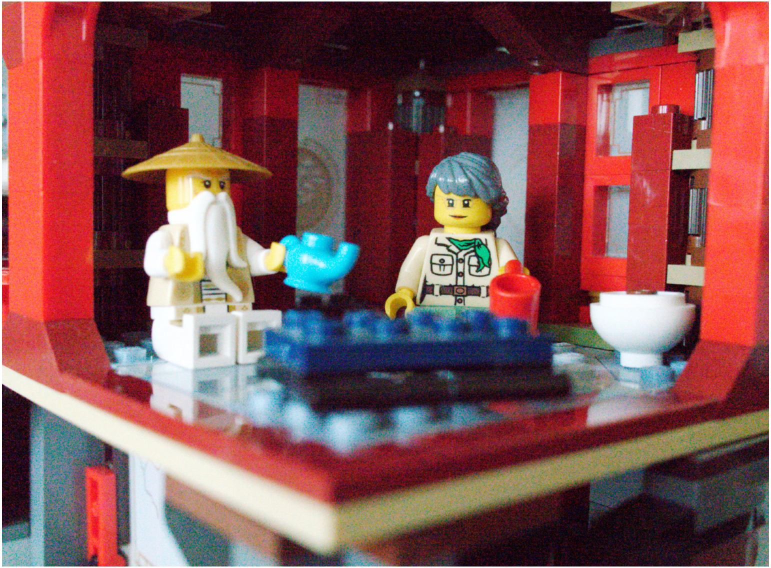
Le débruitage d'images par patchs : point de vue statistique

Antoine Houdard
IMB, Université de Bordeaux

Module Image 2019

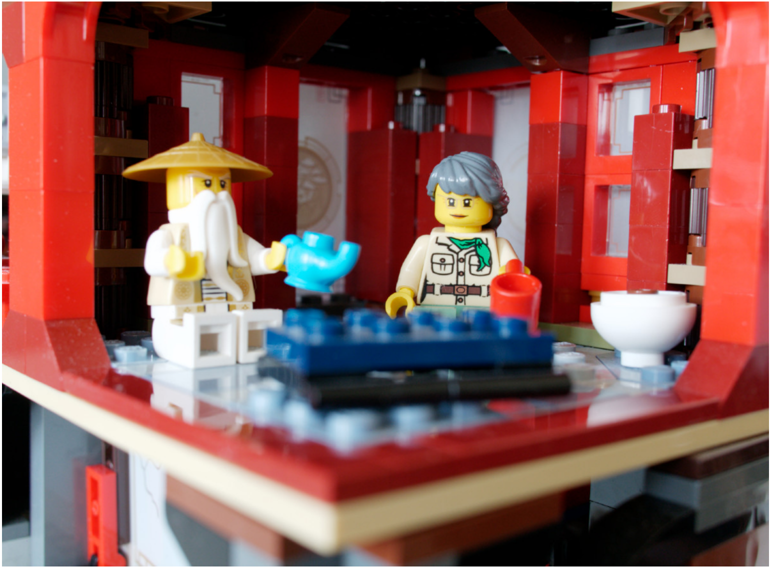
Montpellier, 18 avril

Digital photography: noise in images



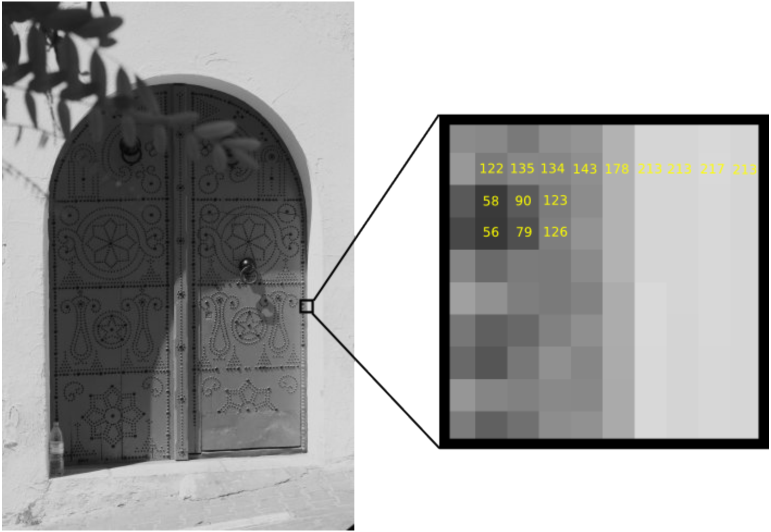
Different ISO settings with constant exposure – 25600 ISO

Digital photography: noise in images



Different ISO settings with constant exposure – 200 ISO

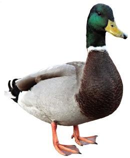
Digital photography



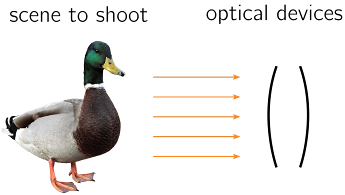
Le monde des pixels

Digital photography: general process

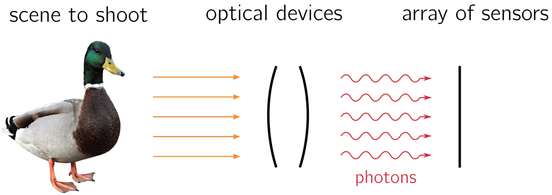
scene to shoot



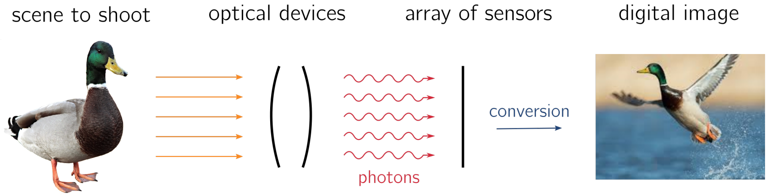
Digital photography: general process



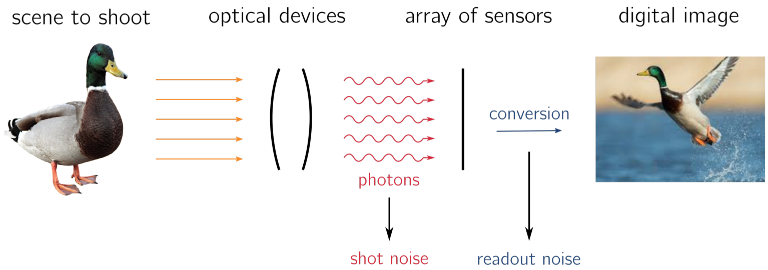
Digital photography: general process



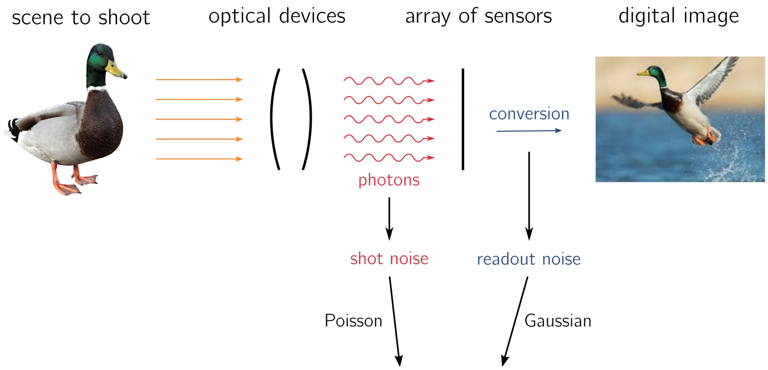
Digital photography: general process



Digital photography: general process



Digital photography: general process



each pixel i is modeled with $$V_i = \alpha P_i + N_i$$

$$u_i = \mathbf{E}[V_i]$$

Noise modeling and denoising problem

Noise modeling - the additive Gaussian white noise model

after variance stabilization

$$V_i = u_i + \mathcal{E}_i$$

observed random variable

underying clean pixel

additive Gaussian white noise $\mathcal{N}(0, \sigma^2)$

one realization v_i is observed

Denoising problem

find an estimate

$$\widehat{u}_i = f(v_i)$$

Need more realizations or prior information

Rappels : variables aléatoires gaussiennes

$V \sim \mathcal{N}(\mu, \sigma^2)$ variable aléatoire gaussienne, sa densité est donnée par

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

- si $V \sim \mathcal{N}(0,1)$ alors $aV + b \sim \mathcal{N}(b, a^2)$
- si $V \sim \mathcal{N}(\mu, \sigma)$ et $W \sim \mathcal{N}(\nu, \tau)$ sont indépendantes
  alors $V + W \sim \mathcal{N}(\mu + \nu, \sigma + \tau)$
- $V \sim \mathcal{N}(\mu, \sigma)$, alors $\mathbf{E}[V] = \mu$ et $\mathrm{var}[V] = \sigma^2$

$\rightarrow$ dans notre cas $V_i \sim \mathcal{N}(u_i, \sigma^2)$

Rappels : variables aléatoires gaussiennes

Loi des grands nombres
$V_1, \ldots, V_n$ variables aléatoires variables indépendantes et identiquement distribuées (iid) de moyenne μ et variance σ^2 alors

$$\mathbf{E}\left[\frac{V_1 + \cdots + V_n}{n}\right] = \mu$$

et

$$\mathrm{var}\left[\frac{V_1 + \cdots + V_n}{n}\right] = \frac{\sigma^2}{n}$$

$\rightarrow$ en moyennant n pixels, on réduit la variance de n !

Moyenner pour débruiter !

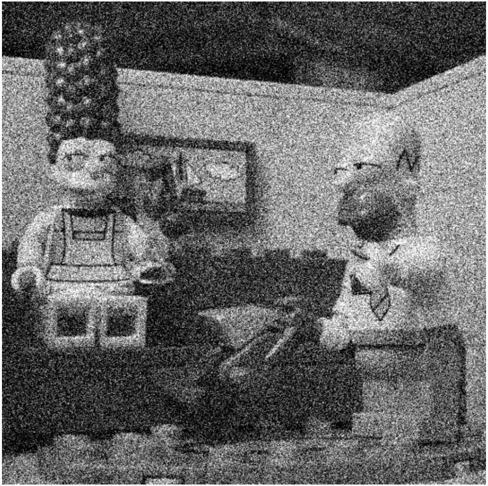
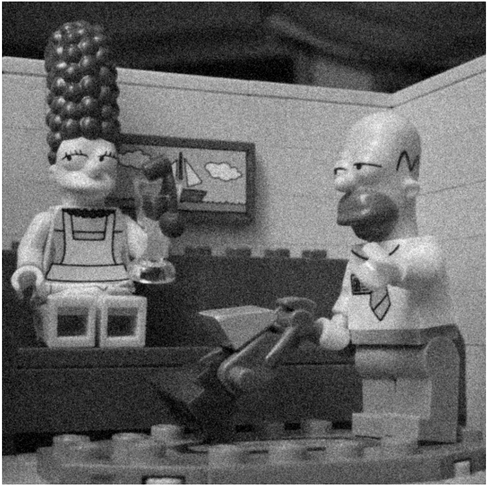


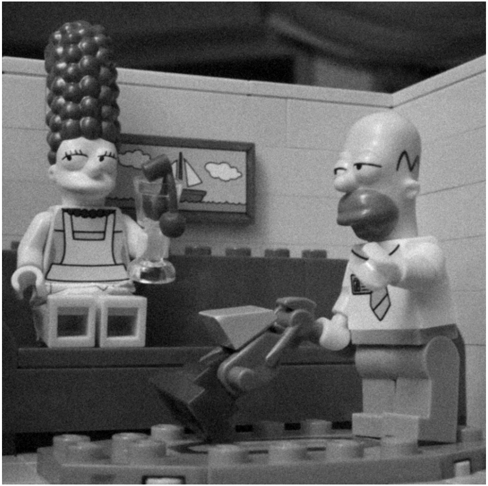
image bruitée à 20% *i.e.* $\sigma^2 = 0.2$ pour une image à valeur dans $[0, 1]$)

Moyenner pour débruiter !



