



**HAL**  
open science

# Clustering en haute dimension pour le débruitage d'image

Antoine Houdard, Charles Bouveyron, Julie Delon

► **To cite this version:**

Antoine Houdard, Charles Bouveyron, Julie Delon. Clustering en haute dimension pour le débruitage d'image. XXVIe colloque GRETSI, Sep 2017, Juan-les-Pins, France. hal-01711999

**HAL Id: hal-01711999**

**<https://hal.science/hal-01711999v1>**

Submitted on 19 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Clustering en haute dimension pour le débruitage d’image

Antoine HOUDARD<sup>1</sup>, Charles BOUYEYRON<sup>2</sup>, Julie DELON<sup>2</sup>

<sup>1</sup>LTCI – Télécom ParisTech  
46 Rue Barrault, 75013 Paris, France

<sup>2</sup>MAP5 – Université Paris Descartes  
45 Rue des Saints-Pères, 75006 Paris, France

antoine.houdard@telecom-paristech.fr

**Résumé** – Dans cet article, nous présentons l’algorithme HDMI de débruitage par patches. HDMI est fondé sur l’apprentissage d’un modèle de mélange de gaussiennes en grande dimension sur l’ensemble des patches bruités. La méthode proposée estime une dimension intrinsèque pour chaque groupe puis les patches débruités sont estimés par une espérance conditionnelle.

**Abstract** – In this work we propose the patch-based denoising algorithm HDMI. It is based on the learning of a probabilistic high-dimensional mixture model on the noisy patches. This model takes into account the lower intrinsic dimension of each group. Finally the restored patches are estimates with a conditional expectation.

## 1 Introduction

**Contexte.** Les méthodes de restauration par patches, très en vogue depuis une quinzaine d’années, consistent à représenter une image par l’ensemble de ses patches d’une taille donnée, puis à tirer parti de la redondance entre ces patches pour restaurer l’image. Ce point de vue a permis des avancées considérables en restauration d’image. Le problème du débruitage par patches possède ainsi une littérature foisonnante, citons les méthodes *NL-means* [2], BM3D [3] puis plus récemment NL-bayes [5] et S-PLE [6], que l’on peut considérer comme faisant parti de l’état-de-l’art dans le domaine.

Une partie des méthodes récentes de débruitage par patches repose sur une écriture bayésienne : chaque patch est restauré en connaissant le modèle de bruit (généralement gaussien *i.i.d.*) et un modèle *a priori* sur la loi du patch sans bruit. Dans la méthode NL-bayes, un modèle gaussien multivarié est estimé séparément pour chaque patch, alors que les méthodes PLE [8] ou S-PLE [6] supposent que les patches de l’image sont des réalisations d’un modèle de mélange de gaussiennes. Ces méthodes sont confrontées à un dilemme : si le modèle de mélange est appris sur une base de données ou généré artificiellement, il est trop peu informatif, mais s’il est appris sur les données du problème, le nombre de paramètres à apprendre est trop élevé par rapport aux nombre de données. Ce problème, connu sous le nom de *fléau de la dimension*, est par exemple résolu dans S-PLE [6] en imposant un rang faible aux matrices de covariance du modèle.

**Contribution.** Dans ce travail, nous proposons d’estimer un modèle statistique complet sur l’ensemble des patches d’une

image à partir de leurs versions bruitées. Le modèle estimé infère pour chaque groupe la dimension de son sous-espace intrinsèque, plus faible que celle de l’espace ambiant. Nous nous inspirons pour cela du *High Dimensional Data Clustering* ou HDDC [1], développé pour le *clustering* de données en haute dimension. Considérons comme données d’entrée l’ensemble des patches bruités  $\{y_1, \dots, y_n\} \in \mathbf{R}^p$  d’une image. Le modèle de dégradation classique de bruit blanc gaussien que nous considérons ici nous permet d’écrire pour chaque patch  $i$  :

$$y_i = x_i + e_i, \quad (1)$$

où les  $x_i$  sont les patches non dégradés et les  $e_i$  sont *i.i.d.* de loi  $\mathcal{N}(0, \sigma^2 I_p)$ . Nous proposons de modéliser les  $x_i$  comme des réalisations *i.i.d.* d’un vecteur aléatoire  $X$ , dont on suppose qu’il suit un modèle de mélange de gaussiennes. En supposant  $X$  indépendant du bruit, ceci implique que les  $y_i$  suivent également un modèle de mélange de gaussiennes. L’inférence de ce modèle est détaillée dans la section 2. Ces hypothèses nous permettent de calculer les espérances conditionnelles  $\mathbf{E}(X_i | Y_i = y_i)$  afin de débruiter les patches dans la section 3. Enfin la section 4 présente les résultats numériques obtenus.

## 2 Modélisation des patches en haute dimension

Dans cette partie, nous commençons par présenter le modèle HDMI – *High-Dimensional Mixture Models for Image Denoising* – dans la section 2.1. Puis nous expliquons brièvement la façon dont nous inférons ce modèle ainsi que la façon dont les dimensions intrinsèques sont estimées dans la section 2.3.

