How to use Gaussian mixture models on patches for solving image inverse problems

Workshop MixStatSeq

Antoine Houdard

LTCI, Télécom ParisTech
MAP5, Université Paris Descartes

antoine.houdard@telecom-paristech.fr
houdard.wp.imt.fr

Joint work with C. Bouveyron & J. Delon

Image restoration : solving an inverse problem

- **Image restoration problem :**
  find the clean image u from the observed degraded image v s.t.

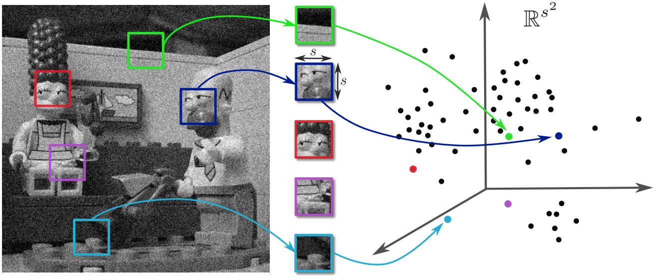
$$v = \Phi u + \epsilon,$$

  with Φ degradation operator and ϵ additive noise.

- **Gaussian white noise case :**
  Here we deal with the simpler problem $\Phi = I$ and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$

Patch-based image denoising

- most of the denoising methods rely on the description of the image by patches (NL-means, NL-Bayes, S-PLE, LDMM, PLE, BM3D, DA3D)



« Les patchs sont aux images ce que les phonèmes sont à la chaîne parlée. »
Pattern Theory, Desolneux & Mumford

Patch-based image denoising

the statistical framework

- We consider each clean patch x_i as a realization of a random vector X_i with some *prior* distribution P_X
- the Gaussian white noise model for patches yields

 =  + 

$$Y_i \quad X_i \quad N_i$$

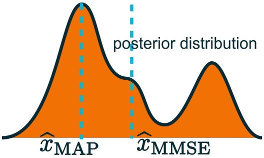
  with $N_i \sim \mathcal{N}(0, I_p)$.

- Hypothesis : N_i and X_i are independent and the N_i's are *i.i.d.*
- so we can write the posterior distribution with Bayes' theorem

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

Patch-based image denoising

denoising strategies



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)

Patch-based image denoising

choice and inference of the model

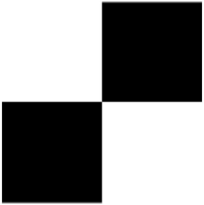
In the literature

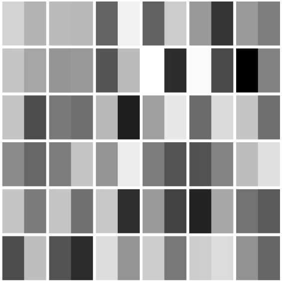
- local Gaussian models [NL-bayes]
- Gaussian mixture models (GMM) [PLE, S-PLE, EPLL]

Advantages of Gaussian models and GMM

- able to encode information of the patches
- make computation of estimators easy

The covariance matrix in Gaussian models and GMM is able to encode geometric structure in patches :



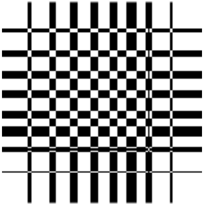


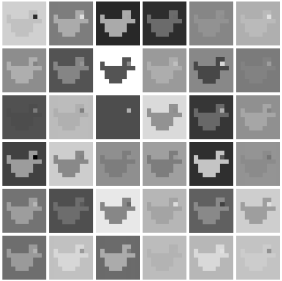
Left : Covariance matrix Σ. Right : patches generated from the Gaussian model $\mathcal{N}(0, \Sigma)$.