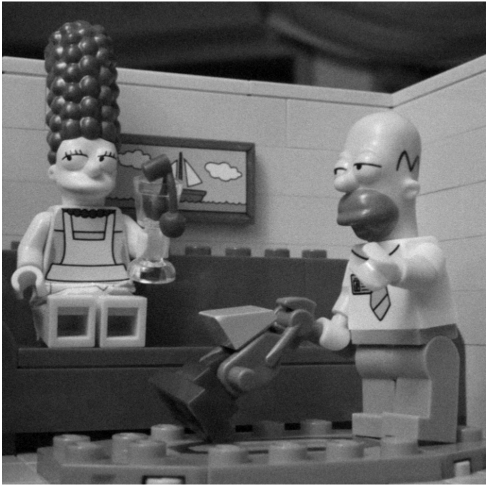
moyenne d'un stack de 10 images bruitées indépendamment

Moyenner pour débruiter !



moyenne d'un stack de 50 images bruitées indépendamment

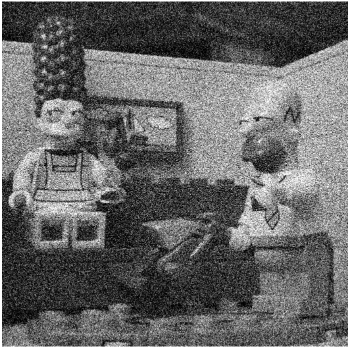
Moyenner pour débruiter !



moyenne d'un stack de 100 images bruitées indépendamment

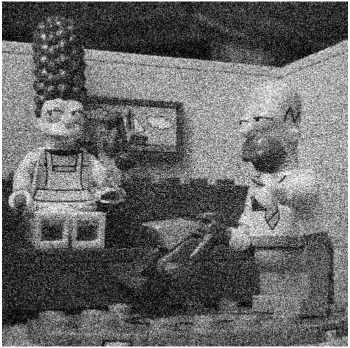
Malheureusement...

...on n'a généralement accès qu'à une seule image, donc une seule réalisation de bruit !



Malheureusement...

...on n'a généralement accès qu'à une seule image, donc une seule réalisation de bruit !



Une première idée ?

Moyennes locales

L'œil humain "sait" débruiter : il est capable de faire des moyennes locales

Moyennes locales

L'œil humain "sait" débruiter : il est capable de faire des moyennes locales

$\rightarrow$ on peut copier ce phénomène par un filtrage gaussien :

$$\hat{u}_i = \sum_j w_{ij} v_j$$

où $w_{ij} = \mathrm{e}^{-\frac{d(i,j)}{2h}}$

Moyennes locales

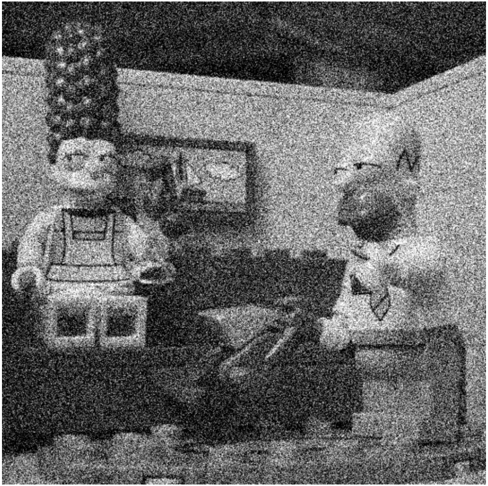
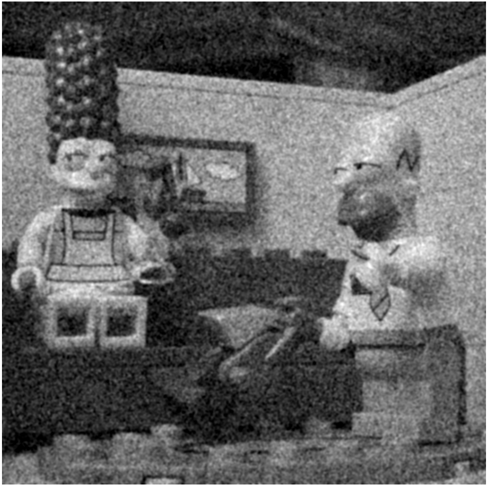


image bruitée à 20% *i.e.* $\sigma^2 = 0.2$ pour une image à valeur dans $[0, 1]$)

Moyennes locales



filtrage gaussien $h = 1$

Moyennes locales

filtrage gaussien $h = 2$

Moyennes locales

filtrage gaussien $h = 5$

Moyennes locales

filtrage gaussien $h = 10$

Moyennes locales $=$ destruction de la structure !

Moyennes selon les pixels qui se ressemblent

Dans la littérature : filtre de Yaroslavsky ou filtre bilatéral

L'idée est de moyenner les pixels qui se ressemblent :

$$\hat{u}_i = \sum_j w_{ij} v_j$$

avec $w_{ij} = \mathrm{e}^{-\frac{\|v_i - v_j\|}{2h_I}}$ (Yaroslavsky)

ou $w_{ij} = \mathrm{e}^{-\frac{d(i,j)}{2h_s}} \mathrm{e}^{-\frac{\|v_i - v_j\|}{2h_I}}$ (Bilatéral)

Moyennes selon les pixels qui se ressemblent

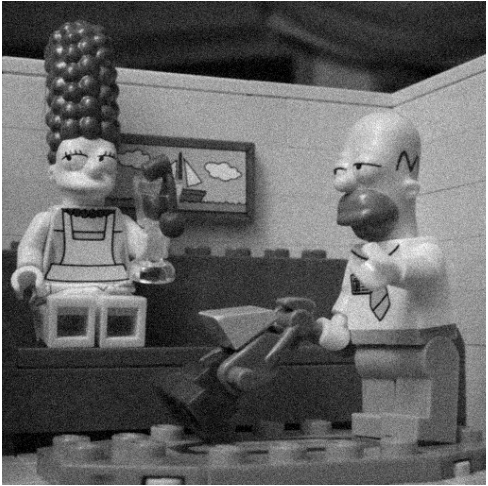
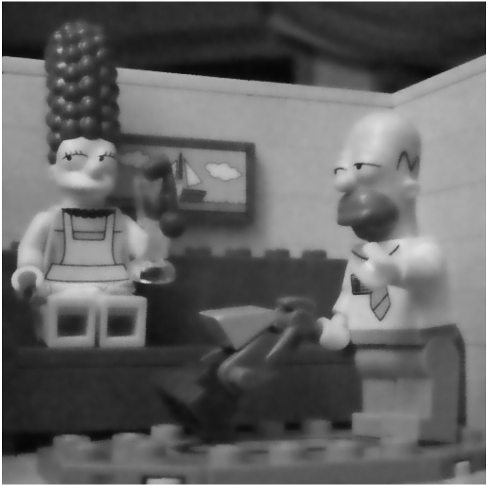


image bruitée à 5% *i.e.* $\sigma^2 = 0.05$

Moyennes selon les pixels qui se ressemblent



filtrage bilatéral $h_s = 3$ et $h_I = \sigma$

Moyennes selon les pixels qui se ressemblent

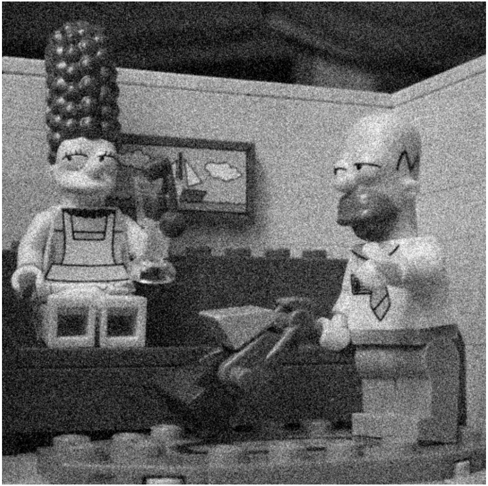
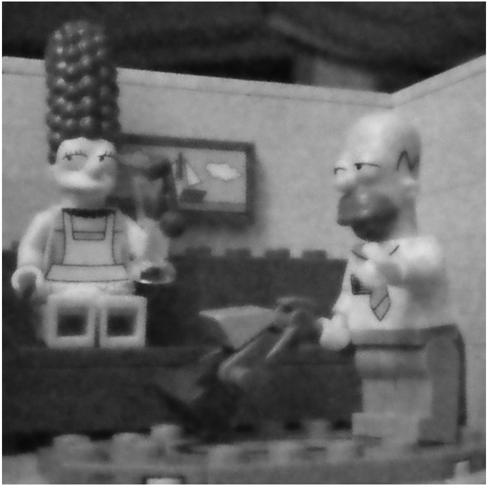


image bruitée à 10% *i.e.* $\sigma^2 = 0.1$

Moyennes selon les pixels qui se ressemblent



filtrage bilatéral $h_s = 3$ et $h_I = \sigma$

Moyennes selon les pixels qui se ressemblent

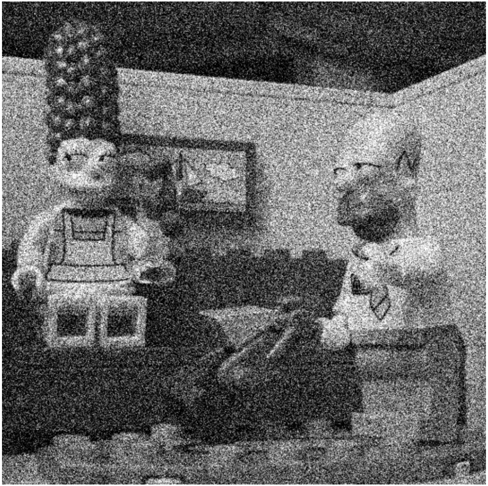
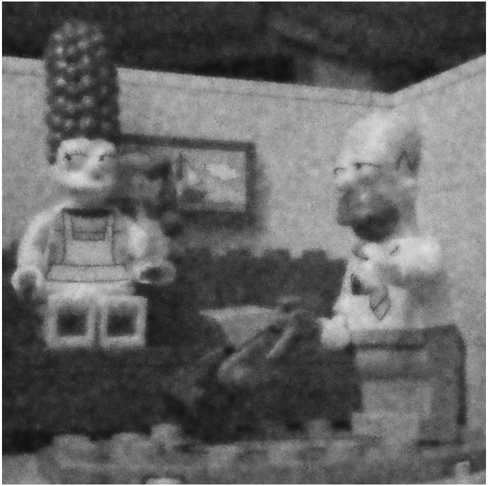


image bruitée à 20% *i.e.* $\sigma^2 = 0.2$

Moyennes selon les pixels qui se ressemblent



filtrage bilatéral $h_s = 3$ et $h_I = \sigma$

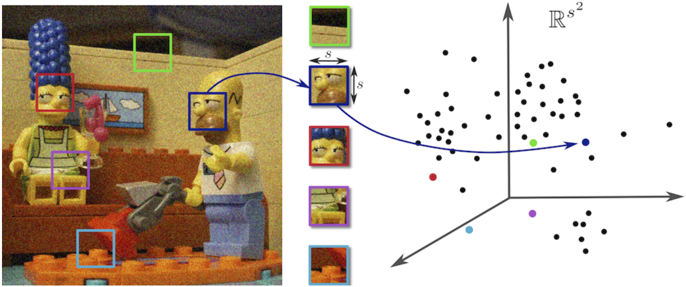
Moyennes selon les pixels qui se ressemblent

Moyennes pixeliques = Peu robuste au bruit !

