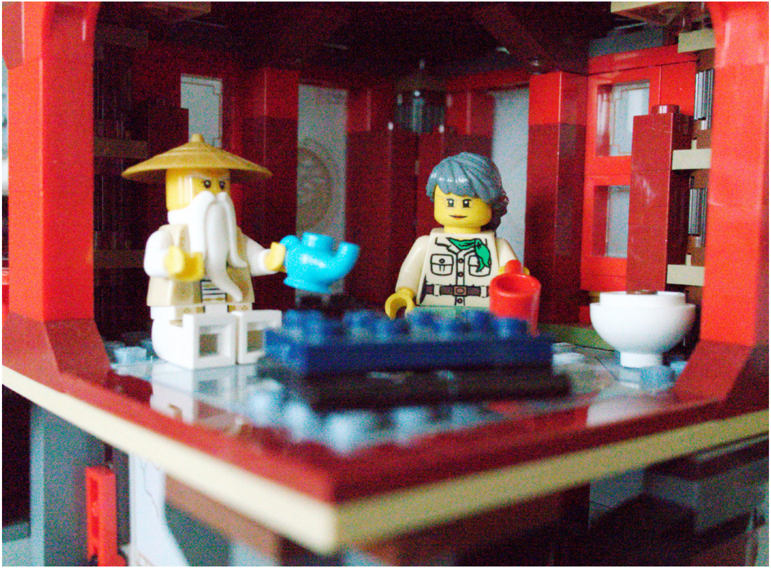
On the use of Gaussian models on patches for image denoising

Antoine Houdard

Young Researchers in Imaging Seminars
Institut Henri Poincaré

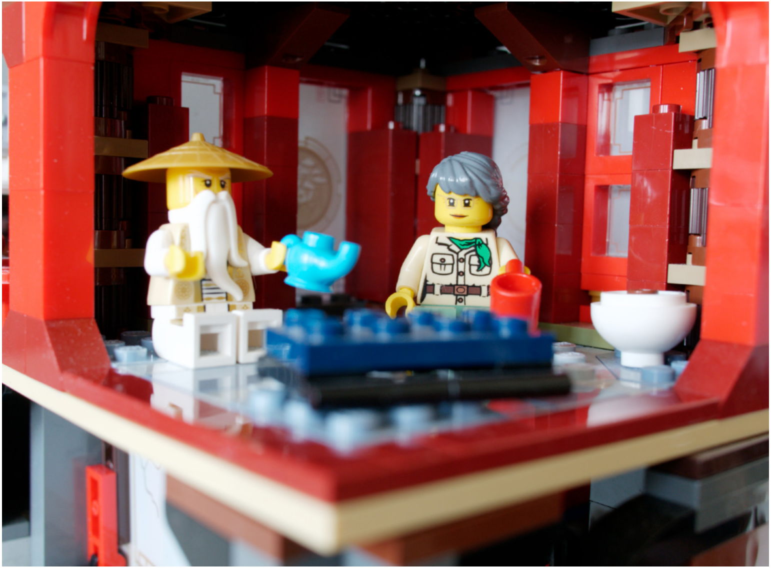
Wednesday, February 27th

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

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

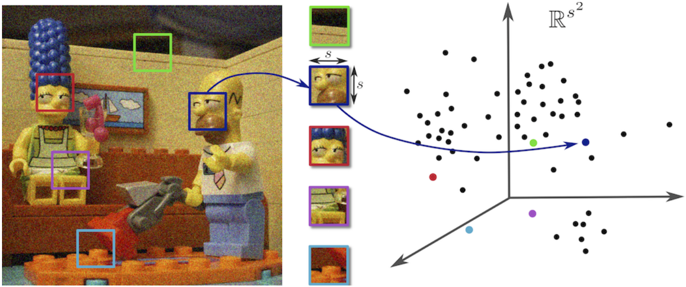
find an estimate

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

Need more realizations or prior information

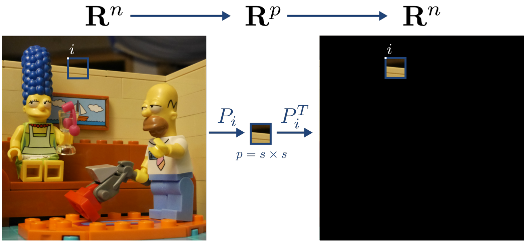
Patch-based image denoising

- 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...