Patch-based image denoising

Gaussian and GMM models

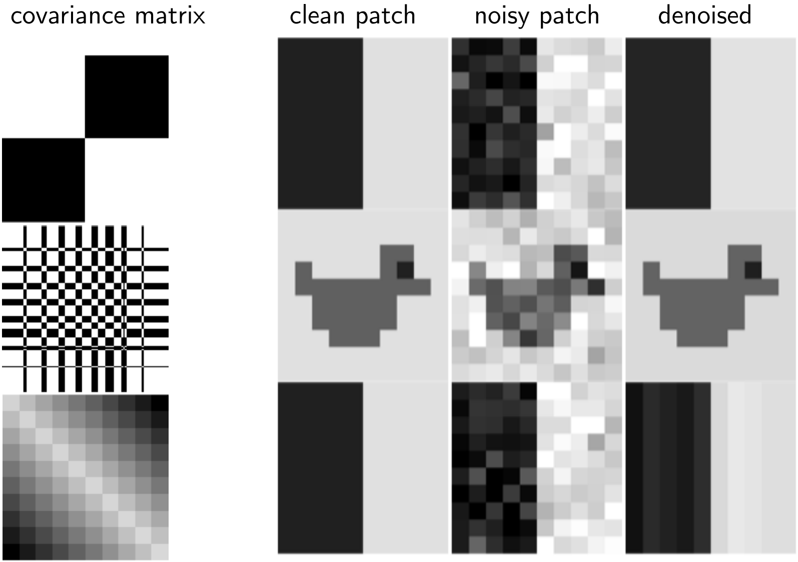
The covariance matrix in Gaussian models and GMM is able to encode geometric structure in patches :





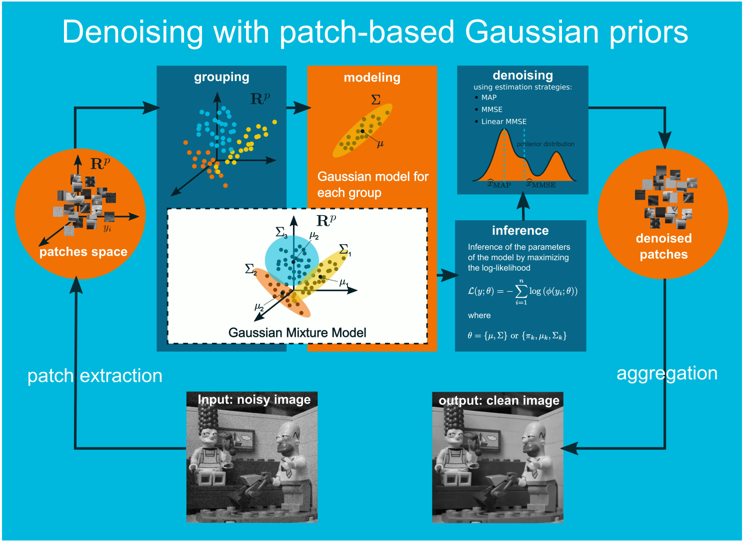
Left : Covariance matrix Σ. Right : patches generated from the Gaussian model $\mathcal{N}(0, \Sigma)$.

Restore with the right model



Patch-based image denoising

summary of the framework



The curse of dimensionality

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

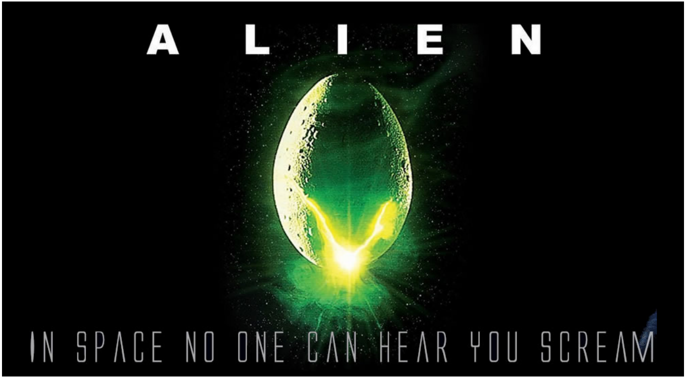
$$\mathbb{R}^{s^2}$$

This term curse was first used by R. Bellman in the introduction of his book "Dynamic programming" in 1957 :