Moyennes non-locales

Non-local means [Buades, Coll, Morel, 2005]

Tenir compte du contexte de chaque pixel → introduction des patchs



$$\hat{u}_i = \sum_j w_{ij} v_j \quad \text{avec} \quad w_{ij} = \mathrm{e}^{-\frac{\|P_i - P_j\|}{2h}}$$

`http://demo.ipol.im/demo/bcm_non_local_means_denoising/`

Moyennes selon les pixels qui se ressemblent

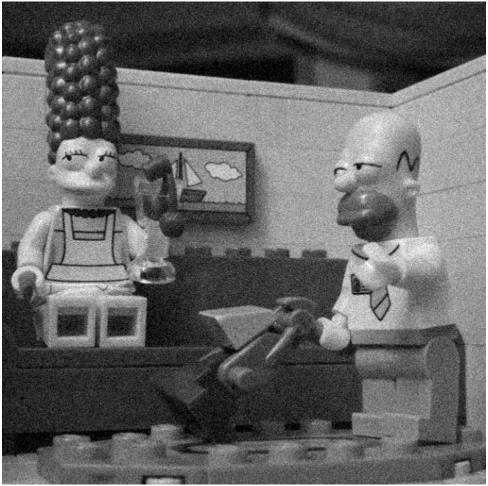
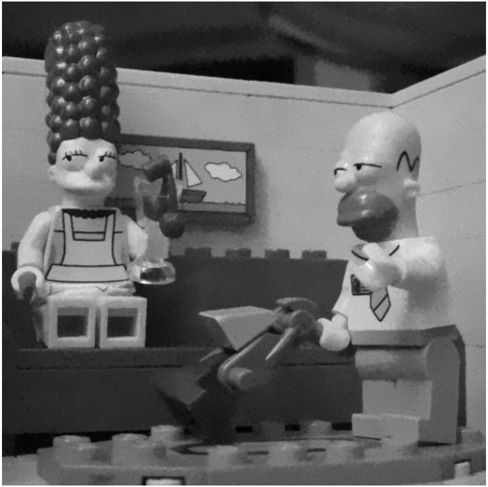


image bruitée à 5% *i.e.* $\sigma^2 = 0.05$

Moyennes selon les pixels qui se ressemblent



Non-local means

Moyennes selon les pixels qui se ressemblent

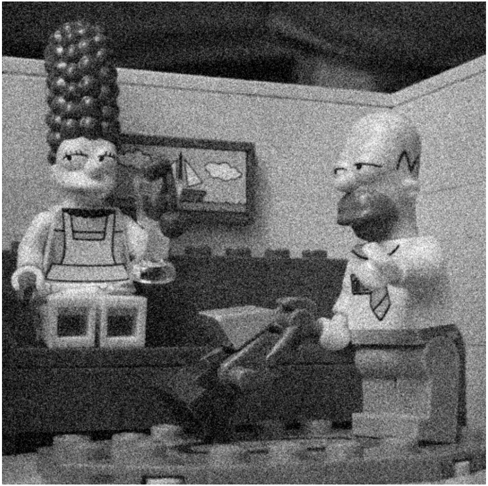
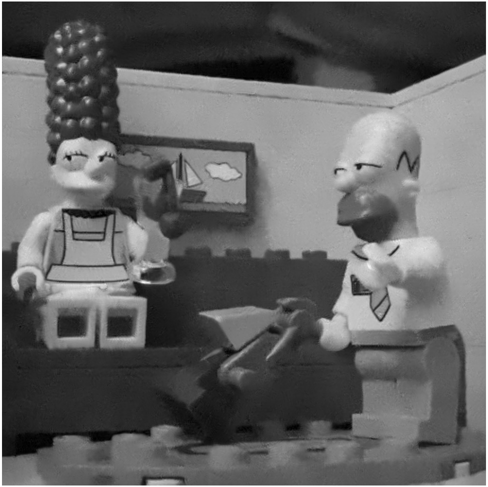


image bruitée à 10% *i.e.* $\sigma^2 = 0.1$

Moyennes selon les pixels qui se ressemblent



Non-local means

Moyennes selon les pixels qui se ressemblent

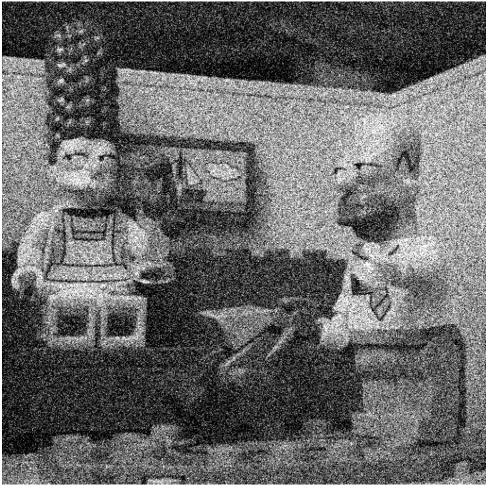
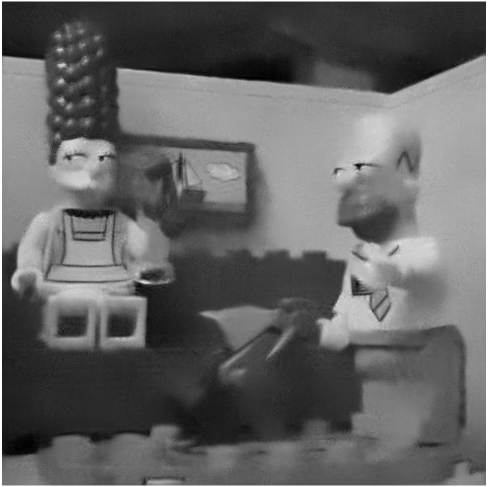


image bruitée à 20% *i.e.* $\sigma^2 = 0.2$

Moyennes selon les pixels qui se ressemblent



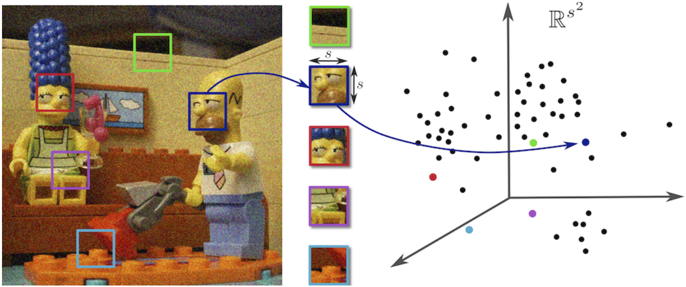
Non-local means

Moyennes selon les pixels qui se ressemblent

Non-local means = avènement du patch !

Patch-based image denoising

- Many denoising methods rely on the description of the image by patches:
  - NL-means Buades, Coll, Morel (2005),
  - BM3D Dabov, Foi, Katkovnik (2007),
  - PLE Yu, Sapiro, Mallat (2012),
  - NL-Bayes Lebrun, Buades, Morel (2012),
  - LDMM Shi, Osher, Zhu (2017),
  - and many others...



Avant d'aller plus loin...

Vecteur gaussien Un vecteur aléatoire $V = (V_1, \dots, V_n)$ est gaussien si, et seulement si, toute combinaison linéaire de ses composantes est une variable aléatoire gaussienne

Un vecteur gaussien $V = (V_1, \dots, V_n)$ est entièrement déterminé par sa moyenne et sa matrice de covariance :

$$\Sigma = \begin{pmatrix} \mathrm{Var}(V_1) & \mathrm{Cov}(V_1, V_2) & \cdots & \mathrm{Cov}(V_1, X_n) \\ \mathrm{Cov}(V_1, V_2) & \ddots & & \\ \vdots & & & \\ \mathrm{Cov}(V_1, V_n) & \cdots & & \mathrm{Var}(V_n) \end{pmatrix} \text{ et } \mu = \begin{pmatrix} \mathbf{E}(V_1) \\ \mathbf{E}(V_2) \\ \vdots \\ \mathbf{E}(V_n) \end{pmatrix}$$

Avant d'aller plus loin...

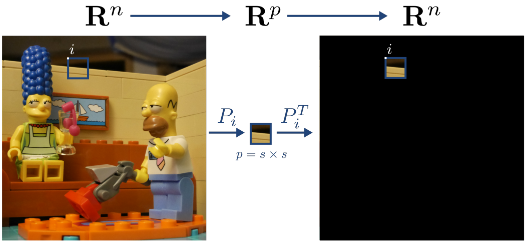
$$V_i = u_i + \mathcal{E}_i$$

Le vecteur $\mathcal{E} = (\mathcal{E}_1, \ldots, \mathcal{E}_n)$ de bruit sur est un vecteur gaussien

$$V = u + \mathcal{E} \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$$

Patch-based image denoising

⋆ Patch extraction operators



⋆ Noise model on the image

$$V = u + \mathcal{E} \longrightarrow \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$$

⋆ Noise model on the patches

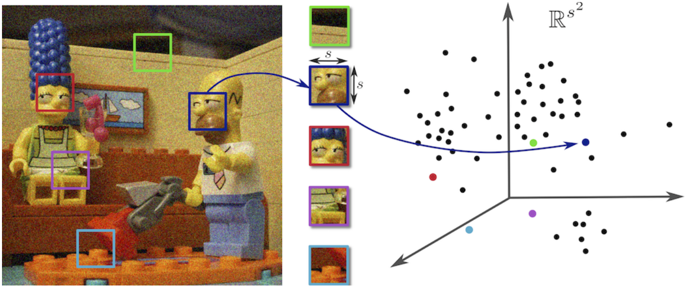
$$P_i V = P_i u + P_i \mathcal{E} \longrightarrow \mathcal{N}(0, P_i \sigma^2 \mathrm{I}_n P_i^T)$$

$$\downarrow \text{def.} \qquad \downarrow \text{def.} \qquad \downarrow \text{def.}$$

$$Y_i = x_i + N_i \longrightarrow \mathcal{N}(0, \sigma^2 \mathrm{I}_{s^2})$$

Hypothesis: the N_i are *i.i.d.*

Patch-based image denoising



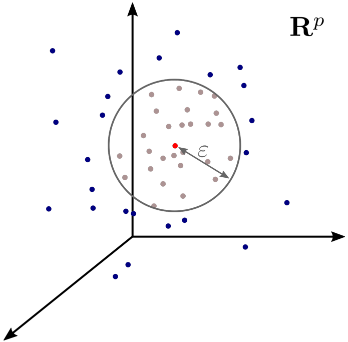
$$Y_i = x_i + N_i$$

$N_i \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$ Hypothèse: N_i indépendants

Pourquoi ne pas travailler directement dans l'espace des patchs ?

