i.d. and centered additive noise of variance σ^2

$$\forall i \in \{1, \dots n\}, \quad \widetilde{U}_i = U_i + B_i.$$

For each pixel $x \in \Omega$,

$$\frac{\widetilde{U}_1(x) + \dots + \widetilde{U}_n(x)}{n} \xrightarrow[n \to +\infty]{}$$

and

$$\mathrm{Var}\left[\frac{\widetilde{U}_1(x) + \dots + \widetilde{U}_n(x)}{n}\right] =$$

Law of large numbers

Assume an i.i.d. and centered additive noise of variance σ^2

$$\forall i \in \{1, \dots n\}, \quad \widetilde{U}_i = U_i + B_i.$$

For each pixel $x \in \Omega$,

$$\frac{\widetilde{U}_1(x) + \dots + \widetilde{U}_n(x)}{n} \xrightarrow[n \to +\infty]{} U(x)$$

and

$$\mathrm{Var}\left[\frac{\widetilde{U}_1(x) + \cdots + \widetilde{U}_n(x)}{n}\right] =$$

Law of large numbers

Assume an i.i.d. and centered additive noise of variance σ^2

$$\forall i \in \{1, \ldots n\}, \quad \widetilde{U}_i = U_i + B_i.$$

For each pixel $x \in \Omega$,

$$\frac{\widetilde{U}_1(x) + \ldots + \widetilde{U}_n(x)}{n} \xrightarrow[n \to +\infty]{} U(x)$$

and

$$\mathrm{Var}\left[\frac{\widetilde{U}_1(x) + \cdots + \widetilde{U}_n(x)}{n}\right] = \frac{\sigma^2}{n}.$$

Law of large numbers

Assume an i.i.d. and centered additive noise of variance σ^2

$$\forall i \in \{1, \dots n\}, \quad \widetilde{U}_i = U_i + B_i.$$

For each pixel $x \in \Omega$,

$$\frac{\widetilde{U}_1(x) + \dots + \widetilde{U}_n(x)}{n} \xrightarrow[n \to +\infty]{} U(x)$$

and

$$\mathrm{Var}\left[\frac{\widetilde{U}_1(x) + \cdots + \widetilde{U}_n(x)}{n}\right] = \frac{\sigma^2}{n}.$$

Fusing n images reduce the noise by a factor $\sqrt{n}$.

A noisy image (Gaussian additive noise, $\sigma = 20$)

Mean of 5 images

Mean of 10 images

Mean of 20 images

Mean of 40 images

Original image

A real-life example of burst denoising

Figure 1.1: From left to right: (a) one long-exposure image (time=0.4 s, ISO=100), one of 16 short-exposure images (time=1/40 s, ISO=1600) and their average after registration. The long exposure image is blurry due to camera motion. (b) The middle short-exposure image is noisy. (c) The third image is about **four times** less noisy, being the result of averaging 16 short-exposure images. From [19].

From [Buadès et al., A note on multi-image denoising, 2009].

Multi-image for denoising : impulse noise

Noise model

$$\forall x \in \Omega, \quad \widetilde{U}_i(x) = (1 - T_i(x)).U(x) + T_i(x).W_i(x),$$

where

- $T_i \sim$ Bernouilli of parameter p ;
- W_i uniform noise on $[0, 255]$;
- T_i, W_i, $i \in \{1 \dots n\}$ are independent.

Multi-image for denoising : impulse noise

Noise model

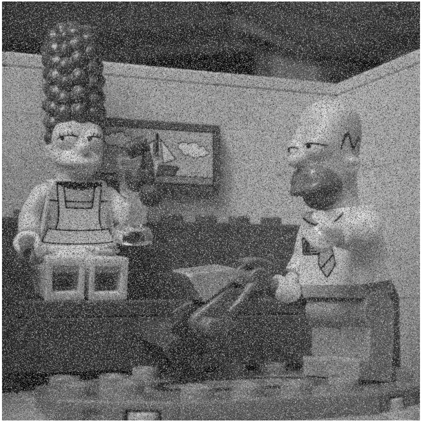
$$\forall x \in \Omega, \quad \widetilde{U}_i(x) = (1 - T_i(x)).U(x) + T_i(x).W_i(x),$$

where

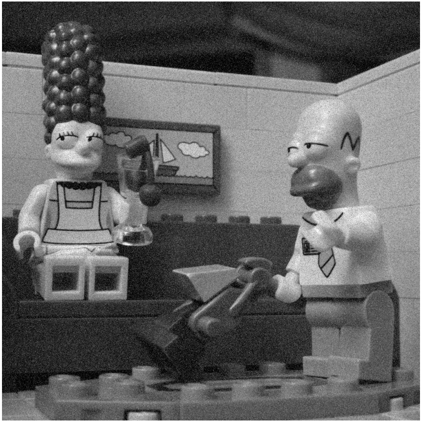
- $T_i \sim$ Bernouilli of parameter p ;
- W_i uniform noise on $[0, 255]$;
- T_i, W_i, $i \in \{1 \dots n\}$ are independent.

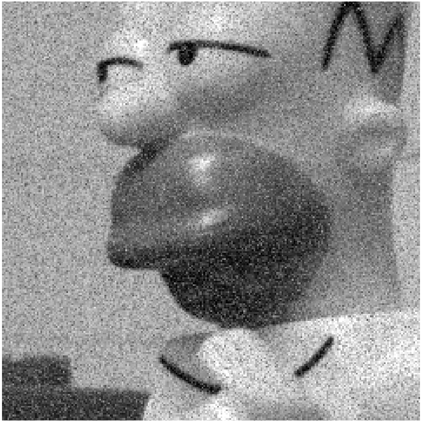
How can we estimate U(x) ?