Patch-based image denoising

⋆ Patch extraction operators



⋆ Noise model on the image $\quad V = u + \mathcal{E} \longrightarrow \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$

⋆ Noise model on the patches $\quad P_i V = P_i u + P_i \mathcal{E} \longrightarrow \mathcal{N}(0, P_i \sigma^2 \mathrm{I}_n P_i^T)$

def. def. def.

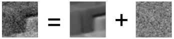
$$Y_i = x_i + N_i \longrightarrow \mathcal{N}(0, \sigma^2 \mathrm{I}_{s^2})$$

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

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:

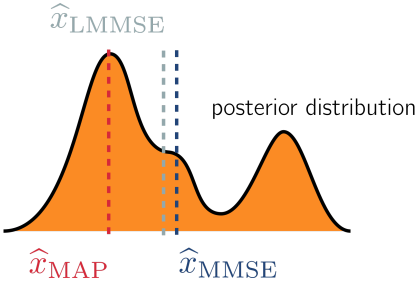


$$Y = X + N \sim \mathcal{N}(0, \sigma^2 \mathbf{I}),$$

then Bayes' theorem yields the posterior distribution:

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

Patch-based image denoising



Denoising strategies

- $\widehat{x} = \mathbf{E}[X|Y=y]$ 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)

Outline

1. Gaussian priors for X: why are they widely used?
2. How to infer parameters in high dimension?
3. Presentation of the HDMI method.
4. Limitations of model-based patch-based approaches.

1. Modeling the clean patches X_i

Choice of the model

In the literature

- local Gaussian models
  - patch-based PCA Deledalle, Salmon, Dalalyan (2011),
  - NL-bayes Lebrun, Buades, Morel (2012),
  - ...
- Gaussian mixture models
  - 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

- Gaussian model

  If $X \sim \mathcal{N}(\mu, \Sigma)$ then

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

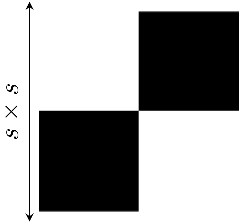
- Gaussian mixture model (GMM)

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

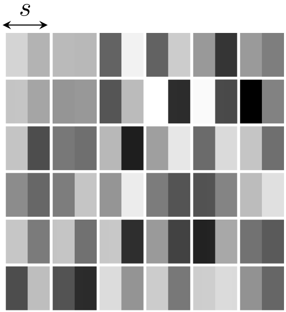
$$\widehat{x}_{\mathsf{MMSE}} = \sum_{k=1}^{K} \mathbb{P}(Z = k | Y = y) \left[\mu_k + \Sigma_k(\Sigma_k + \sigma^2 \mathrm{I})^{-1}(y - \mu_k) \right].$$

What do Gaussian models encode?

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



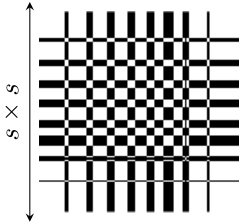
Covariance matrix Σ.



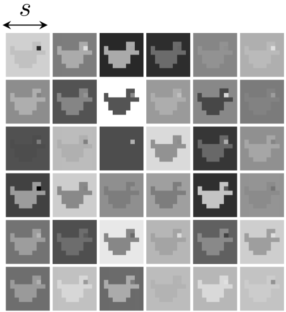
Patches generated from $\mathcal{N}(m, \Sigma)$.

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

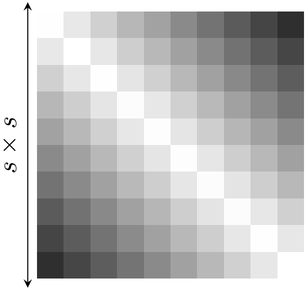
What do Gaussian models encode?