Un filtre Non-local légèrement modifié :

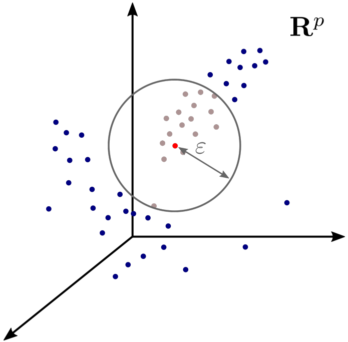
$$\hat{x}_i = \frac{\sum_j w_{ij} y_j}{\sum_j w_{ij}} \quad \text{avec} \quad w_{ij} = \mathbf{1}_{\|P_i - P_j\| \leqslant \epsilon}$$



Chaque patch est débruité par la moyenne empirique d'un petit échantillon autour de lui

Un filtre Non-local légèrement modifié :

$$\hat{x}_i = \frac{\sum_j w_{ij} y_j}{\sum_j w_{ij}} \quad \text{avec} \quad w_{ij} = \mathbf{1}_{\|P_i - P_j\| \leqslant \epsilon}$$

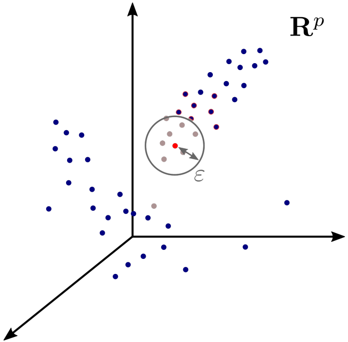


Chaque patch est débruité par la moyenne empirique d'un petit échantillon autour de lui

un NL-means dans l'espace des patchs

Un filtre Non-local légèrement modifié :

$$\hat{x}_i = \frac{\sum_j w_{ij} y_j}{\sum_j w_{ij}} \quad \text{avec} \quad w_{ij} = \mathbf{1}_{\|P_i - P_j\| \leqslant \epsilon}$$

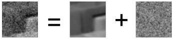


Chaque patch est débruité par la moyenne empirique d'un petit échantillon autour de lui

The Bayesian paradigm

- ⋆ We consider each clean patch x as a realization of a random vector X with *prior* distribution P_X.

→ The Gaussian white noise model rewrites:



$$Y = X + N \sim \mathcal{N}(0, \sigma^2 \mathbf{I}),$$

then Bayes' theorem yields the posterior distribution:

$$P_{X|Y}(x|y) = \frac{P_{Y|X}(y|x) P_X(x)}{P_Y(y)}.$$

L'espérance conditionnelle

Meilleur estimateur du patch x en terme d'erreur quadratique moyenne

sachant le modèle X et l'observation y

$$\hat{x}_{MMSE} := \mathbf{E}[X|Y=y] = \int x P_{X|Y}(x|y) \mathrm{d}x$$

Nécessite une connaissance explicite de $P_{X|Y}(x|y)$!

Le maximum *a posteriori*

$$\begin{aligned}
\hat{x}_{MAP} &= \arg\max_x P_{X|Y}(x|y) \\
&= \arg\max_x P_{Y|X}(y|x)P_X(x) \quad P_Y(y) \text{ indépendant de } x \\
&= \arg\min_x -\log\left(P_{Y|X}(y|x)\right) - \log\left(P_X(x)\right) \\
&= \arg\min_x \frac{\|x-y\|^2}{\sigma^2} - \log\left(P_X(x)\right)
\end{aligned}$$

On retrouve un problème variationnel avec un terme d'attache aux données et un terme de régularisation

Peut être loin du MMSE !

Le meilleur estimateur linéaire

L'espérance conditionnelle est un vecteur aléatoire $\mathbf{E}[X|Y] = g(Y)$ où

$$g = \arg\min_f \mathbf{E}\left[\|X - g(Y)\|^2\right]$$

Le meilleur estimateur linéaire LMMSE est une approximation linéaire $g(Y)$

$$\hat{x}_{LMMSE} = \hat{D}y + \hat{\alpha} \quad \text{avec}$$

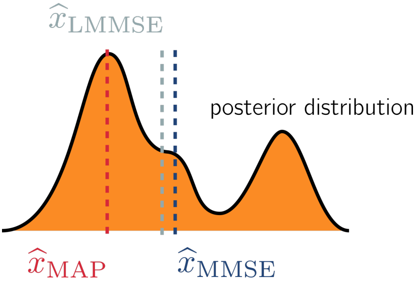
$$\hat{D}, \hat{\alpha} = \arg\min_{D,\alpha} \mathbf{E}\left[\|X - DY - \alpha\|^2\right]$$

si X et Y admettent des moments d'ordre 1 et 2 $\mu_Y, \mu_Y, \Sigma_X, \Sigma_Y$ pour notre problème de débruitage on trouve

$$\hat{x}_{LMMSE} = \mu_X + \Sigma_X \left(\Sigma_X + \sigma^2 \mathrm{I}\right)^{-1} (y - \mu_X)$$

aussi appelé Wiener estimator

Débruiter sachant un modèle *a priori*



Récapitulatif

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

Esquisse d'un algorithme

1. Extraire les patchs de l'image avec les opérateurs P_i

1. Extraire les patchs de l'image avec les opérateurs P_i

2. Définir et apprendre un modèle *a priori* X sur les patchs sans bruit

1. Extraire les patchs de l'image avec les opérateurs P_i

2. Définir et apprendre un modèle *a priori* X sur les patchs sans bruit

3. Débruiter chaque patch avec un des estimateurs précédent

Esquisse d'un algorithme

1. Extraire les patchs de l'image avec les opérateurs P_i
2. Définir et apprendre un modèle *a priori* X sur les patchs sans bruit
3. Débruiter chaque patch avec un des estimateurs précédent
4. Reconstruire l'image à partir de ses patchs débruités

1. Extraire les patchs de l'image avec les opérateurs P_i

2. **Définir** et **apprendre** un modèle *a priori* X sur les patchs sans bruit

3. Débruiter chaque patch avec un des estimateurs précédent

4. Reconstruire l'image à partir de ses patchs débruités

Modéliser les patchs sans bruit X_i

Choix du modèle

Dans la littérature

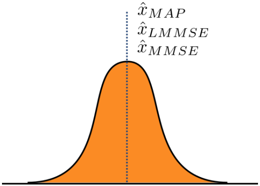
- Modèles gaussiens locaux
  - ⋆ patch-based PCA Deledalle, Salmon, Dalalyan (2011),
  - ⋆ NL-bayes Lebrun, Buades, Morel (2012),
  - ⋆ ...
- Modèles de mélange de gaussiennes
  - ⋆ EPLL Zoran, Weiss (2011),
  - ⋆ PLE Yu, Sapiro, Mallat (2012),
  - ⋆ Single-frame Image Denoising Teodoro, Almeida, Figueiredo (2015).
  - ⋆ ...

Why Gaussian models are so widely used?

Gaussian is convenient

Si $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ alors tous les estimateurs introduit précédemment coïncident

$$\widehat{x}_{\mathsf{MMSE}} = \widehat{x}_{\mathsf{Wiener}} = \widehat{x}_{\mathsf{MAP}} = \mu_X + \Sigma_X(\Sigma_X + \sigma^2 \mathrm{I})^{-1}(y - \mu_X)$$



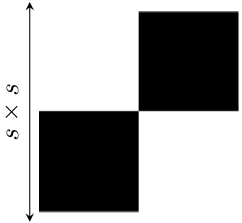
la distribution de X conditionnellement à Y est un vecteur gaussien de moyenne m et de covariance S

$$m = \mu_X + \Sigma_X(\Sigma_X + \sigma^2 \mathrm{I})^{-1}(y - \mu_X)$$

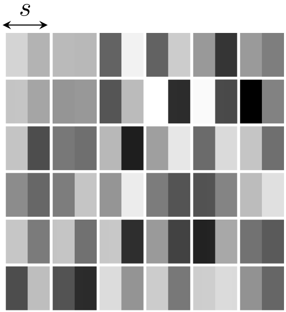
$$S = \Sigma_X - \Sigma_X(\Sigma_X + \sigma^2 \mathrm{I})^{-1}\Sigma_X$$

What do Gaussian models encode?

The covariance matrix in Gaussian models and GMM encodes geometric structures up to some contrast change:



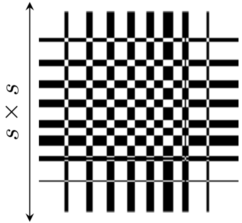
Covariance matrix Σ.



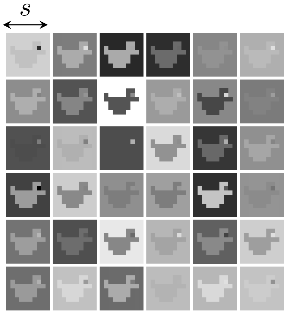
Patches generated from $\mathcal{N}(m, \Sigma)$.

What do Gaussian models encode?

The covariance matrix in Gaussian models and GMM encodes geometric structures up to some contrast change:



Covariance matrix Σ.



Patches generated from $\mathcal{N}(m, \Sigma)$.

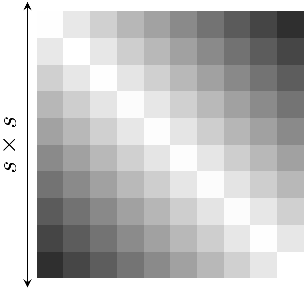
What do Gaussian models encode?

A covariance matrix cannot encode multiple translated versions of a structure:

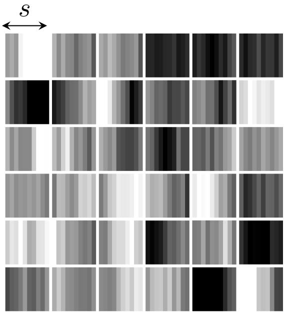
A set of 10000 patches representing edges with random grey levels and random translations.

What do Gaussian models encode?