Impulse noise, one sample, $p = 0.3$

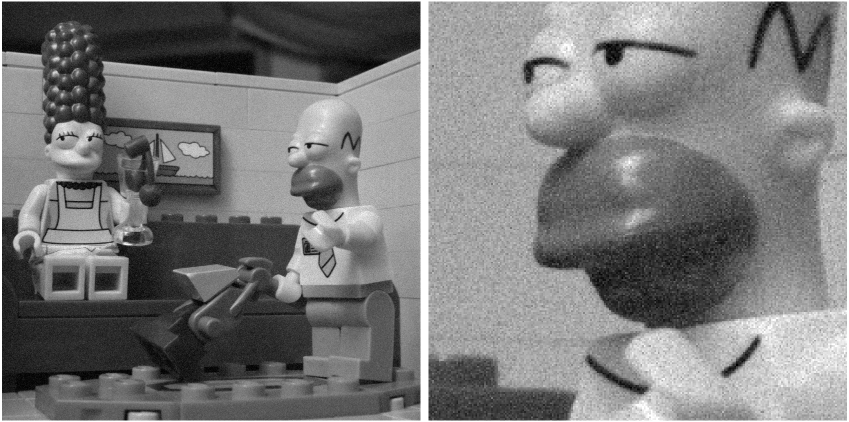


Unbiased estimation from mean of 10 images

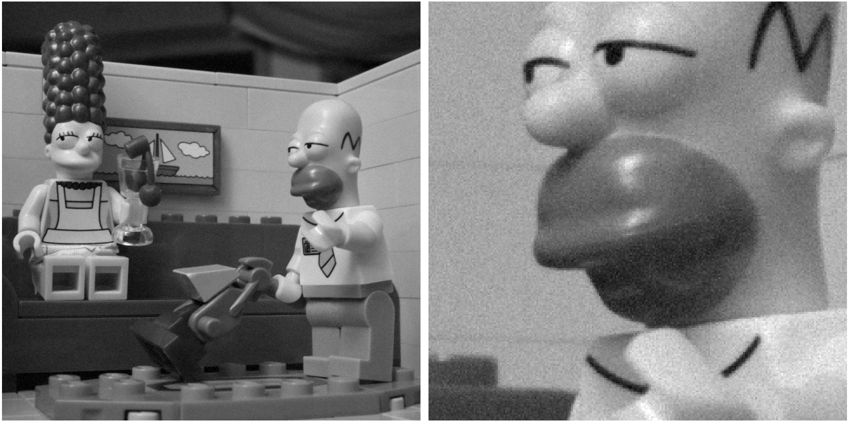




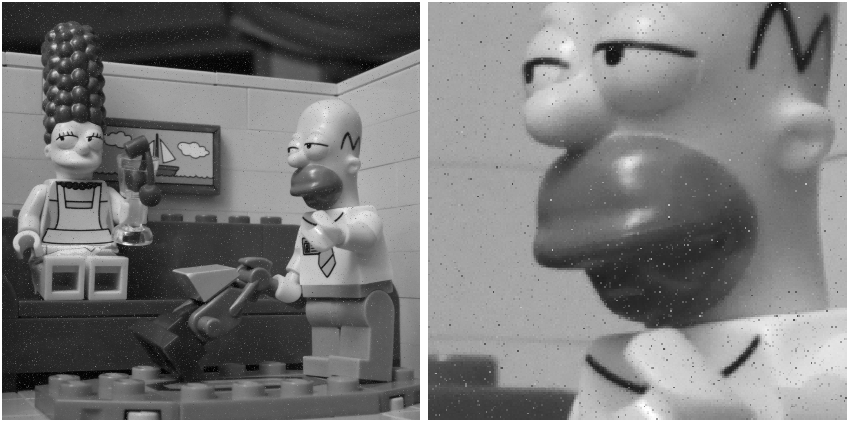
Unbiased estimation from mean of 30 images



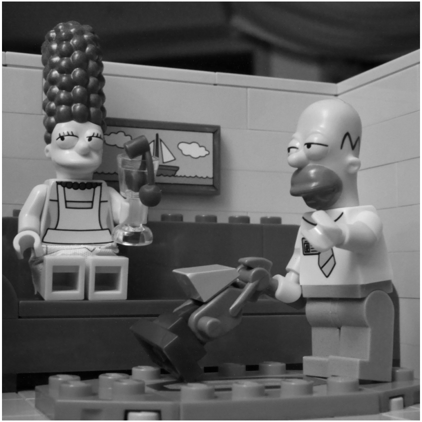
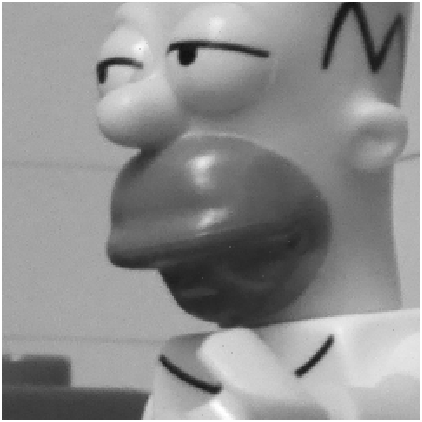
Unbiased estimation from mean of 100 images



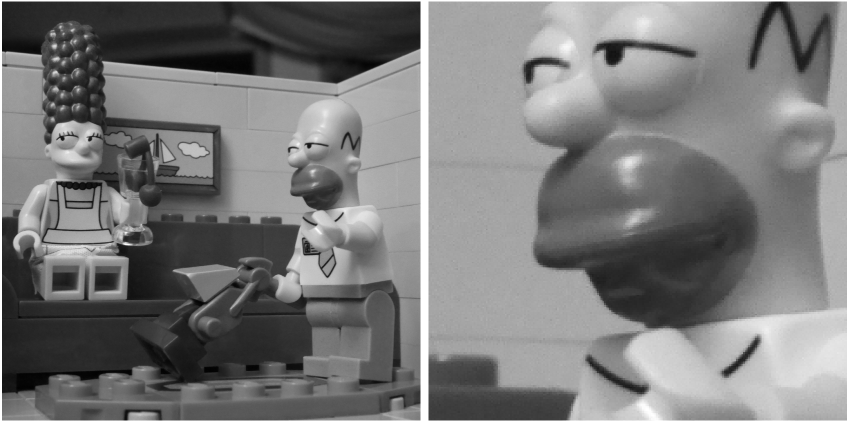
Estimation from median, 10 images



Estimation from median, 30 images

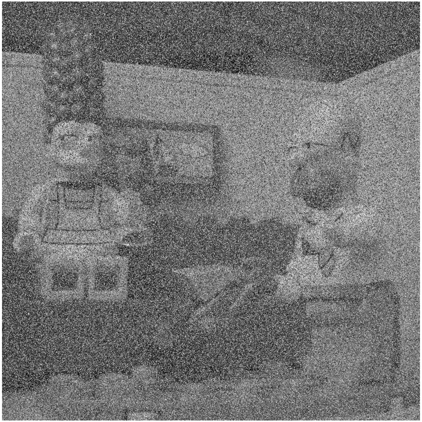



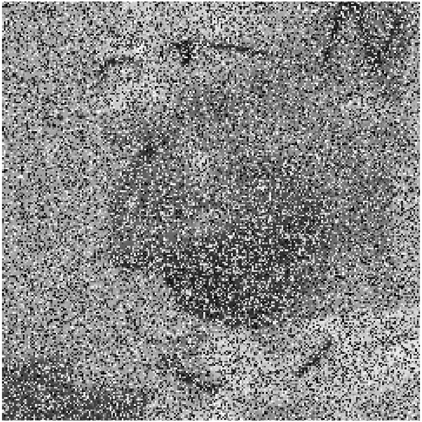
Estimation from median, 100 images



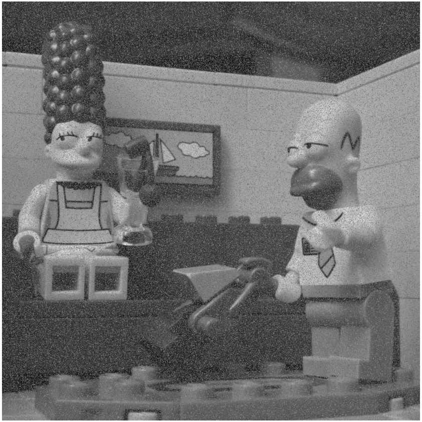
What happens if we increase the noise level p ? Is the median still a good estimator ?