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

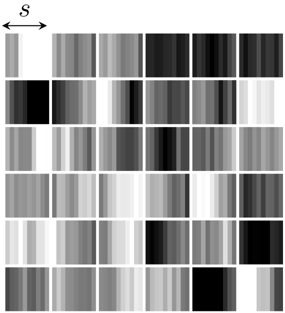
A set of 10000 patches representing edges with random grey levels and random translations.

What do Gaussian models encode?

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

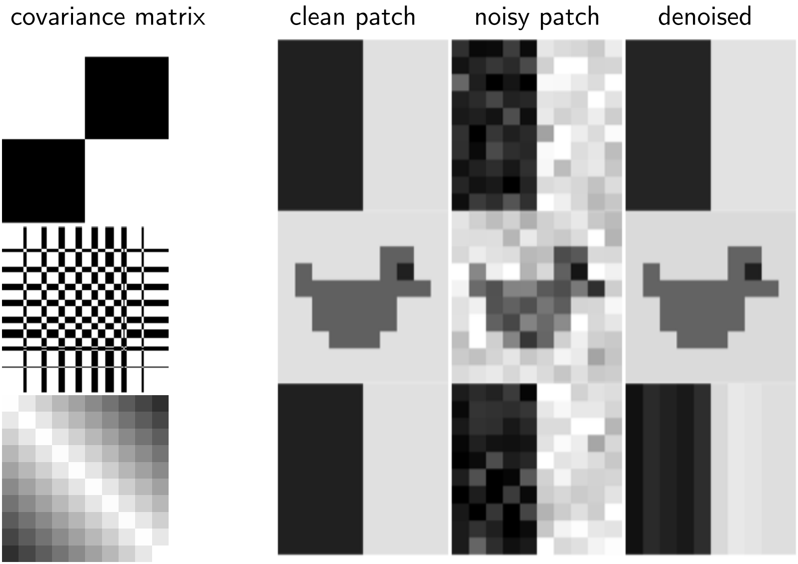


Covariance matrix Σ.



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

Restore with the right model



Conclusion

Modeling the patches with Gaussian models is a good idea:

- They are convenient for computing the estimates;
- They are able to encode the geometric structures of the patches.

Need of good parameters for the model!

2. How to infer parameters in high dimension?

Parameters inference

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

observed data $\{y_1, \dots, y_n\}$ 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)

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

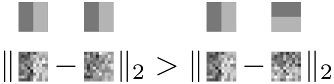
Since $\Sigma_Y = \Sigma_X + \sigma^2 \mathrm{I}_p$, it yields

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

How to group patches?

Need to group the patches representing the same structure together

- For instance with $\| \cdot \|_2$ $\rightarrow$ not robust for strong noise:



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

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 $n_k = \sum_i t_{ik}$.

Sketch of a denoising algorithm

With all these ingredients, we can design a denoising algorithm:

- Extract the patches from the image with P_i operators
- Learn a GMM for the clean patches X from the observations of Y
- Denoise each patch with the MMSE
- Aggregate all the denoised patches with the P_i^T operators

Sketch of a denoising algorithm

With all these ingredients, we can design a denoising algorithm:

- Extract the patches from the image with P_i operators
- Learn a GMM for the clean patches X from the observations of Y
- Denoise each patch with the MMSE
- Aggregate all the denoised patches with the P_i^T operators

But...

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 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?

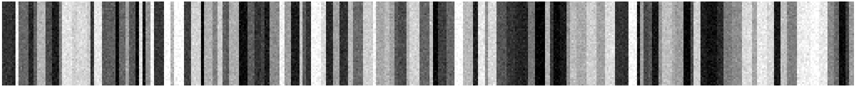
In high-dimensional spaces, it is easier to separate data:

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.