## 2.1 Présentation du modèle HDMI

On fait l'hypothèse dans cet article que les patches non bruités  $\{x_1, \dots, x_n\} \in \mathbf{R}^p$  sont des réalisations d'un vecteur  $X$  qui suit un modèle de mélange de gaussiennes à  $K$  groupes. Comme souligné dans [6], il est raisonnable de penser que la plupart des classes de ce modèle ne devraient pas être de rang plein, et que chaque classe devrait avoir sa dimension spécifique. Soit  $Z$  la variable aléatoire à valeurs dans  $\{1, \dots, K\}$  modélisant l'appartenance de  $X$  aux groupes. Afin de tenir compte de la dimension de chaque classe, on fait l'hypothèse que conditionnellement à  $Z = k$ , le vecteur aléatoire  $X \in \mathbf{R}^p$  s'écrit

$$X_{|Z=k} = U_k T + \mu_k, \quad (2)$$

où  $Q_k$  est une matrice de taille  $p \times d_k$  orthonormale,  $\mu_k \in \mathbf{R}^p$  est la moyenne de la  $k$ -ième classe, et  $T \in \mathbf{R}^{d_k}$  est un vecteur aléatoire supposé gaussien conditionnellement à  $Z$ , de loi

$$T | Z = k \sim \mathcal{N}(0, \Lambda_k), \quad (3)$$

où  $\Lambda_k = \text{diag}(\lambda_1^k, \dots, \lambda_{d_k}^k)$ .

En supposant que le bruit est indépendant de  $X$ , on déduit de ce modèle sur  $X$  un modèle sur notre vecteur bruité  $Y$  :

$$Y_{|Z=k} = U_k T + \mu_k + \varepsilon, \quad (4)$$

où  $\varepsilon \in \mathbf{R}^p$  est un terme de bruit centré gaussien de variance  $\sigma^2$ .

Cela implique que  $Y$  suit un mélange de gaussiennes dont la distribution  $p$  s'écrit

$$p(y) = \sum_{k=1}^K \pi_k g(y; \mu_k, \Sigma_k) \quad (5)$$

où  $\pi_k$  est la proportion de la  $k$ -ième classe,  $g$  est la densité d'une gaussienne multivariée avec  $\Sigma_k = U_k^T \Lambda_k U_k + \sigma^2 I_p$ . En notant  $Q_k = [U_k, R_k]$  la matrice  $p \times p$  formée de  $U_k$  complétée orthonormalement par  $R_k$ , la matrice de covariance projetée  $\Delta_k = Q_k \Sigma_k Q_k^t$  est de la forme suivante :

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

avec  $a_{kj} = \lambda_j^k + \sigma^2$  et  $a_{kj} > \sigma^2$ , pour  $k = 1, \dots, K$  et  $j = 1, \dots, d_k$ . Avec ces notations, le modèle est entièrement paramétré par l'ensemble  $\theta = \{\pi_k, \mu_k, Q_k, a_{kj}, d_k, \sigma; k = 1 \dots K, j = 1 \dots d_k\}$ .

## 2.2 Inférence du modèle

L'inférence du modèle ne peut malheureusement pas être faite directement par maximisation de la vraisemblance puisque

les appartenances aux classes ne sont pas connues. Nous utilisons donc l'algorithme d'espérance-maximisation (EM) afin de maximiser selon  $\theta$  l'espérance conditionnelle du logarithme de la vraisemblance des données complètes :

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

où  $t_{ik} = E[z = k | y_i, \theta^*]$  et  $\theta^*$  est un ensemble donné de paramètres. Partant d'un ensemble de paramètres initial  $\theta^0$ , l'algorithme EM alterne deux étapes. Premièrement, l'étape d'espérance (*E-step*) qui consiste à calculer l'espérance de la log-vraisemblance  $\Psi(\theta, \theta^q)$  conditionnellement à la valeur des paramètres courants  $\theta^q$ . Ensuite, l'étape de maximisation (*M-step*) consiste à maximiser  $\Psi(\theta, \theta^q)$  suivant  $\theta$  afin de mettre à jour l'ensemble des paramètres. Ces deux étapes sont alternées jusqu'à ce qu'un critère d'arrêt soit vérifié. Cet algorithme construit finalement une suite  $(\theta^q)_q$  dont la convergence est garantie vers un maximum local de la vraisemblance [7].

Dans notre cas particulier, les deux étapes E et M sont calculables explicitement. Pour l'étape E, les  $t_{ik}$  se calculent en utilisant la formule de Bayes :

$$t_{ik} = \frac{\hat{\pi}_k g(y_i; \theta_k)}{\sum_{\ell=1}^K \hat{\pi}_\ell g(y_i; \theta_\ell)}. \quad (6)$$

Tandis que pour l'étape M, les paramètres du modèle sont calculés par :

$$\hat{\pi}_k = \frac{n_k}{n} \quad (7)$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_i t_{ik} y_i \quad (8)$$

$$\hat{S}_k = \frac{1}{n_k} \sum_i t_{ik} (y_i - \mu_k)(y_i - \mu_k)^T \quad (9)$$

avec  $n_k = \sum_i t_{ik}$  puis les matrices d'orientation  $\hat{Q}_k$  sont les matrices formées par les  $d_k$  vecteurs propres de  $\hat{S}_k$  correspondants aux  $d_k$  plus grandes valeurs propres,  $\hat{a}_{kj}$  est estimé par la  $j$ -ième plus grande valeur propre de  $S_k$  pour  $j = 1, \dots, d_k$ .