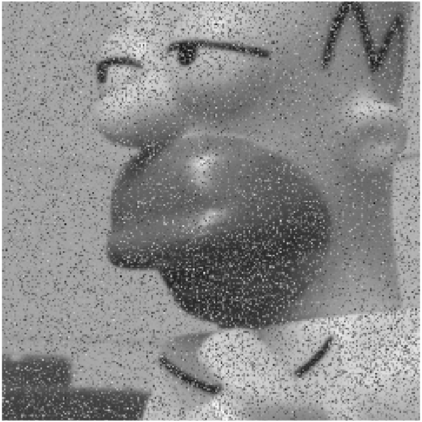
Impulse noise, one sample, $p = 0.6$



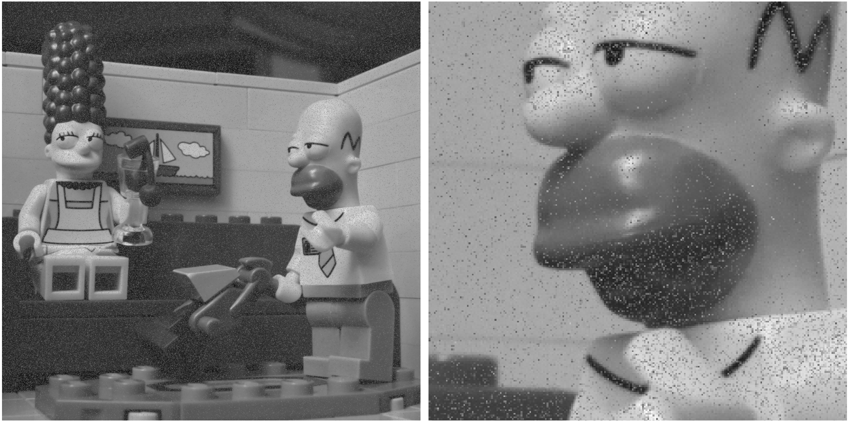


Estimation from median, 10 images

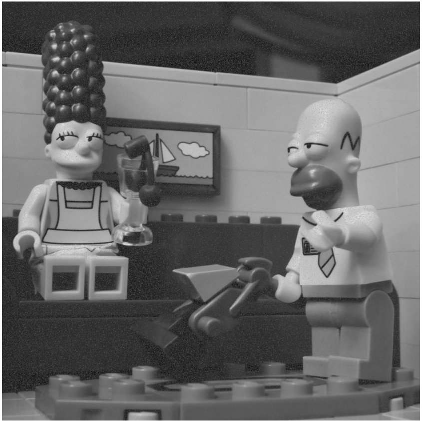


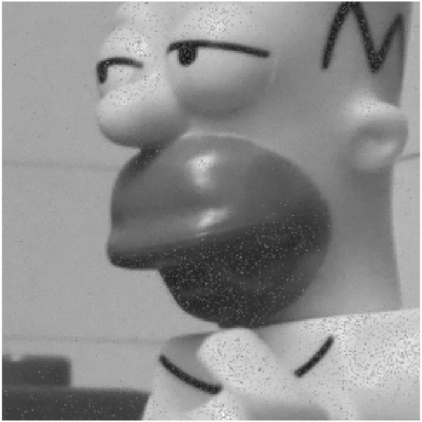


Estimation from median, 30 images



Estimation from median, 100 images





Maximum-likelihood for impulse noise

The variables $U_1(x), \dots, U_n(x)$ follow the law

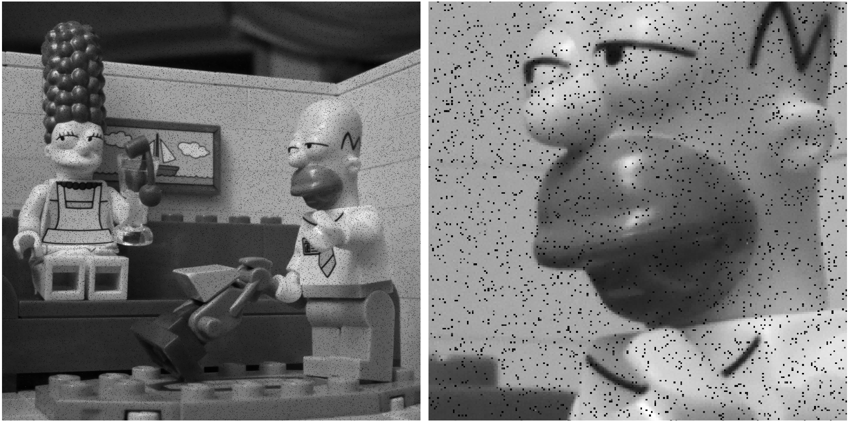
$$(1-p)\delta_{U(x)} + p\mathbf{1}_{[0,255]}.$$

Maximum likelihood

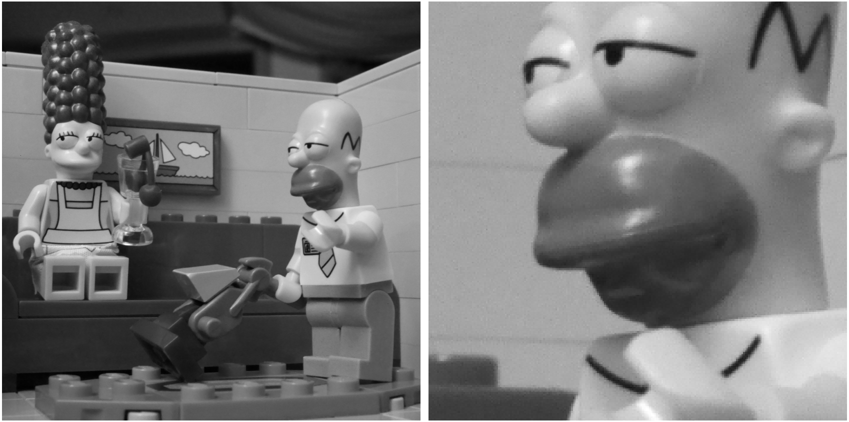
$$\begin{aligned}
\widehat{U(x)} &= \arg\max_{U(x)} \mathbb{P}[\widetilde{U}_1(x), \dots, \widetilde{U}_n(x) | U(x)] \\
&= \arg\max_{U(x)} \sum_{k=1}^{n} \log \mathbb{P}[\widetilde{U}_k(x) | U(x)] \\
&= \arg\max_{U(x)} \sum_{k=1}^{n} \log \left[(1-p)\delta_{U(x)}(U_k(x)) + p * \frac{1}{256} \right] \\
&= \arg\max_{U(x)} h(U(x)),
\end{aligned}$$

with h the histogram of the values $\{\widetilde{U}_1(x), \dots, \widetilde{U}_n(x)\}$.

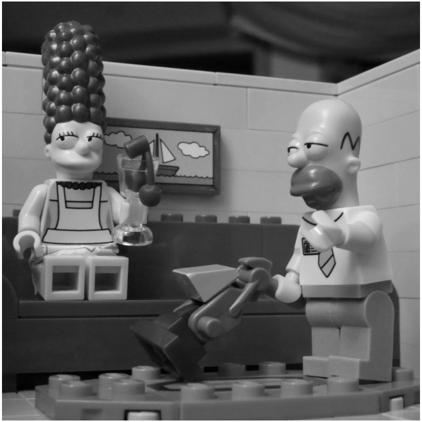
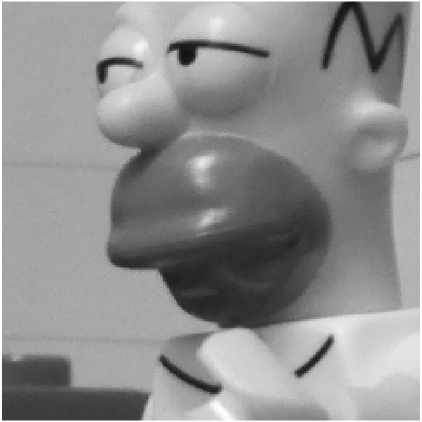
Estimation from histogram, 10 images, $p = 0.6$



Estimation from histogram, 30 images, $p = 0.6$

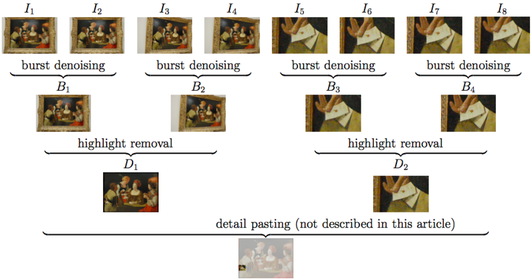


Estimation from histogram, 100 images, $p = 0.6$

Photographing paintings by image fusion

[Haro, Buadès, Morel, 2012]



See `www.ipol.im` for an online demo.