All [problems due to high dimension] *may be subsumed under the heading "the curse of dimensionality". Since this is a curse,* [...], *there is no need to feel discouraged about the possibility of obtaining significant results despite it.*

The curse of dimensionality

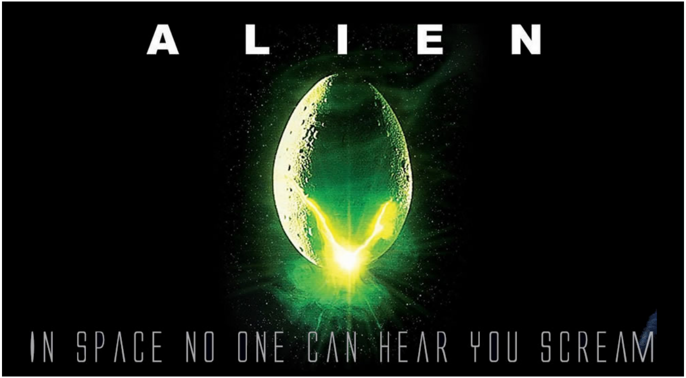
High-dimensional spaces are empty



In high-dimensional space no one can hear you scream!

The curse of dimensionality

High-dimensional spaces are empty



Neighborhoods are no more local !

Data are isolated

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 worked around by

- the use of small patches in NL-Bayes (3×3 or 5×5)
- a model of mixture with fixed lower dimensions covariances in S-PLE

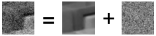
We propose a fully statistical model, that estimates a lower dimension for each group.

Reminder : Noise model and notations

We denote

- $\{y_1, \ldots, y_n\} \in \mathbf{R}^p$ the (observed) noisy patches of the image ;
- $\{x_1, \ldots, x_n\} \in \mathbf{R}^p$ the corresponding (unobserved) clean patches.

We suppose they are realizations of random variables Y and X that follow the classical degradation model :



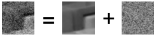
$$Y = X + N \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$

Reminder : Noise model and notations

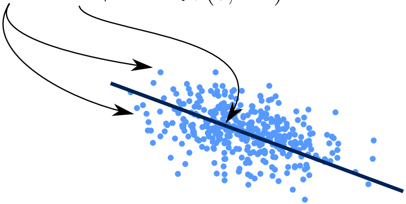
We denote

- $\{y_1, \ldots, y_n\} \in \mathbf{R}^p$ the (observed) noisy patches of the image ;
- $\{x_1, \ldots, x_n\} \in \mathbf{R}^p$ the corresponding (unobserved) clean patches.

We suppose they are realizations of random variables Y and X that follow the classical degradation model :



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



We design for X the High-Dimensional Mixture Model for Image Denoising (HDMI)

The HDMI model

- Model on the actual patches X. Let Z be the latent random variable indicating the group from which the patch X has been generated. We assume that X lives in a low-dimensional subspace which is specific to its latent group :

$$X_{|Z=k} = U_k T + \mu_k,$$

  where U_k is a $p \times d_k$ orthonormal transformation matrix and $T \in \mathbb{R}^{d_k}$ such that

$$T \mid Z = k \sim \mathcal{N}(0, \Lambda_k),$$

  with $\Lambda_k = \mathrm{diag}(\lambda_1^k, \ldots, \lambda_{d_k}^k)$.

- Model on the noisy patches. This implies that Y follow

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

  where π_k is the mixture proportion for the kth component and $\Sigma_k = U_k \Lambda_k U_k^T + \sigma^2 \mathrm{I}_p$.