A covariance matrix cannot encode multiple translated versions of a structure:

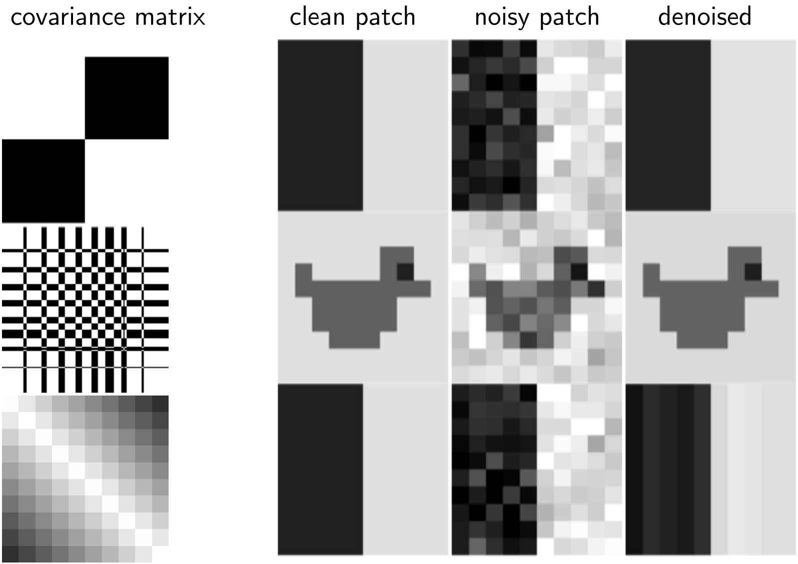


Covariance matrix Σ.



Patches generated from $\mathcal{N}(m, \Sigma)$.

Restore with the right model



Conclusion sur les modèles gaussiens

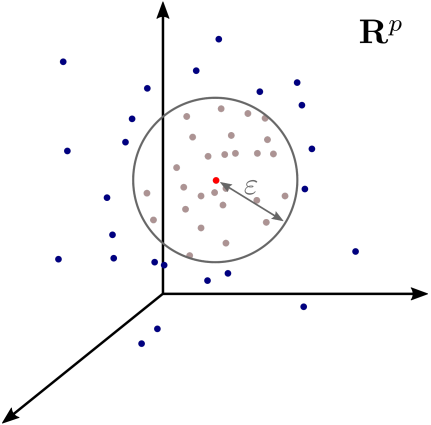
Modéliser les patchs par un modèle gaussien

- permet d'avoir une formule explicite pour l'estimateur de chaque patch
- permet de représenter une structure géométrique spécifique à l'échelle du patch

Nécessité de modèles locaux : comment grouper les patchs ?

Comment grouper les patchs ?

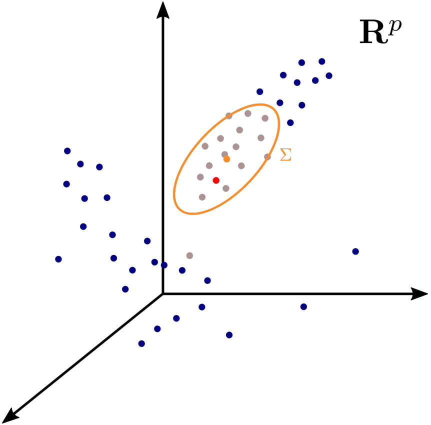
Non-local Bayes [Lebrun, Buades, Morel (2013)]



http://demo.ipol.im/demo/16/

Comment grouper les patchs ?

Non-local Bayes [Lebrun, Buades, Morel (2013)]



`http://demo.ipol.im/demo/16/`

Résultats Non-Local Bayes

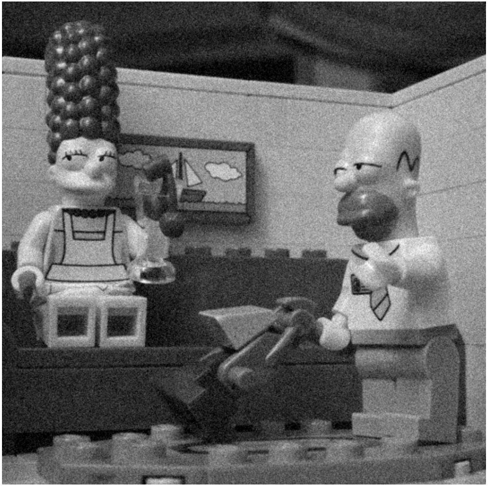
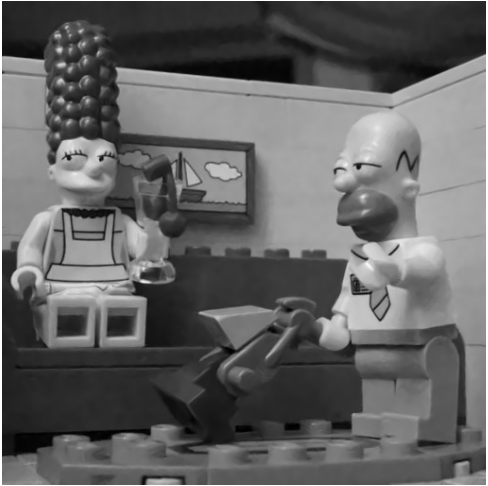


image bruitée à 5% *i.e.* $\sigma^2 = 0.05$

Résultats Non-Local Bayes



Non-local Bayes

Résultats Non-Local Bayes

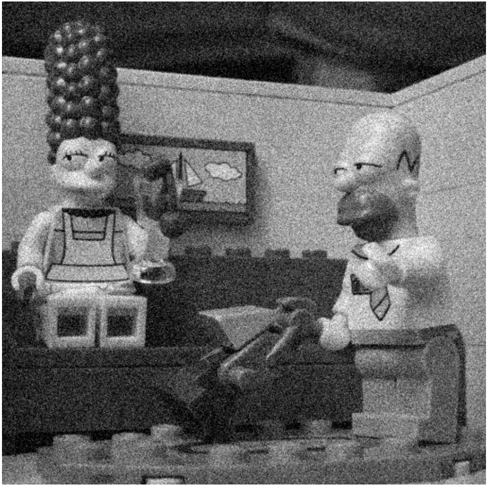
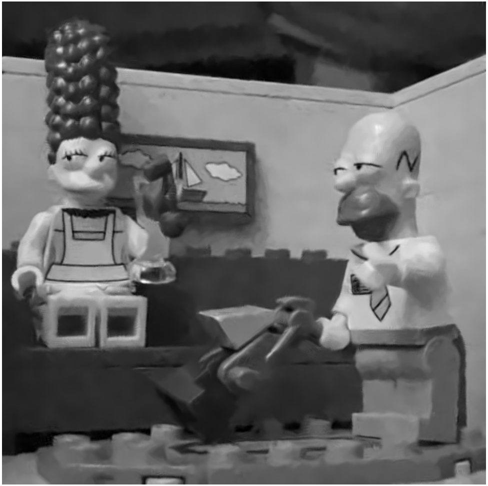


image bruitée à 10% *i.e.* $\sigma^2 = 0.1$

Résultats Non-Local Bayes



Non-local Bayes

Résultats Non-Local Bayes

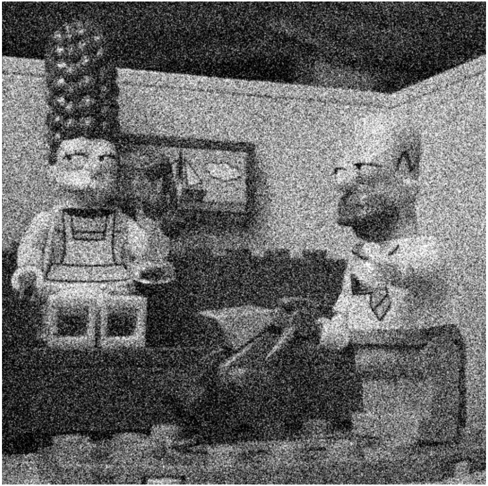


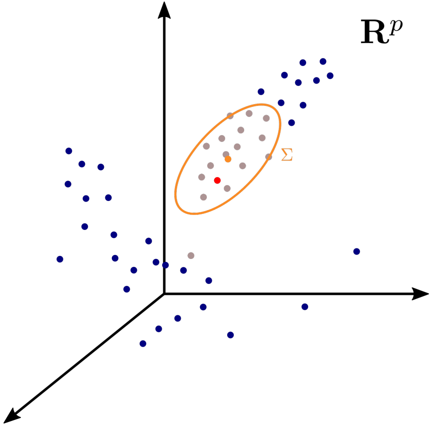
image bruitée à 20% *i.e.* $\sigma^2 = 0.2$

Résultats Non-Local Bayes

Non-local Bayes

Comment grouper les patchs ?

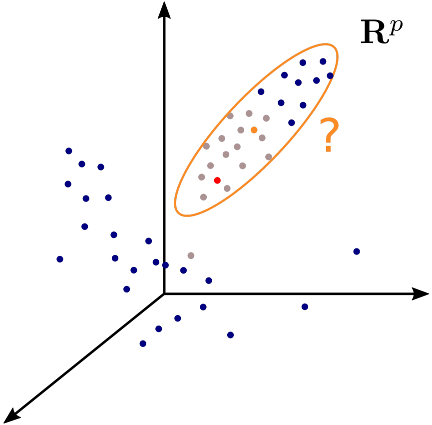
Pourquoi aller chercher un voisinage dans une boule ?



Utiliser la géométrie de l'espace des patchs

Comment grouper les patchs ?

Pourquoi aller chercher un voisinage dans une boule ?

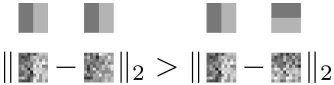


Utiliser la géométrie de l'espace des patchs

Comment grouper les patchs ?

On chercher à regrouper les patchs représentant la même structure

- la norme $\|\cdot\|_2 \rightarrow$ n'est pas robuste au fort bruit

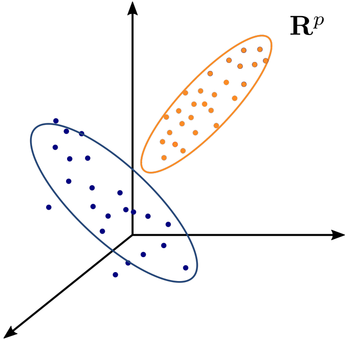


- Gaussian Mixture Models (GMM) naturally provide a (more robust) grouping!

Les modèles de mélange de gaussiennes

modéliser l'espace des patchs par un modèle de mélange de gaussiennes

$$X \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$$



$$\widehat{x}_{\mathsf{MMSE}} = \sum_{k=1}^{K} \mathbb{P}(Z = k | Y = y) \left[\mu_k + \Sigma_k (\Sigma_k + \sigma^2 \mathrm{I})^{-1} (y - \mu_k) \right]$$

1. Extraire les patchs de l'image avec les opérateurs P_i
2. **Définir** et **apprendre** un modèle *a priori* X sur les patchs sans bruit
3. Débruiter chaque patch avec un des estimateurs précédent
4. Reconstruire l'image à partir de ses patchs débruités

Comment inférer les paramètres ?

Parameters inference

Gaussian model case: $X \sim \mathcal{N}(\mu_X, \Sigma_X)$

observed data $\{y_1, \ldots, y_n\}$ sampled from $Y = X + N \sim \mathcal{N}(\mu_Y, \Sigma_Y)$.