Photographing paintings by image fusion

[Haro, Buadès, Morel, 2012]

See `www.ipol.im` for an online demo.

Photographing paintings by image fusion

[Haro, Buadès, Morel, 2012]

See www.ipol.im for an online demo.

Photographing paintings by image fusion

[Haro, Buadès, Morel, 2012]

See `www.ipol.im` for an online demo.

General case

- Not always possible to take several images : astronomy, medical imaging, etc ;
- Even if we can take several shots, a global registration is not always enough : object motions, large camera motions, etc.

SOLUTION ?

General case

- Not always possible to take several images : astronomy, medical imaging, etc ;
- Even if we can take several shots, a global registration is not always enough : object motions, large camera motions, etc.

SOLUTION ?

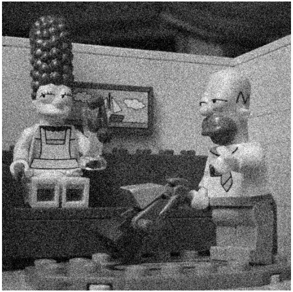
Exploit the patch redundancy of natural images for restoration.

Quatrième partie IV

Patch-based methods

Non local models for image restoration

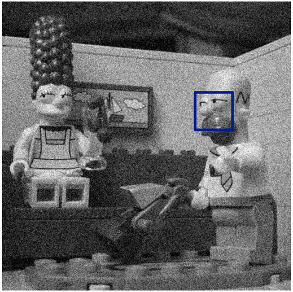
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

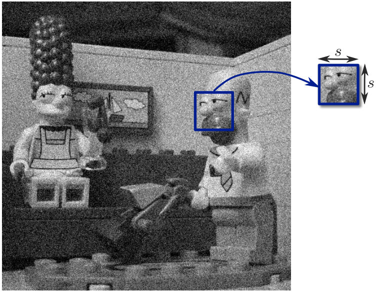
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

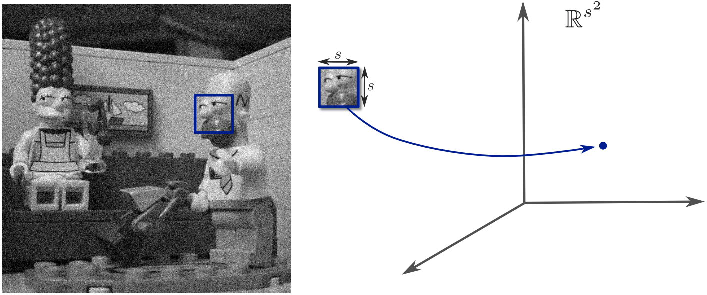
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

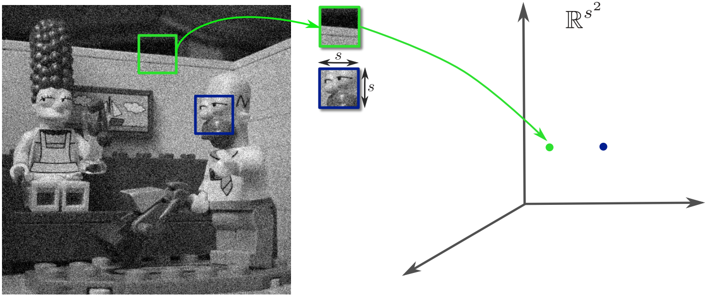
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

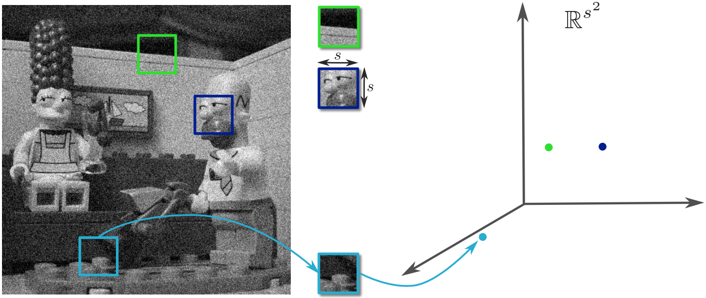
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

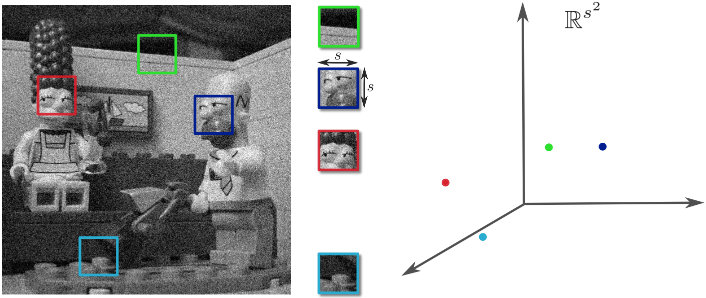
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

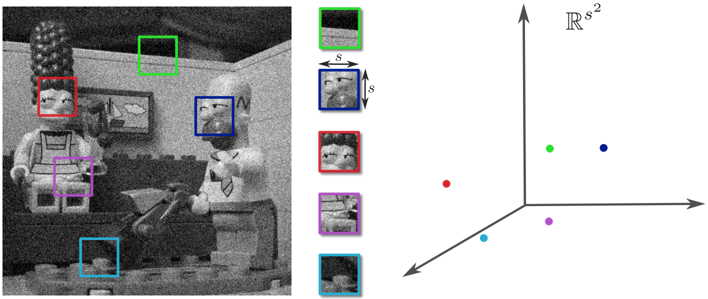
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

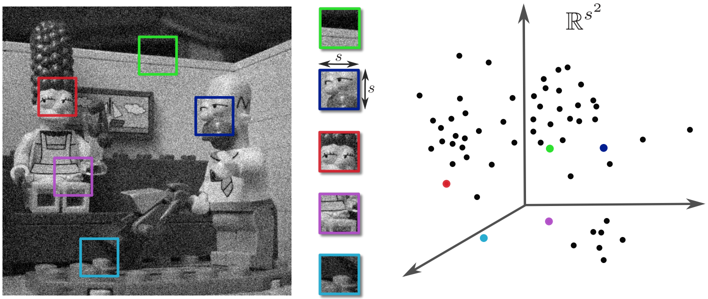
Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Non local models for image restoration

Non local models = all models (either variational, stochastic or geometric) which represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image.