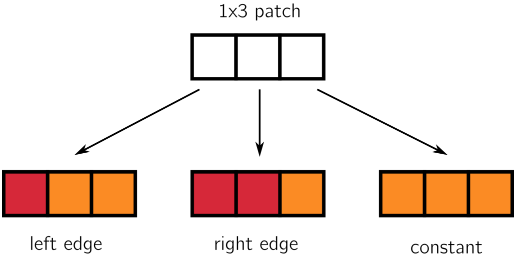
This “bless” is used in clustering algorithms designed for high-dimension
High-Dimensional Data Clustering [Bouveyron, Girard, Schmid] 2007

The bless of dimensionality?

An illustration in the context of patches:

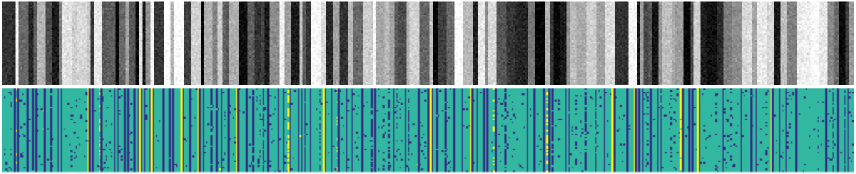


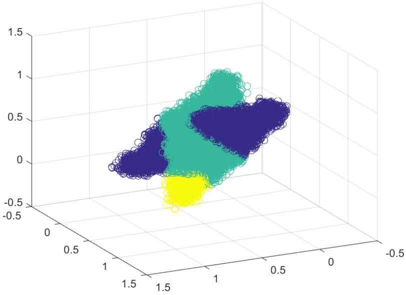
an image made of vertical stripes of width >2 pixels with random grey levels.



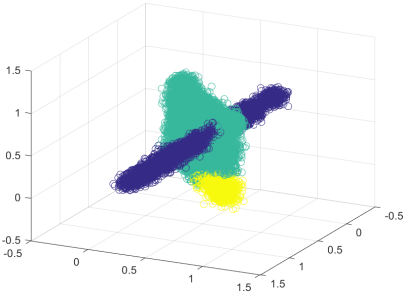
The bless of dimensionality?

An illustration in the context of patches:





view 1

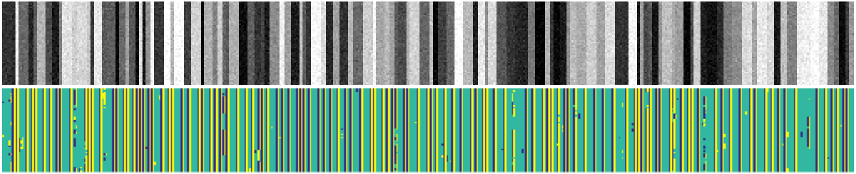


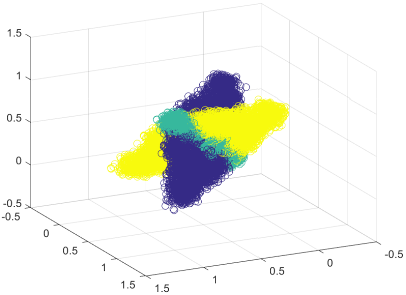
view 2

In the patch space, we cannot distinguish three classes

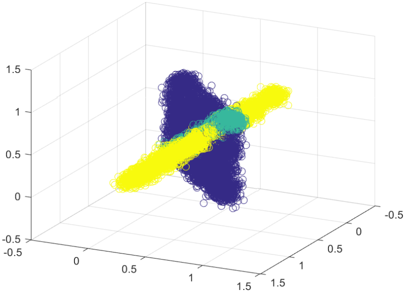
The bless of dimensionality?

An illustration in the context of patches:





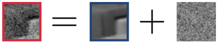
view 1 of the first 3 pixels



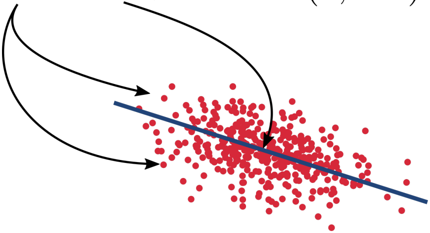
view 2 of the first 3 pixels

The algorithm is now able to separate these classes!