The maximization of the likelihood

$$\mathcal{L}(y;\theta) = \frac{1}{2}\sum_{i=1}^{n}(y-\mu_Y)^T {\Sigma_Y}^{-1}(y-\mu_Y),$$

yields the Maximum Likelihood estimators (MLE)

$$\widehat{\mu}_Y = \frac{1}{n}\sum_{i=1}^{n} y_i, \quad \widehat{\Sigma}_Y = \frac{1}{n}\sum_{i=1}^{n}(y_i-\widehat{\mu}_Y)^T(y_i-\widehat{\mu}_Y).$$

Since $\Sigma_Y = \Sigma_X + \sigma^2 \mathrm{I}_p$, it yields

$$\widehat{\mu}_X = \widehat{\mu}_Y, \quad \widehat{\Sigma}_X = \widehat{\Sigma}_Y - \sigma^2 \mathrm{I}_p.$$

Parameters inference

Gaussian Mixture Model case: $X \sim \sum \pi_k \mathcal{N}(\mu_k, \Sigma_k)$

This implies a GMM on the noisy patches $Y \sim \sum \pi_k \mathcal{N}(\mu_k, S_k)$

EM algorithm: maximize the conditional expectation of the complete log-likelihood:

$$\sum_{k=1}^{K} \sum_{i=1}^{n} t_{ik} \log\left(\pi_k g\left(y_i; \theta_k\right)\right),$$

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

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

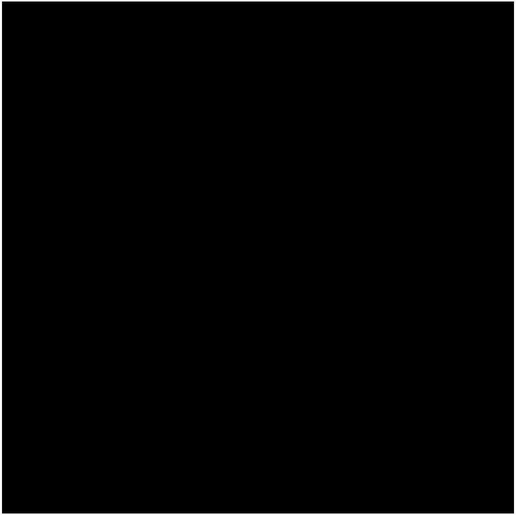
$$\widehat{\pi}_k = \frac{n_k}{n}, \quad \widehat{\mu}_k = \frac{1}{n_k}\sum_i t_{ik} y_i, \quad \widehat{S}_k = \frac{1}{n_k}\sum_i t_{ik}(y_i - \mu_k)(y_i - \mu_k)^T,$$

with $n_k = \sum_i t_{ik}$.

Esquisse d'un algorithme

1. Extraire les patchs de l'image avec les opérateurs P_i
2. **apprendre** un modèle GMM pour les patchs sans bruit
3. Débruiter chaque patch avec le MMSE
4. Reconstruire l'image à partir de ses patchs débruités

NaN



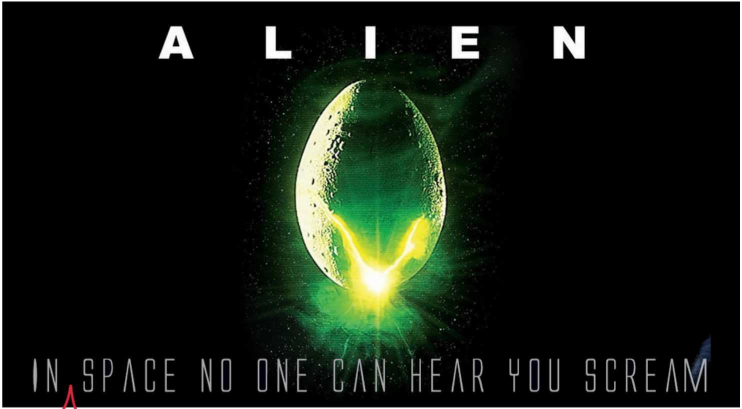
The curse of dimensionality

Parameter estimation for Gaussian models or GMMs suffers from the curse of dimensionality

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

The number of samples needed for the estimation of a parameter grows exponentially with the dimension

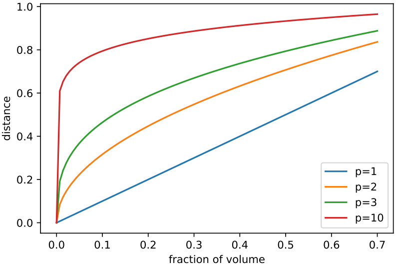
The curse of dimensionality



High-Dimensional

The curse of dimensionality

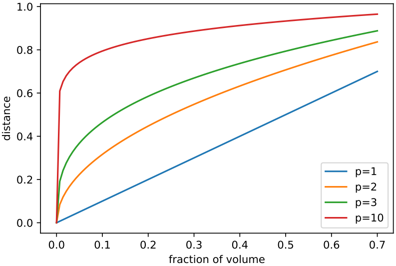
Consider data uniformly sampled in $[0,1]^p$. An hypercube of size $s^{1/p}$ is needed to capture a neighborhood of a point representing $s\%$ of the total volume.



To capture 10% of your data in order to compute a local average, you must cover 80% of the range of each input variable!

The curse of dimensionality

Consider data uniformly sampled in $[0,1]^p$. An hypercube of size $s^{1/p}$ is needed to capture a neighborhood of a point representing $s\%$ of the total volume.



To capture 10% of your data in order to compute a local average, you must cover 80% of the range of each input variable!

$\boxed{\text{Neighborhoods are no more local!}}$

The curse of dimensionality

The volume of an hypercube of size $r = 0.1$ is 0.1^p so when the dimension p grows, the probability that it contains a point of the dataset goes to zero..

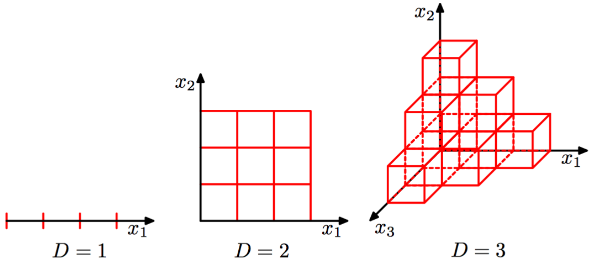
$\boxed{\text{In High-dimensional spaces, data are isolated}}$

The curse of dimensionality

The volume of an hypercube of size $r = 0.1$ is 0.1^p so when the dimension p grows, the probability that it contains a point of the dataset goes to zero..

In High-dimensional spaces, data are isolated

To tackle this, we need the number of data to exponentially increase with p...



The curse of dimensionality in patches space

We consider patches of size $p = 10 \times 10 \rightarrow$ High dimension.

$\rightarrow$ the estimation of sample covariance matrices is difficult: ill conditioned, singular...

The curse of dimensionality in patches space

We consider patches of size $p = 10 \times 10 \rightarrow$ High dimension.

$\rightarrow$ the estimation of sample covariance matrices is difficult: ill conditioned, singular...

In the literature, this issue is generally worked around by

- the use of small patches $(3 \times 3$ or $5 \times 5)$ NL-Bayes [Lebrun, Buades, Morel]
- adding $\varepsilon \mathrm{I}$ to singular covariance matrices PLE [Yu, Sapiro, Mallat]
- fixing a lower dimension for covariance matrices S-PLE [Wang, Morel]

But, there is no reason to be afraid of this curse!

The bless of dimensionality?

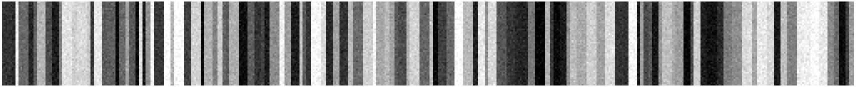
In high-dimensional spaces, it is easier to separate data:

Many patches represent structures that live locally in a low dimensional space: using this latent lower dimension allows to group the patches in a more robust way.

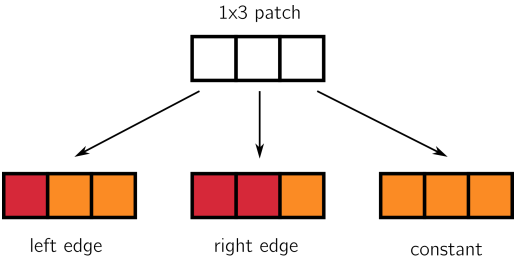
This "bless" is used in clustering algorithms designed for high-dimension
High-Dimensional Data Clustering [Bouveyron, Girard, Schmid] 2007

The bless of dimensionality?

An illustration in the context of patches:

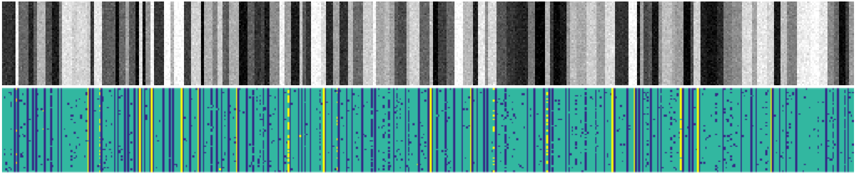


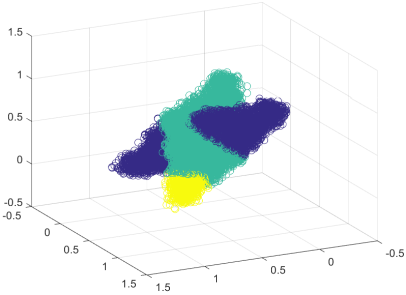
an image made of vertical stripes of width >2 pixels with random grey levels.



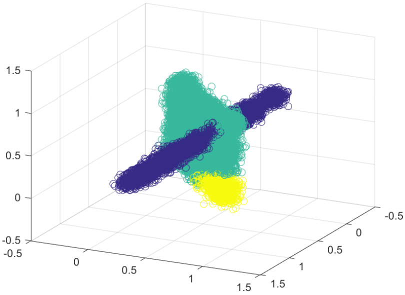
The bless of dimensionality?

An illustration in the context of patches:





view 1

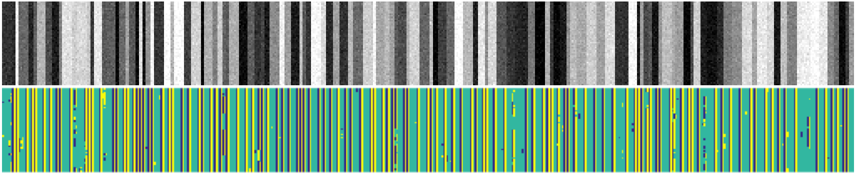


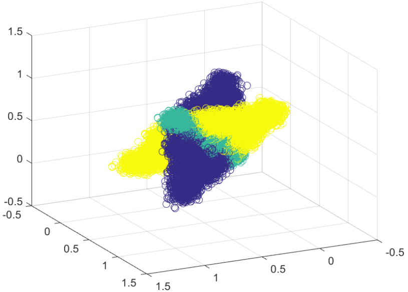
view 2

In the patch space, we cannot distinguish three classes

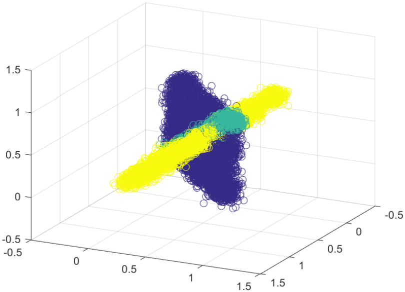
The bless of dimensionality?