Patches are "the analogs of the phonemes of speech".
Pattern Theory, Desolneux & Mumford [10]

Applications in image restoration and editing

Image editing and synthesis

- texture synthesis Efros-Leung [99]
- image *retargeting* or *reshuffling* Barnes et al. [09]
- style transfer Frigo et al. [16]

Image restoration

- denoising Buades et al. [05], Awate Whitaker [06], Dabov et al. [08], Lebrun et al. [12],
- non gaussien denoising, Poisson, Speckle Deledalle et al. [10], [12], impulse noise Delon Desolneux [13]
- inpainting Wexler et al. [04], Criminisi Perez [04], Newson et al. [14]
- interpolation Yu et al. [12], demosaicing Buades et al. [07]
- HDR Aguerrebere et al. [17], compression
- General inverse problems Peyré [08], PLE Yu et al. [12]

Applications in image restoration and editing

Image editing and synthesis

- texture synthesis Efros-Leung [99]
- image *retargeting* or *reshuffling* Barnes et al. [09]
- style transfer Frigo et al. [16]

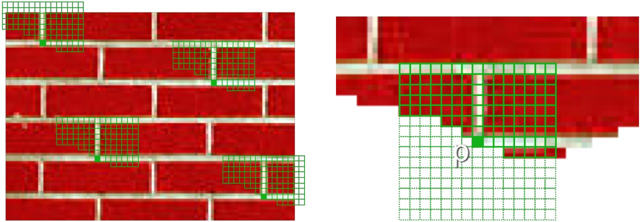
Image restoration

- denoising Buades et al. [05], Awate Whitaker [06], Dabov et al. [08], Lebrun et al. [12],
- non gaussien denoising, Poisson, Speckle Deledalle et al. [10], [12], impulse noise Delon Desolneux [13]
- inpainting Wexler et al. [04], Criminisi Perez [04], Newson et al. [14]
- interpolation Yu et al. [12], demosaicing Buades et al. [07]
- HDR Aguerrebere et al. [17], compression
- General inverse problems Peyré [08], PLE Yu et al. [12]

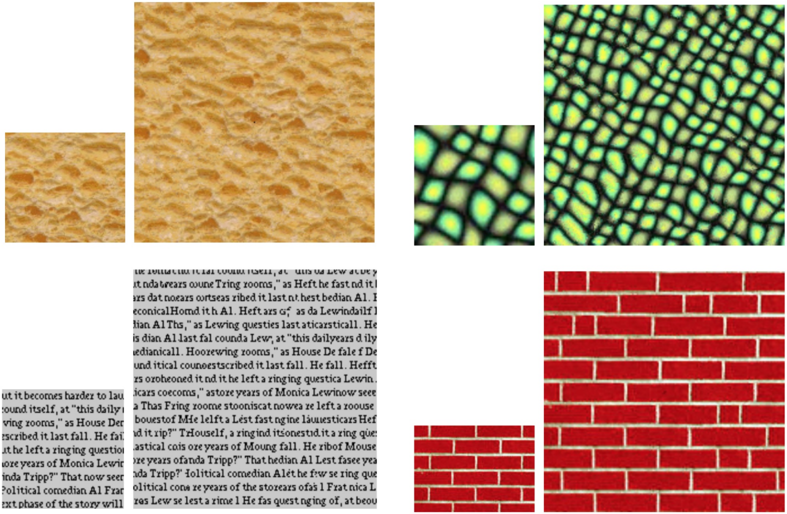
Texture synthesis : Efros-Leung (1999)

Efros - Leung [99]

- Markov random fields models (inspired from Shannon models for text synthesis). We want to estimate $p(u_i | x_i^u)$
- first paper with a patch-based approach : idea to exploit the patch *redundancy* in natural images
- global optimization instead of sequential synthesis Kwatra et al. [03]



Texture synthesis : Efros-Leung (1999)



Try by yourself ! `demo.ipol.im/demo/59/`

Image denoising : NLMeans (2005)

- Observation

$$\tilde{u} = u + n$$

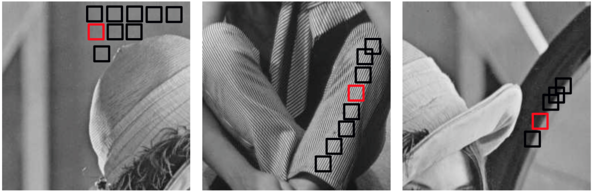
with $n \sim \mathcal{N}(0, \sigma^2)$, we want to reconstruct u.

- **Non Local Means**, Buades, Coll, Morel, [05]

$$\forall i \in \Omega, \quad NLu_i = \frac{\sum_{j \in \Omega} w_{i,j} \tilde{u}_j}{\sum_j w_{i,j}}.$$

with $w_{i,j}$ weights measuring the similarity between patches centered at i and j, typically

$$w_{i,j} = e^{-\|x_i^{\tilde{u}} - x_j^{\tilde{u}}\|_2^2 / 2h^2}.$$



Région uniforme — Région texturée — Contour géométrique

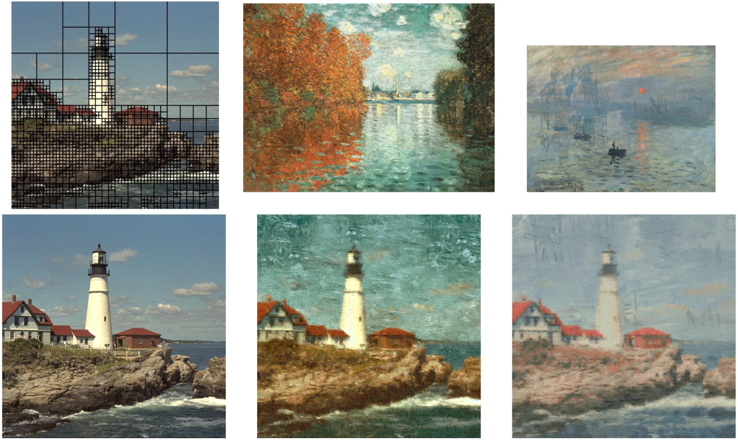
Image denoising : NLMeans (2005) Buades, Coll, Morel, [05]

Noisy Image, $\sigma = 20$

NL-means

Try by yourself ! demo.ipol.im/demo/bcm_non_local_means_denoising/

Patch style transfer (2016)



technicolor

Frigo et al. [16] MRF models

Inverse problems

Model : observation v

$$\underset{\text{observation}}{v} = \underset{\text{acquisition operator}}{A} \; \underset{\text{unknown}}{u} + \underset{\text{noise}}{n}$$

Goal : estimate u from v

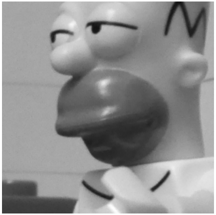
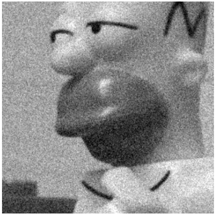
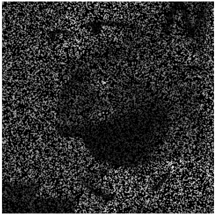


image u



noise

blur



missing data

Inverse problems

Model : for each patch y_i from v

$$\underbrace{y_i}_{\text{observation}} = \underbrace{A_i}_{\text{acquisition operator}} \underbrace{x_i}_{\text{unknown}} + \underbrace{n_i}_{\text{noise}}$$

Goal : estimate all clean patches $x_i \in \mathbb{R}^p$ from the observations $\{y_i\}_i$

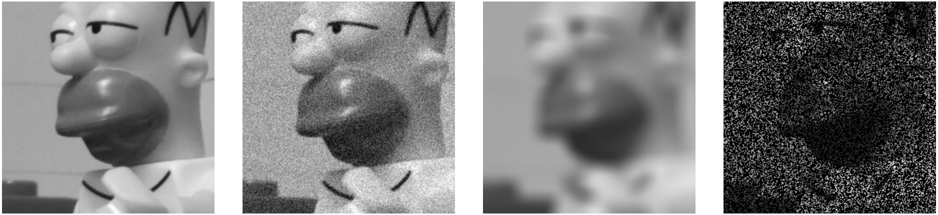


image u   noise   blur   missing data

Restoration strategies

Assuming a prior distribution $p(x)$ for X, the *posterior distribution* is

$$p(x|y) =\propto p(y|x)p(x) \propto e^{-\frac{\|Ax-y\|^2}{2\sigma^2}} p(x).$$

Restoration strategies

Assuming a prior distribution $p(x)$ for X, the *posterior distribution* is

$$p(x|y) =\propto p(y|x)p(x) \propto e^{-\frac{\|Ax-y\|^2}{2\sigma^2}} p(x).$$

Some restoration strategies

- $\widehat{x} = \mathbb{E}[X|Y = y]$ the minimum mean square error (MMSE) estimator