3. 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:

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

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

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

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

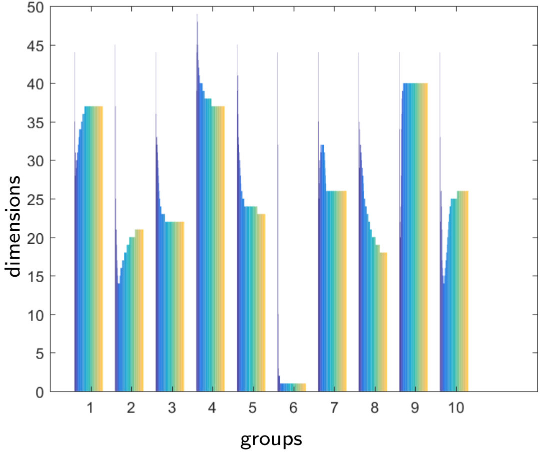
By re-evaluating the dimensions, we change the model at each M-step!

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

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

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!

Clean image

Numerical Experiments

Noisy image $\sigma = 50$

Numerical Experiments

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

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

Numerical Experiments

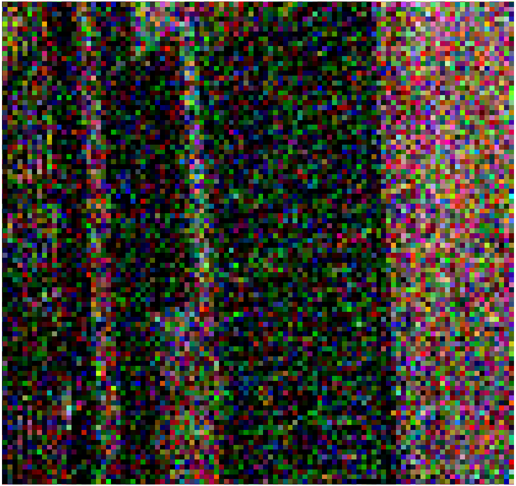
Denoised with HDMI $K = 50$, psnr = 27.28dB

Numerical Experiments – zooms

Clean image

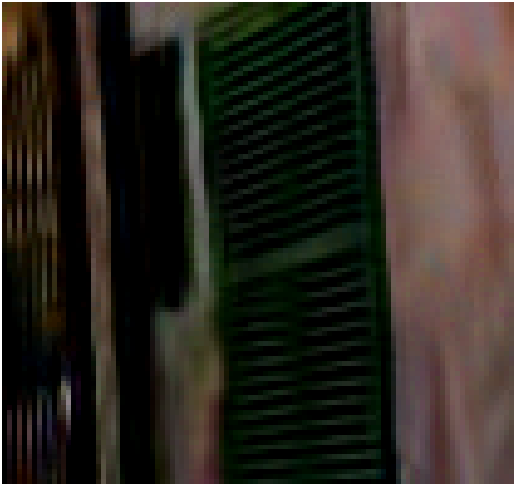
Numerical Experiments – zooms

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

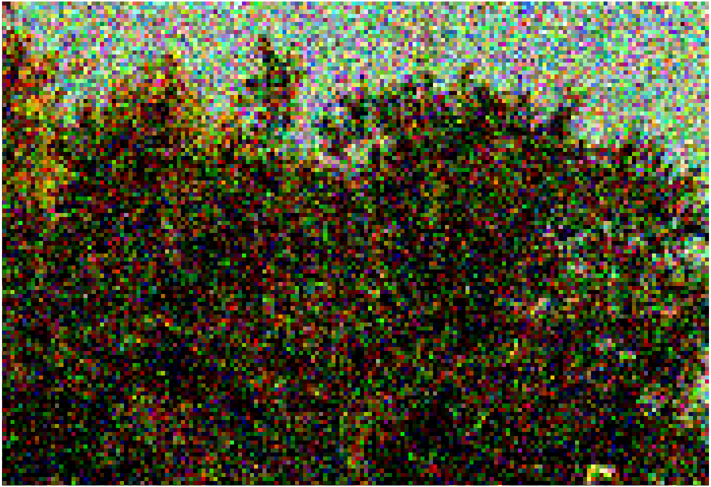
Denoised with HDMI $K = 50$, psnr = 27.05dB

Numerical Experiments – zooms

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

4. 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.