The HDMI model

The projection of the covariance matrix $\Delta_k = Q_k \Sigma_k Q_k^t$ has the specific structure :

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

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

The HDMI model

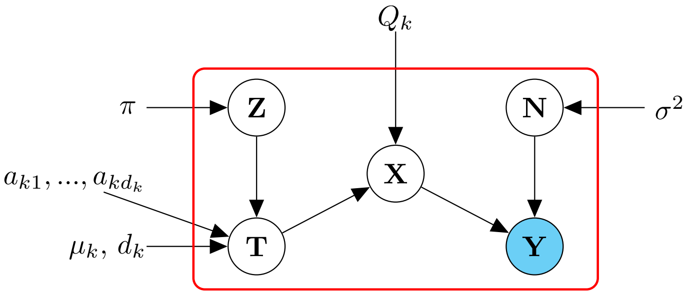


Figure – Graphical representation of the HDMI model.

Denoising with the HDMI model

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

$$\widehat{x_i} = \mathbf{E}[X|Y = y_i],$$

which can be computed as follow :

Proposition.

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

with t_{ik} the posterior probability for the patch y_i to belong in the kth 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

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

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

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 $n_k = \sum_i t_{ik}$. 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 $\widehat{S}_k$.

Model inference

The hyper-parameters

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

We propose to set K at a given value (in the experiments we use $K = 40$ and $K = 90$) and to choose the intrinsic dimensions d_k :

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

Estimation of intrinsic dimensions

when σ is known

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

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

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.

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

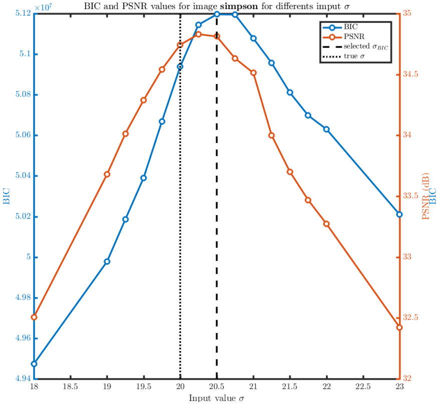
when σ is unknown

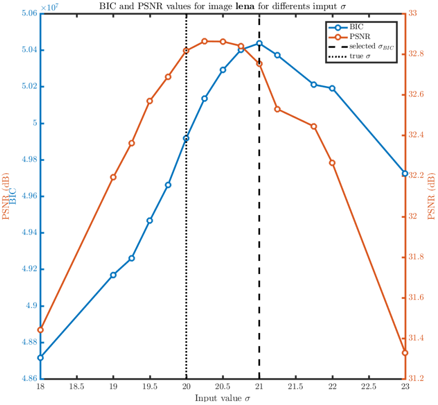
$$\Delta_k = \left(\begin{array}{cc} \begin{array}{|ccc|}\hline a_{k1} & & 0\\ & \ddots & \\ 0 & & a_{kd}\\\hline\end{array} & \mathbf{0} \\ \mathbf{0} & \begin{array}{|ccc|}\hline \sigma^2 & & 0\\ & \ddots & \\ 0 & & \sigma^2\\\hline\end{array}\end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\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

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.