Restoration strategies

Assuming a prior distribution $p(x)$ for X, the *posterior distribution* is

$$p(x|y) = \propto p(y|x)p(x) \propto e^{-\frac{\|Ax-y\|^2}{2\sigma^2}} p(x).$$

Some restoration strategies

- $\widehat{x} = \mathbb{E}[X|Y = y]$ the minimum mean square error (MMSE) estimator
- $\widehat{x} = Dy + \alpha$ s.t. D and α minimize $\mathbb{E}[\|DY + \alpha - X\|^2]$ (linear MMSE or Wiener estimator)

Restoration strategies

Assuming a prior distribution $p(x)$ for X, the *posterior distribution* is

$$p(x|y) =\propto p(y|x)p(x) \propto e^{-\frac{\|Ax-y\|^2}{2\sigma^2}} p(x).$$

Some restoration strategies

- $\widehat{x} = \mathbb{E}[X|Y = y]$ the minimum mean square error (MMSE) estimator
- $\widehat{x} = Dy + \alpha$ s.t. D and α minimize $\mathbb{E}[\|DY + \alpha - X\|^2]$ (linear MMSE or Wiener estimator)
- $\widehat{x} = \arg\max_{x\in\mathbb{R}^p} p(x|y)$ the maximum *a posteriori* (MAP)

Restoration strategies

Assuming a prior distribution $p(x)$ for X, the *posterior distribution* is

$$p(x|y) =\propto p(y|x)p(x) \propto e^{-\frac{\|Ax-y\|^2}{2\sigma^2}} p(x).$$

Some restoration strategies

- $\widehat{x} = \mathbb{E}[X|Y=y]$ the minimum mean square error (MMSE) estimator
- $\widehat{x} = Dy + \alpha$ s.t. D and α minimize $\mathbb{E}[\|DY + \alpha - X\|^2]$ (linear MMSE or Wiener estimator)
- $\widehat{x} = \arg\max_{x \in \mathbb{R}^p} p(x|y)$ the maximum *a posteriori* (MAP)

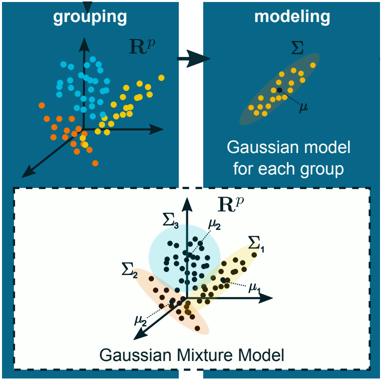
For Gaussian priors, MAP = MMSE = Linear MMSE.

If $X \sim \mathcal{N}(\mu, \Sigma)$ and $N \sim \mathcal{N}(0, \sigma^2 I_p)$ are independent,

$$\begin{aligned}
\hat{x} = \psi(y) &:= \arg\max_x \log p[x|y] \\
&= \arg\min_x \frac{1}{2\sigma^2}(Ax-y)^t(Ax-y) + (x-\mu)^t\Sigma^{-1}(x-\mu) \\
&= \mu + \Sigma A^t(A\Sigma A^t + \sigma^2 I_p)^{-1}(y - A\mu)
\end{aligned}$$

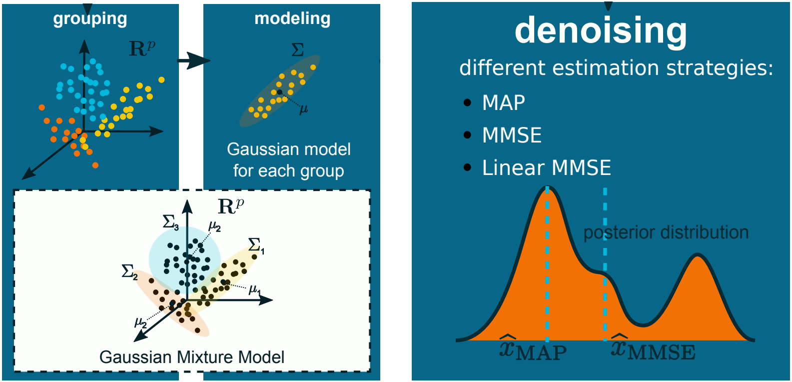
Denoising with Gaussian priors

Bayesian framework with Gaussian or GMM priors on patches, EPLL [11], NL-Bayes [12], PLE [12], S-PLE [13], DA3D [15].

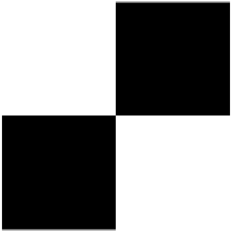


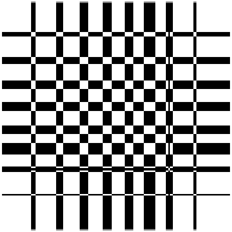
Denoising with Gaussian priors

Bayesian framework with Gaussian or GMM priors on patches, EPLL [11], NL-Bayes [12], PLE [12], S-PLE [13], DA3D [15].

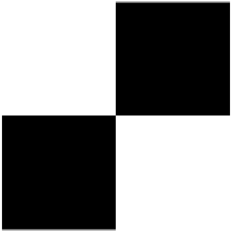


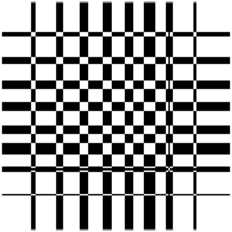
Why Gaussian or GMM priors?

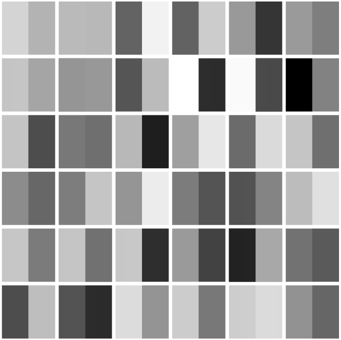




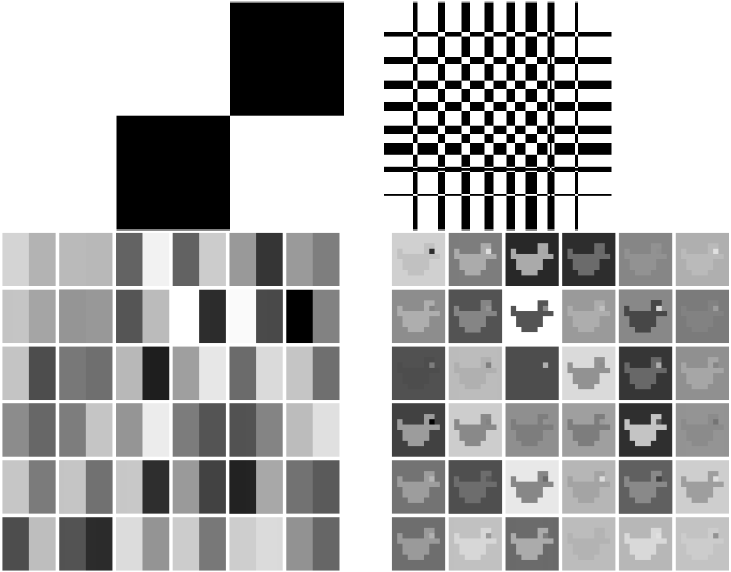
Why Gaussian or GMM priors?