An illustration in the context of patches:





view 1 of the first 3 pixels

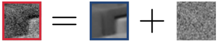


view 2 of the first 3 pixels

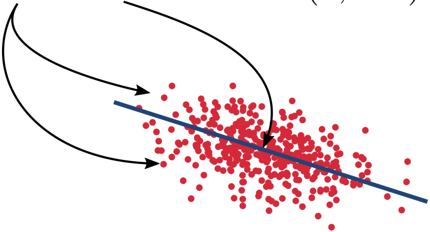
The algorithm is now able to separate these classes!

1. Extraire les patchs de l'image avec les opérateurs P_i
2. **Définir** et **apprendre** un modèle *a priori* avec une réduction de dimension X sur les patchs sans bruit
3. Débruiter chaque patch avec un des estimateurs précédent
4. Reconstruire l'image à partir de ses patchs débruités

High-Dimensional Mixture Models for Image Denoising



$$Y = X + N \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$



HDMI: presentation of the model

- **model the clean patches** X

+ Z latent random variable indicating group membership
+ X lives in a low-dimensional subspace which is specific to its latent group:

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

where U_k is a $p \times d_k$ orthogonal matrix and $\Lambda_k = \mathrm{diag}(\lambda_1^k, \dots, \lambda_{d_k}^k)$ a diagonal matrix of size $d_k \times d_k$.

HDMI: induced model

- **Induced model on the noisy patches Y**

The model on X implies that Y follows a full rank GMM

$$p(y) = \sum_{k=1}^{K} \pi_k g\left(y; \mu_k, \Sigma_k\right)$$

where $U_k \Sigma_k U_k^t$ has the specific structure:

$$\begin{array}{cc}\left(\begin{array}{ccc|ccc} a_{k1} & & 0 & & & \\ & \ddots & & & \mathbf{0} & \\ 0 & & a_{kd} & & & \\ \hline & & & \sigma^2 & & 0 \\ & \mathbf{0} & & & \ddots & \\ & & & 0 & & \sigma^2 \end{array}\right) & \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\ \ddots\\ 0\end{array}}\right\} \quad d_k \\ \left.\vphantom{\begin{array}{c}a\\ \ddots\\ 0\end{array}}\right\} \quad (p-d_k)\end{array}\end{array}$$

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

Denoising with the HDMI model

The HDMI model being known, each patch is denoised with the MMSE

$$\hat{x}_i = \mathbf{E}[X|Y = y_i] = \sum_{k=1}^{K} t_{ik}\psi_k(y_i)$$

where t_{ik} is the posterior probability for the patch y_i to belong in the k-th group and

$$\psi_k(y_i) = \mu_k + U_k \begin{pmatrix} \frac{a_{k1}-\sigma^2}{a_{k1}} & & 0 \\ & \ddots & \\ 0 & & \frac{a_{kd_k}-\sigma^2}{a_{kd_k}} \end{pmatrix} U_k^T (y_i - \mu_k).$$

Model inference

with an EM algorithm, the parameters are updated during the M-step :

- $\widehat{U}_k$ is formed by the d_k first eigenvectors of the sample covariance matrix
- $\widehat{a}_{kj}$ is the j-th eigenvalue of the sample covariance matrix

Model inference

with an EM algorithm, the parameters are updated during the M-step :

- $\widehat{U}_k$ is formed by the d_k first eigenvectors of the sample covariance matrix
- $\widehat{a}_{kj}$ is the j-th eigenvalue of the sample covariance matrix

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

→ Each set of K and $d_1, \ldots, d_K$ corresponds to a different model.

Model inference

We propose to set K at a given value and to choose the intrinsic dimensions d_k:

- using an heuristic that links d_k with the noise variance σ^2 when known;
- using a model selection tool in order to select the best variance σ^2 when unknown.

Estimation of intrinsic dimensions – known variance

With d_k begin fixed, the MLE for the noise variance in the kth group is

$$\widehat{\sigma}^2_{|k} = \frac{1}{p - d_k} \sum_{j=d_k+1}^{p} \widehat{a}_{kj}.$$

When the noise variance σ is known, this gives us the following heuristic:

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

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

Estimation of intrinsic dimensions – convergence

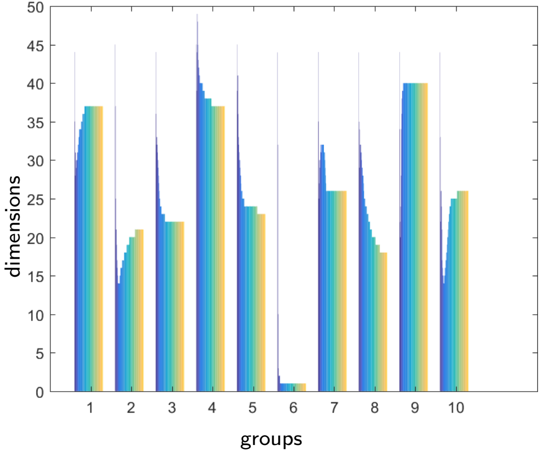
By re-evaluating the dimensions, we change the model at each M-step!

Question: is the convergence ensured?

Estimation of intrinsic dimensions – convergence

By re-evaluating the dimensions, we change the model at each M-step!

Question: is the convergence ensured?



the dimensions stabilize → there exists an iteration where the algorithm becomes a classic EM.

Estimation of intrinsic dimensions – unknown variance

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.

Why BIC is well-adapted for the selection of σ?

- If σ is too small, the likelihood is good but the complexity explodes;
- if σ is too high, the complexity is low but the likelihood is bad.

Estimation of intrinsic dimensions – unknown variance

$$\Delta_k = \left(\begin{array}{cc|c} \begin{array}{|ccc|} \hline a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd} \\ \hline \end{array} & \mathbf{0} \\ \hline \mathbf{0} & \begin{array}{ccc|} \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \sigma^2 \\ \hline \end{array} \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c} a \\ \ddots \\ a \end{array}}\right\} \quad d_k \\ \left.\vphantom{\begin{array}{c} a \\ \ddots \\ a \end{array}}\right\} \quad (p - d_k) \end{array}$$

Why BIC is well-adapted for the selection of σ?

- If σ is too small, the likelihood is good but the complexity explodes;
- if σ is too high, the complexity is low but the likelihood is bad.

Summary: the HDMI algorithm

We presented the HDMI model for image denoising:

- which models the full process of the generation of the noisy patches;
- a fully statistical modeling without the usual "denoising cuisine";
- can be used in a "blind" way thanks to BIC selection;
- attains state-of-the-art performances!

Numerical Experiments

Clean image

Numerical Experiments

Noisy image $\sigma = 50$

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

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

Numerical Experiments

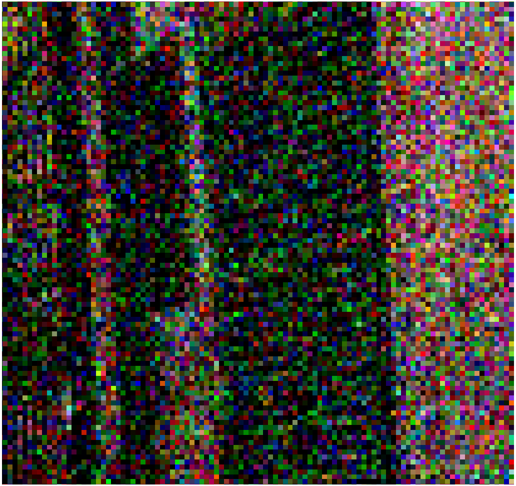
Denoised with HDMI $K = 50$, psnr = 27.28dB

Numerical Experiments – zooms

Clean image

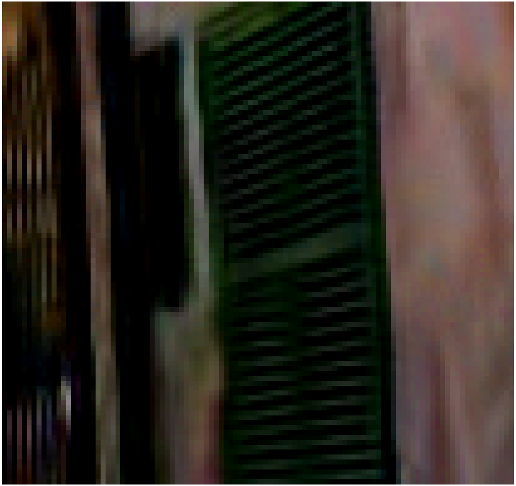
Numerical Experiments – zooms

Noisy image $\sigma = 50$



Numerical Experiments – zooms

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



Numerical Experiments – zooms

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

Numerical Experiments – zooms

Denoised with HDMI $K = 50$, psnr = 27.28dB

Numerical Experiments

Clean image

Numerical Experiments

Noisy image $\sigma = 50$

Numerical Experiments

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

Numerical Experiments

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

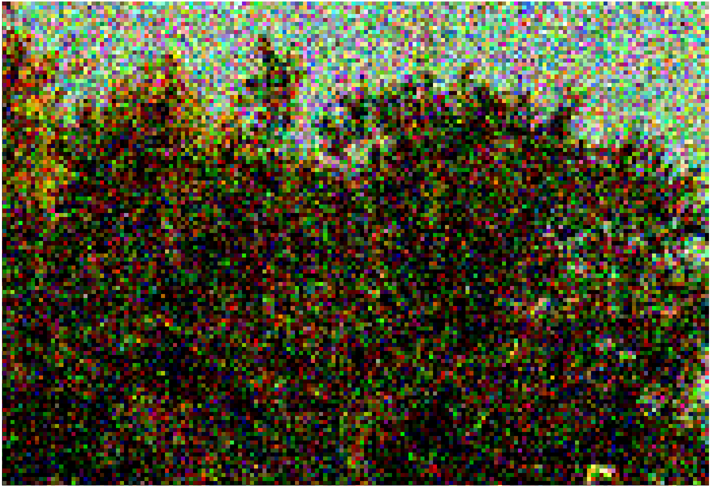
Numerical Experiments

Denoised with HDMI $K = 50$, psnr = 27.05dB

Clean image

Numerical Experiments – zooms

Noisy image $\sigma = 50$



Numerical Experiments – zooms

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

Numerical Experiments – zooms

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

Numerical Experiments – zooms

Denoised with HDMI $K = 50$, psnr = 27.05dB

Limitations of denoising in the patch-space

The lower bound for patch-based image denoising

"Is denoising dead" [Chatterjee, Milanfar] 2010 proposed a lower bound for patch-based image denoising.