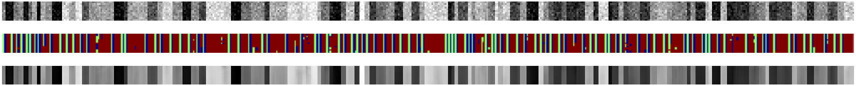
In this context, denoting m_k the number of patches in the k-th group and N the total number of patches, the bound for HDMI is

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

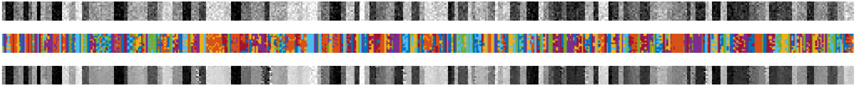
even if the number of samples increases by stretching the image size to infinity, the noise variance cannot be reduced more than a factor p.

The lower bound for patch-based image denoising

HDMI (patches 3×10) - PSNR = 30.12

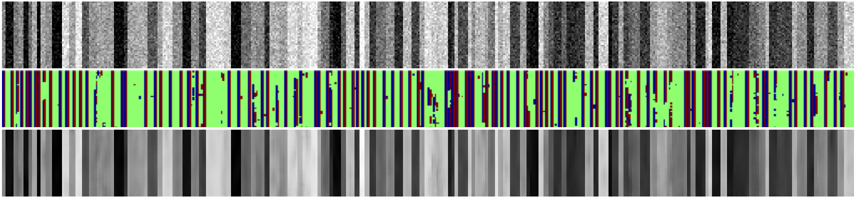


L2 grouping (patches 3×10) - PSNR = 25.03

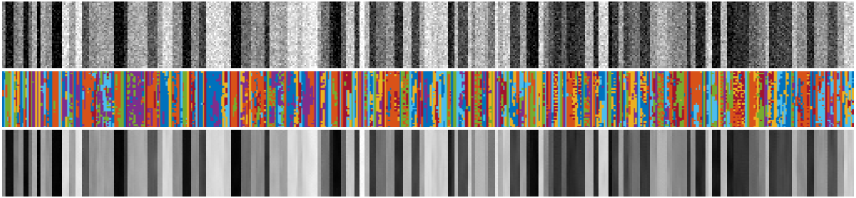


The lower bound for patch-based image denoising

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

- 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.

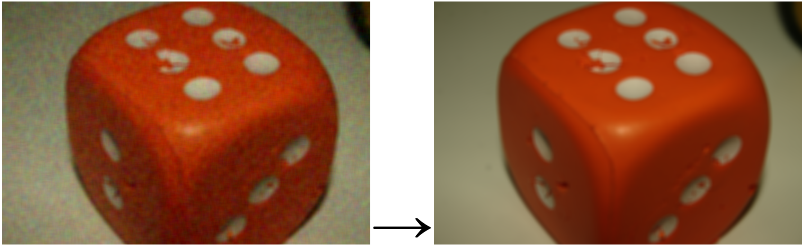
- The noise model can then be divided into the two following problems

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

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

Removing low frequency noise by denoising the DC component

- The DC component can be reshaped as an image



- 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 → can be denoised with the HDMI method

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

Results

Denoised with FFDNet, Zhang et al. 2018, psnr = 36.72dB

Conclusion and future work

We explored model-based patch-based image denoising and we designed the HDMI model that performs state-of-the-art results. This work open several questions and future works:

- Statistical modeling versus deep learning?
  $\longrightarrow$ Statistical modeling is not dead yet! $\rightarrow$ complementary approaches
- Lower-bound for the denoising quality
  $\longrightarrow$ change of paradigm: use the HDMI model in a global way.
- Some miss-classifications when the noise variance is high
  $\longrightarrow$ use of robust estimators such as the geometric median.
- Extension to other image problem
  $\longrightarrow$ missing pixels, inpainting, texture generation.