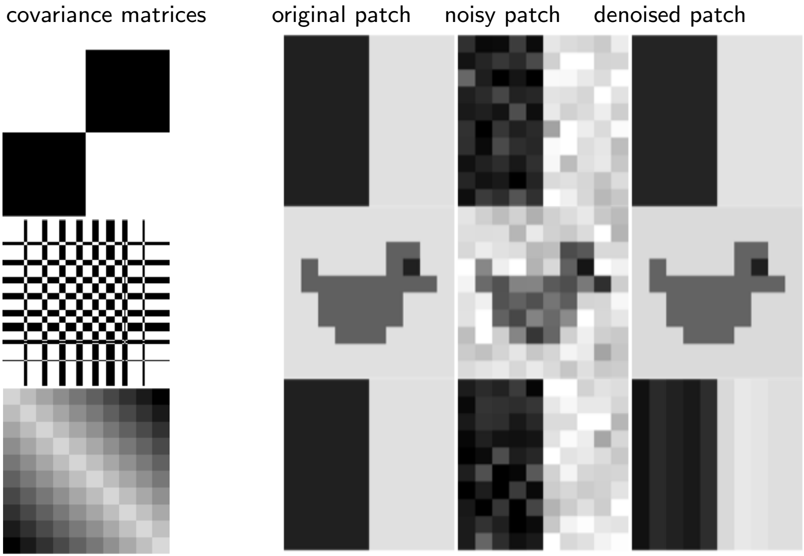




Why Gaussian or GMM priors?

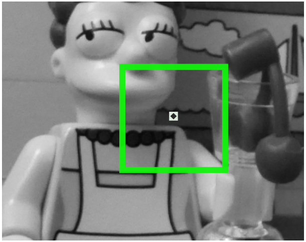


Denoising with the "right" model



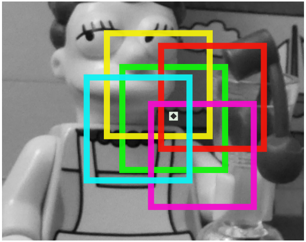
1. How to restore the image from the set of restored patches ?
2. How to estimate (μ, Σ) from the degraded patches $\{y_i\}$?

Reconstruction of u from restored patches



Central value

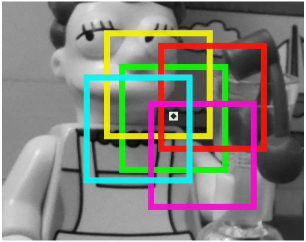
Reconstruction of u from restored patches



Central value

Aggregation of estimators

Reconstruction of u from restored patches



Central value

Aggregation of estimators

Global Optimisation in u, EPLL, Zoran-Weiss [11]

$$\arg\min_{u} \frac{\lambda}{2}\|Au - v\|_2^2 - \sum_{j} \log\ p(x_i^u).$$

How to infer Gaussian or GMM priors ?

- Global or spatially local models, Deledalle et al. [11]
- Local Gaussian models : nearest neighbours, NL-Bayes denoising, Lebrun et al. [13]
- GMM $\sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$
  - estimated on a large external database, EPLL, Zoran-Weiss [11]
  - estimated on the degraded image with a synthetic initialization, PLE, Yu et al. [12] ;
  - estimated on the degraded image, Teodoro et al. [15] ;
  - estimated on the degraded image, based on mixture of PPCA, SURE-PLE, Wang et Morel [13]
  - estimated on the degraded image, based on HDDC, HDMI (High-Dimensional Mixture Model for Image denoising), Houdard et al. [17]
- GGMM (generalized Gaussian mixture models) Deledalle et al. [18]

The curse of dimensionality

Estimation of **Sample Covariance Matrices $\widehat{\Sigma}$** from n samples in high dimension is difficult : estimates tend to be ill-conditionned or even singular...

$\rightarrow$ but $\widehat{\Sigma}$ has to be inverted to compute Wiener estimators...

$\mathbb{R}^{s^2}$

Estimation of **Sample Covariance Matrices** $\widehat{\Sigma}$ from n samples in high dimension is difficult : estimates tend to be ill-conditionned or even singular...

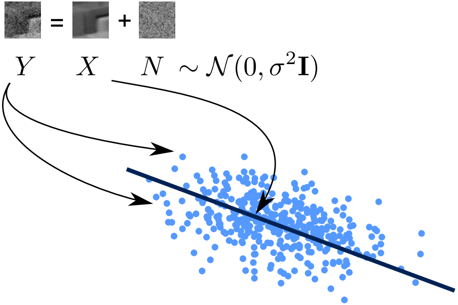
$\rightarrow$ but $\widehat{\Sigma}$ has to be inverted to compute Wiener estimators...

Some workarounds

- use small patches + flat area trick (3×3 or 5×5 in NL-Bayes, Lebrun et al. [13])
- use covariances of fixed lower dimensions, SURE-PLE, Wang et al. [13]
- add regularization ($+|\varepsilon| I_p$) or hyperpriors HBE, Aguerrebere et al. [17]
- infer a specific dimension for each Gaussian HDMI, Houdard et al. [17]

HDMI (Bouveyron, D., Houdard [17])

Assume that patches live in low-dimensional subspaces, specific to their latent groups.



$Z \in \{1 \dots K\}$ latent r.v. indicating the group from which X is generated

$$X_{|Z=k} \sim \mathcal{N}\left(\mu_k, U_k \Lambda_k U_k^t\right),$$

with

- $\mathcal{P}[Z = k] = \pi_k$
- U_k $p \times d_k$ orthonormal,
- $\Lambda_k = \mathrm{diag}(\lambda_{k1}, \dots, \lambda_{kd_k})$,
- $\mu_k \in \mathbb{R}^p$

This model

- is a generalization of the full GMM if $d_k = p$,
- has strong links with the MPPCA model, Tipping, [96].

THE HDMI MODEL

The distribution of Y is also a mixture of Gaussians :

$$p(y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y; \mu_k, \Sigma_k) \text{ with}$$

$$\Sigma_k = U_k \Lambda_k U_k^t + \sigma^2 \mathrm{I}_p.$$

Let $Q_k = [U_k, R_k]$ be a $p \times p$ matrix made of U_k and an orthonormal complementary, then

$$\Delta_k = Q_k^t \Sigma_k Q_k = \left(\begin{array}{cc} \begin{array}{|ccc|} \hline a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd_k} \\ \hline \end{array} & \mathbf{0} \\ \mathbf{0} & \begin{array}{|ccc|} \hline \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \sigma^2 \\ \hline \end{array} \end{array} \right) \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} d_k \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} (p - d_k) \end{array}$$

with $a_{kj} = \lambda_{kj} + \sigma^2$ and $a_{kj} > \sigma^2$, for $j = 1, ..., d_k$ and $k = 1, \ldots, K$.