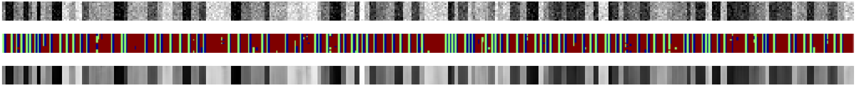
In this context, denoting m_k the number of patches in the k-th group and N the total number of patches, the bound for HDMI is

$$\begin{aligned}\mathbf{E}\left[\|u-\widehat{u}_{\mathsf{HDMI}}\|^2\right] &\geqslant \frac{1}{N}\sum_{k=1}^{K} m_k \frac{\mathrm{Tr}(\Sigma_k)\sigma^2}{p+\sigma^2}, \\ &\geqslant C\frac{\sigma^2}{N(p+\sigma^2)}\sum_{k=1}^{K} m_k \\ &= C\frac{\sigma^2}{p+\sigma^2} \quad \text{independent of N.}\end{aligned}$$

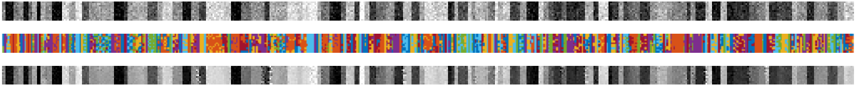
even if the number of samples increases by stretching the image size to infinity, the noise variance cannot be reduced more than a factor p.

The lower bound for patch-based image denoising

HDMI (patches 3×10) - PSNR = 30.12

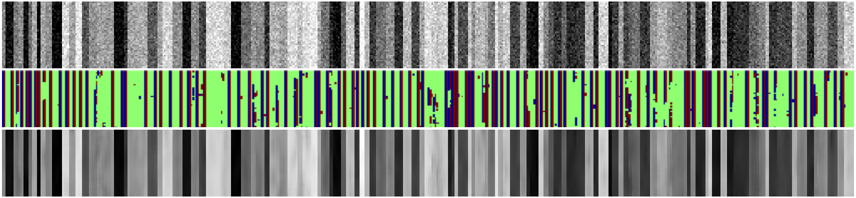


L2 grouping (patches 3×10) - PSNR = 25.03

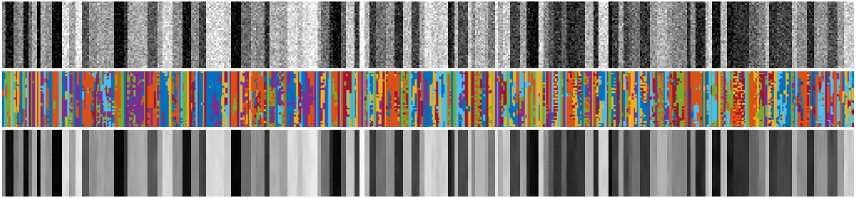


The lower bound for patch-based image denoising

HDMI (patches 3×10) - PSNR $=$ 30.27



L2 grouping (patches 3×10) - PSNR $=$ 30.84



cropped: actual images height is 500 pixels.

The low frequency noise

Denoised with HDMI $K = 50$, psnr = 36.47 dB

Removing low frequency noise by denoising the DC component (preprint)

- Define the centered observed random variable $Y_i^c = Y_i - \bar{Y}_i \mathbf{1}_p$, where

$$\bar{Y}_i = \frac{1}{p} \sum_{j=1}^{p} Y_i(j),$$

  is the DC component of the patch.

- The noise model can then be divided into the two following problems

$$\bar{Y}_i = \bar{X}_i + \bar{N}_i \in \mathbf{R}, \tag{1}$$

$$Y_i^c = X_i^c + N_i^c \in \mathbf{R}^p. \tag{2}$$

Modélisation du bruit centré

$N_i^c = N_i - \overline{N_i}\mathbf{1}_p$ est un vecteur gaussien et $N_i^c \sim \mathcal{N}(0, \Sigma_{N_i^c})$ avec

$$\Sigma_{N_i^c} = \frac{\sigma^2}{p} \begin{pmatrix} p-1 & -1 & \cdots & -1 \\ -1 & p-1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & p-1 \end{pmatrix}.$$

Il existe une matrice orthogonale Q telle que

$$\Sigma_{N_i^c} = Q \begin{pmatrix} \sigma^2 \mathrm{I}_{p-1} & 0 \\ 0 & 0 \end{pmatrix} Q^T.$$

$Q^T N_i^c \sim \mathcal{N}(0, \mathrm{diag}(\sigma^2 \mathrm{I}_{p-1}, 0))$ bruit blanc gaussien réduit d'une dimension !

Modélisation de la composante moyenne

- The DC component can be reshaped as an image

 $\longrightarrow$ 

- Extract patches from this image yields additive Gaussian noise problem with colored noise

- A change of basis brings us back to an additive white Gaussian noise

Débruitage des deux problème séparément

On se retrouve avec deux problèmes de bruit blanc gaussien

→ utilisation de son algorithme de débruitage des patchs préféré HDMI

Pour chaque patch i

- $\widehat{\overline{X}}_i$ estimateur de sa moyenne
- $\widehat{X_i^c}$ estimateur du patch centré

$$\boxed{\widehat{X}_i = \widehat{X_i^c} + \widehat{\overline{X}}_i \mathbf{1}_p.}$$

Results

Noisy with $\sigma = 50$

Results

Denoised with HDMI $K = 50$, psnr = 36.47 dB

Results

+ corrected DC component (HDMI $K = 30$), psnr = 36.90 dB

Un petit résumé de ce qu'on a abordé

- différentes méthodes de débruitage issus de la modélisation du bruit
- une étude des modèles gaussiens pour le débruitage par patchs
- une modélisation statistique de l'espace des patchs
- un combat contre les mystères de la grande dimension
- les limitations de ce type d'approche

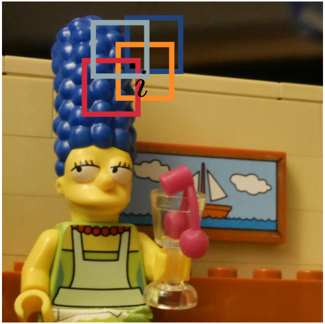
Merci de votre attention !

des questions ?

Retrouvez l'article HDMI et le preprint sur ma page
houdard.wp.imt.fr

Aggregation problem

Each pixel belongs in p patches:



In all the experiments here: uniform aggregation.

In the literature: there exist different aggregation methods
→ able to improve visual results but in many cases, the final pixel is still obtained from a fixed number of realizations.

Other inverse problem : missing pixels

70% missing pixels

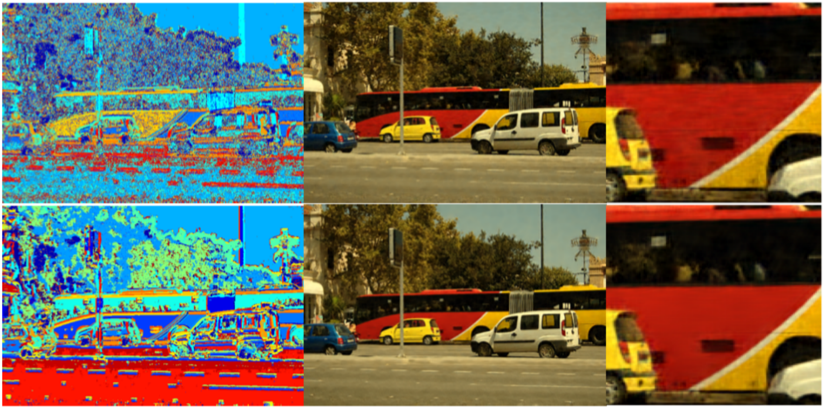
EM is well-adapted for missing data → the model can be easily adapted for missing pixel restoration

Other inverse problem : missing pixels

restored with HDMI

EM is well-adapted for missing data → the model can be easily adapted for missing pixel restoration

Regularizing effect of the dimension reduction



The HDMI algorithm

Input u noisy image, p patch size, K number of groups, $\{\sigma_1, \ldots, \sigma_m\}$ list of standard deviation.
Output $\hat{u}$ denoised image.
Extract $\{y_1, \ldots, y_n\}$ patches from u;
for $\sigma = \sigma_1, \ldots, \sigma_m$ **do**
  Initialization few iteration of k-means.
  $dl \leftarrow \infty$.
  while $dl > \epsilon$ **do**
    M-step update parameters and dimensions d_k
    E-step compute t_{ik}.
    update the log-likelihood l and compute the relative error $dl = |l - lex|/|l|$.
    $lex \leftarrow l$.
  end while
  compute the BIC for the model associated with σ
end for
select the model with the better BIC.
compute denoised patches $\{x_1, \ldots, x_n\}$ with conditional expectation;
aggregate patches x_i in order to recover the denoised image v.

Learning on a sub-sample of the patches

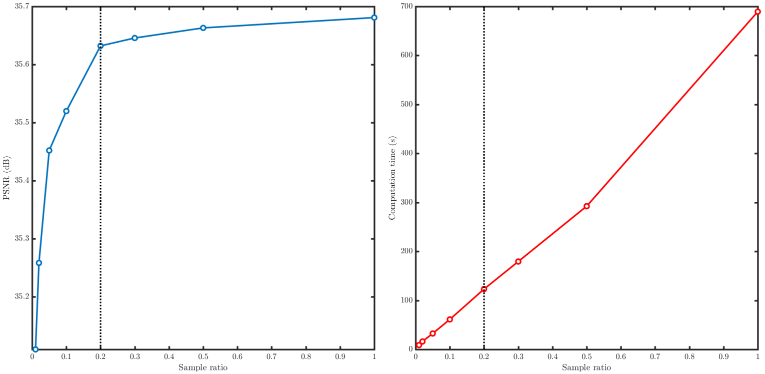
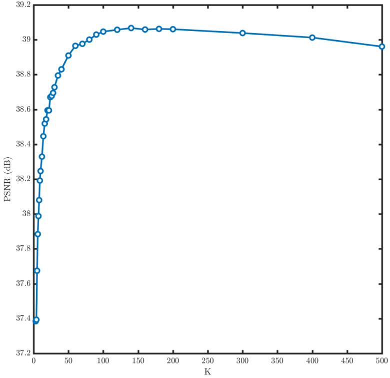


Figure: Effect of the subsampling on the computing time and the denoising performance with HDMI. Left: PSNR versus sampling size. Right: Computation time versus same sampling size. Dotted-lines: 20% subsampling.

Influence of the number of group K



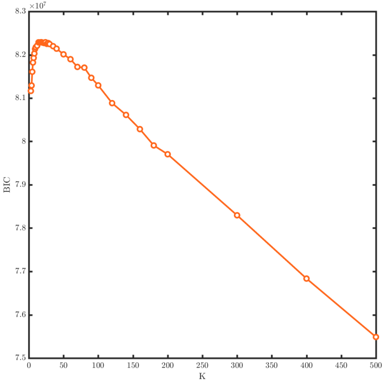


Figure: Denoising results (PSNR) with regard to K (left) and choice of K with BIC (right).

Selection of σ^2 with BIC