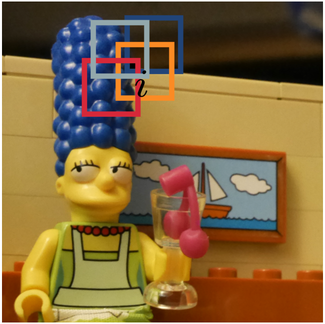
Thank you for your attention!

Any question?

More information on the HDMI model and my new preprint:
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

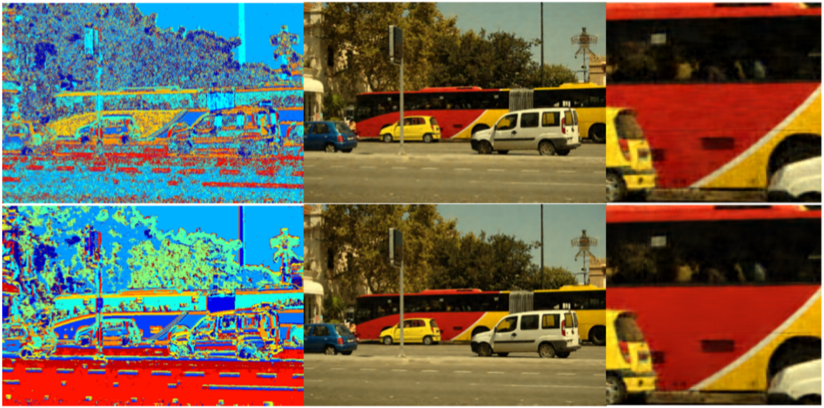
EM is well-adapted for missing data → the model can be easily adapted for missing pixel restoration

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

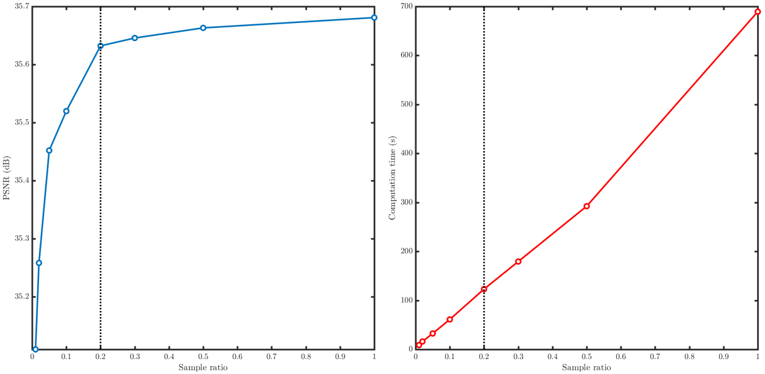


Figure: Effect of the subsampling on the computing time and the denoising performance with HDMI. Left: PSNR versus sampling size. Right: Computation time versus same sampling size. Dotted-lines: 20% subsampling.

Influence of the number of group K

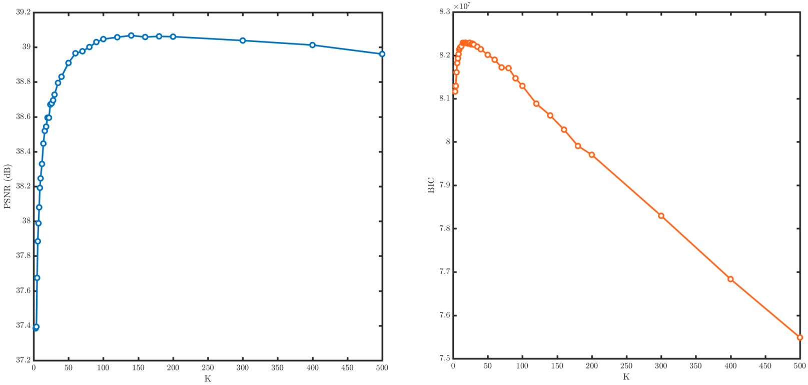


Figure: Denoising results (PSNR) with regard to K (left) and choice of K with BIC (right).

Selection of σ^2 with BIC