Experiment : selection of σ with BIC

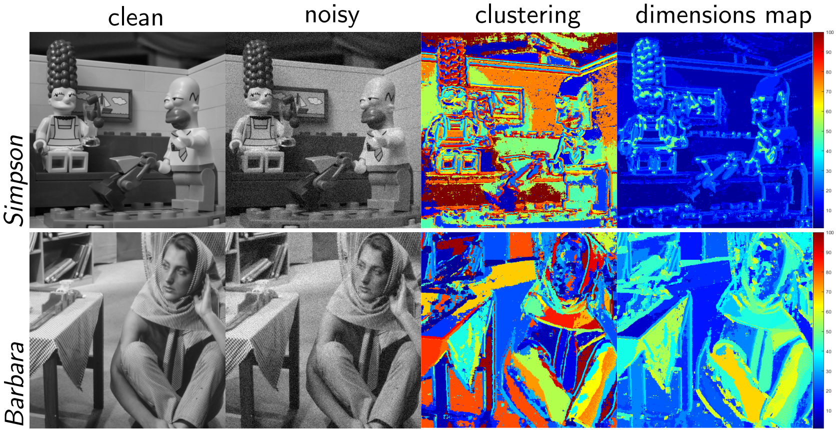




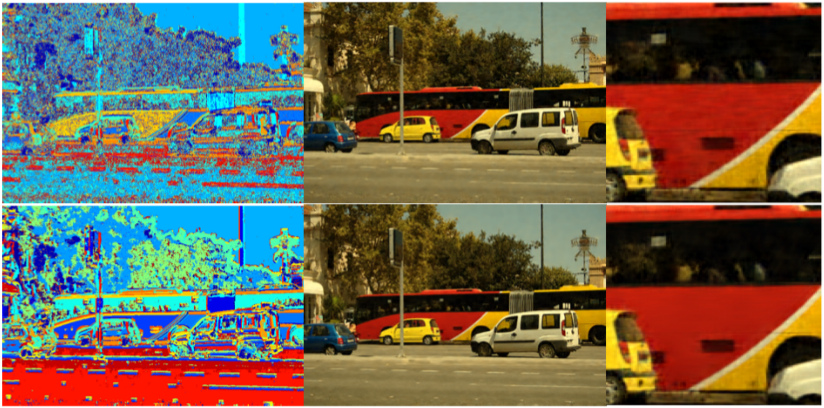
Numerical experiments

Visualization of the intrinsic dimensions

We display for each pixel the dimension of the most probable group of the patch around it.



Regularizing effect of the dimension reduction



Numerical Experiments

Clean image

Numerical Experiments

Noisy image $\sigma = 50$

Numerical Experiments

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 = 27.28dB

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

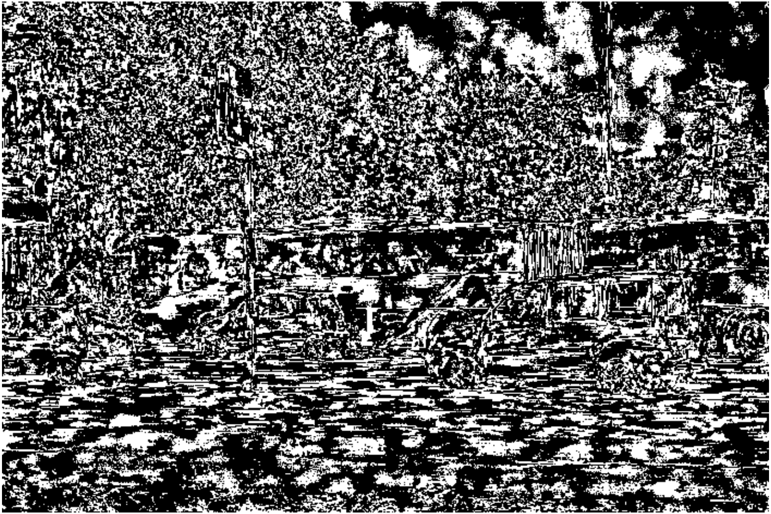
Denoised with FFDNet, Zhang et al. 2018, psnr = 27.45dB

Numerical Experiments

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

Numerical Experiments

PSNR HDMI vs FFDNet



Numerical Experiments

Best of both worlds, psnr = 27.86dB

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 $\rightarrow$ the model can be easily adapted for missing pixel restoration

Conclusion and further work

High dimensional mixtures models for patches

- can model the full process of the generation of the noisy patches ;
- for denoising : can be used *unsupervised* (σ unknown) and reach state-of-the-art performances ;
- not restricted to denoising : interpolation, inpainting, image synthesis ;
- complementary to DL approaches : yield simple image models, easy to interpret ;

Some issues and further work

- high computation time $\rightarrow$ learn the model on a subsample of the patches
- in the case of high σ some miss-classification can yield artifacts $\rightarrow$ explore other initialization ?
- low-frequency noise in flat areas $\rightarrow$ explore aggregation methods (weighted, EPLL) ?

Preprint available at : up5.fr/HDMI

or

houdard.wp.imt.fr/hdmi/

Thank you for your attention !

Any question ?

Preprint available at : up5.fr/HDMI