Denoising with the HDMI model

To denoise the patch y_i, we compute

$$\hat{x}_i = \mathbb{E}[\mathrm{X}|\mathrm{Y} = y_i].$$

Proposition. Under the assumptions of the HDMI model, the conditional expectation $\mathbb{E}[\mathrm{X}|\mathrm{Y} = y_i]$ can be written

$$\mathbb{E}[\mathrm{X}|\mathrm{Y} = y_i] = \sum_{k=1}^{K} \mathcal{P}(Z = k|\mathrm{Y} = y_i)\psi_k(y_i),$$

where

$$\begin{aligned}\psi_k(y) &= \mu_k + (\Sigma_k - \sigma^2 \mathrm{I}_p)\Sigma_k^{-1}(y - \mu_k) \\ &= \mu_k + \tilde{Q}_k \left(\mathrm{I}_p - \sigma^2 \Delta_k^{-1}\right) \tilde{Q}_k^t (y - \mu_k),\end{aligned}$$

with $\tilde{Q}_k = [U_k, 0_{p,p-d_k}]$.

Inference

Before denoising the patches $\{y_1, ..., y_n\}$, the HDMI model has to be inferred from the data :

- estimate model parameters $\theta = \{\pi_k, \mu_k, a_{kj}, \sigma^2, Q_k\}$,
- determine hyper-parameters K and d_k.

Model inference

EM algorithm : maximize *w.r.t.* θ the conditional expectation of the complete log-likelihood :

$$\Psi(\theta, \theta^*) \stackrel{\text{def}}{=} \sum_{k=1}^{K} \sum_{i=1}^{n} t_{ik} \log\left(\pi_k p\left(y_i ; \theta_k\right)\right),$$

where $t_{ik} = \mathbb{E}\left[Z = k | y_i, \theta^*\right]$ and θ^* a given set of parameters.

- E-step estimation of t_{ik} knowing the current parameters
- M-step compute maximum likelihood estimators (MLE)

$$\widehat{\pi}_k = \frac{n_k}{n}, \qquad \widehat{\mu}_k = \frac{1}{n_k} \sum_i t_{ik} y_i, \qquad \widehat{S}_k = \frac{1}{n_k} \sum_i t_{ik}(y_i - \mu_k)(y_i - \mu_k)^T,$$

with $n_k = \sum_i t_{ik}$. Then $\widehat{Q}_k$ is formed by the d_k first eigenvectors of $\widehat{S}_k$ and $\widehat{a}_{kj}$ is the jth eigenvalue of $\widehat{S}_k$.

Hyper-parameters

The hyper-parameters K and $d_1, \ldots, d_K$ cannot be determined by maximizing the log-likelihood since they control the model complexity.

We propose to set K at a given value (for instance $K = 90$) and to choose the intrinsic dimensions d_k :

- using an heuristic that links d_k with the noise variance σ when it is known (**supervised case**) ;
- using a model selection tool in order to select the best σ when unknown (**unsupervised case**).

Estimation of intrinsic dimensions

when σ is known

Heuristic. Given a value of σ^2 and for $k = 1, ..., K$, we estimate the dimension d_k by

$$\widehat{d_k} = \mathrm{argmin}_d \left| \frac{1}{p-d} \sum_{j=d+1}^{p} \widehat{a}_{kj} - \sigma^2 \right| .$$

Estimation of intrinsic dimensions

when σ is unknown

Each value of σ yields a different model, we propose to select the one with the better BIC (Bayesian Information Criterion)

$$\mathrm{BIC}(\mathcal{M}) = \ell(\hat{\theta}) - \frac{\xi(\mathcal{M})}{2}\log(n),$$

where $\xi(\mathcal{M})$ is the complexity of the model.

Role of the intrinsic dimensions d_k

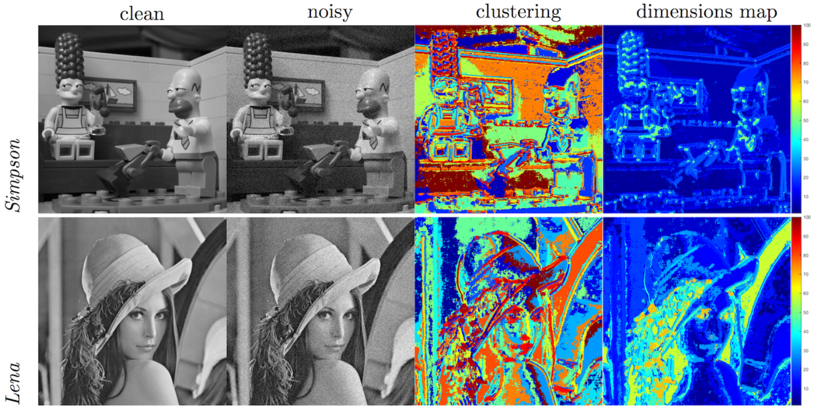
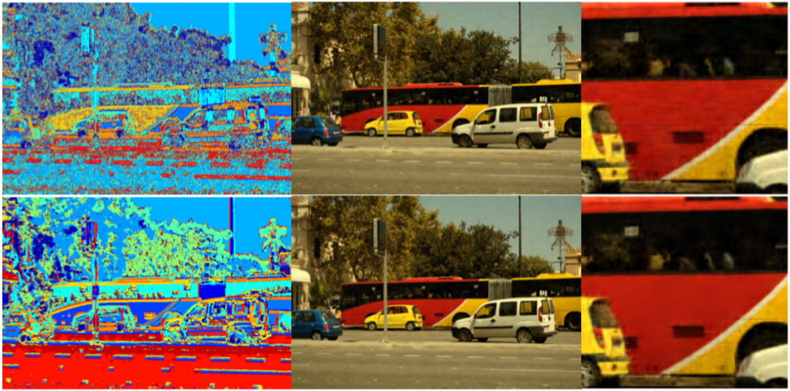


Figure – Map of intrinsic dimensions d_k.

Effect of the dimension reduction inside EM



Original Image

Noisy image $\sigma = 50$

Denoised with BM3D, Foi et al. 2007, psnr = 27.17dB

Numerical experiments

Denoised with FFDNet, Zhang et al. 2018, psnr = 27.58dB

Numerical experiments

Denoised with HDMI_{sup} $K = 90$, psnr $= 27.28$dB

Numerical experiments

Original Image

Noisy image $\sigma = 50$

Denoised with BM3D, Foi et al. 2007, psnr = 26.55.dB

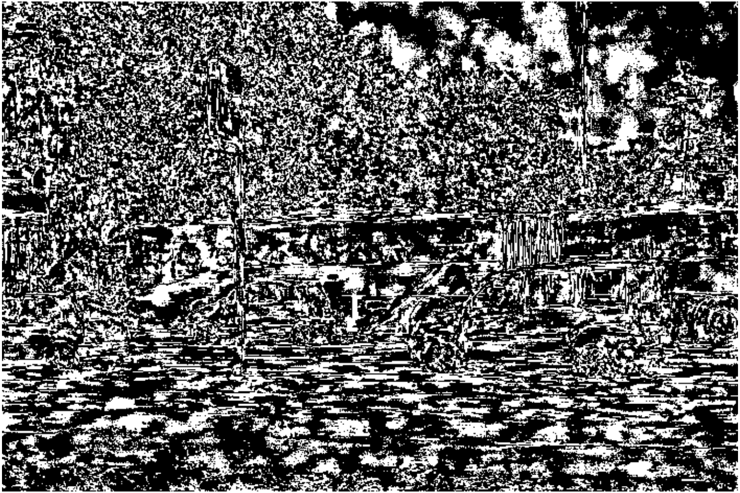
Numerical experiments

Denoised with FFDNet, Zhang et al. 2018, psnr = 27.45dB

Numerical experiments

Denoised with HDMI_{sup} $K = 90$, psnr $= 27.05$dB

PSNR HDMI vs FFDNet



Numerical experiments

Best of both worlds, psnr = 27.86dB

Original Image