## 2.3 Estimation des dimensions intrinsèques $d_k$

Nous avons vu au paragraphe précédent qu'avec l'algorithme EM nous pouvions inférer le modèle HDMI à condition de connaître les hyper-paramètres  $K$  et  $d_k$ . En effet, le choix de ces paramètres ne peut être fait en maximisant la vraisemblance puisqu'ils contrôlent la complexité du modèle. Cependant, puisque nous avons supposé dans notre problème que les données étaient bruitées par un bruit blanc gaussien de matrice de covariance  $\sigma^2 I_p$  connue, il est possible d'estimer les dimensions  $d_k$ . Nous proposons donc une heuristique dont l'idée est de choisir la dimensions  $d_k$  telle que la moyenne des  $p - d_k$  plus faibles valeurs propres de  $S_k$  soit la plus proche possible de  $\sigma^2$ . Cela revient à résoudre le problème de minimisation :

$$\hat{d}_k = \underset{d}{\text{argmin}} \left| \frac{1}{p-d} \sum_{j=d+1}^p a_{kj} - \sigma^2 \right|. \quad (10)$$

Finalement, à  $K$  fixé, nous disposons d'un modèle statistique de mélange de gaussiennes de dimensions intrinsèques variables pour l'ensemble des patches de notre image qui tient compte du modèle de bruit. Il ne reste plus qu'à choisir ce nombre de groupes  $K$ . Nous proposons en section 4 une discussion sur le choix de cet hyper-paramètre. Finalement nous obtenons une modélisation statistique réaliste de l'ensemble des patches de l'image bruitée. Fort de cela, nous allons voir dans la partie suivante comment utiliser ce modèle pour restaurer l'image.

### 3 Débruitage des patches avec HDMI

Étant données les observations, nous pouvons exprimer l'estimateur statistique  $\hat{X}$  qui minimise en espérance l'erreur quadratique moyenne à  $X$ . Cet estimateur est donné par l'espérance conditionnelle :  $\hat{X} = \mathbf{E}[X|Y]$ . Chaque  $x_i$  est ainsi estimé par  $\hat{x}_i = \mathbf{E}[X|Y = y_i]$ . La proposition suivante donne une expression explicite pour ces estimations :

**Proposition 1.** *Pour  $X$  et  $Y$  définis ci-dessus et en conservant les mêmes notations, on a*

$$\mathbf{E}[X|Y = y_i] = \sum_{k=1}^K \psi_k(y_i) t_{ik}, \quad (11)$$

avec  $t_{ik}$  la probabilité d'appartenance à la classe  $k$  définie en (6), et  $\psi_k$  défini par

$$\psi_k(y_i) = \mu_k + \tilde{Q}_k(I_p - \sigma^2 \Delta_k^{-1}) \tilde{Q}_k^T (y_i - \mu_k), \quad (12)$$

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

Finalement, nous obtenons un ensemble de patches suivant le modèle  $X$  qui minimise l'erreur quadratique moyenne avec les patches sans bruit. Cela nous permet d'obtenir une méthode de restauration d'images bruitées par un bruit blanc gaussien dont on connaît la variance qui ne dépend que d'un seul paramètre d'entrée : la taille des patches. Les grandes étapes de cette méthode sont résumées dans l'algorithme 1.

### 4 Expériences numériques

Nous présentons dans cette partie les expériences numériques illustrant la méthode proposée. Nous commençons par discuter du choix du nombre  $K$  de groupes. Puis nous montrons les performances de la méthode avec  $K$  fixé ainsi qu'une comparaison avec l'état-de-l'art. Toutes les expériences sont faites à taille de patches fixée  $p = 10 \times 10$ .

**Influence du nombre de groupes.** Le tableau 1 présente le PSNR obtenu pour différentes valeurs de  $K$  sur l'image *simpson* bruité à  $\sigma = 10$ . Premièrement nous pouvons constater qu'en utilisant peu de groupes, le débruitage avec le modèle HDMI est déjà relativement satisfaisant. Deuxièmement, ces résultats montrent que le PSNR a tendance à augmenter avec le nombre de classe puisque le modèle est de plus en plus

#### Algorithm 1 Débruitage HDMI

**Entrée**  $u \in \mathbf{R}^{m \times n}$  image bruitée,  $p = s \times s$  taille de patch,  $K$  nombre de classes,  $\sigma^2$  variance du bruit.

**Sortie**  $v \in \mathbf{R}^{m \times n}$  image débruitée.

Extraire  $\{y_1, \dots, y_n\}$  les patches de taille  $p$  de  $u$  ;

**Initialisation** avec k-means.

$dl \leftarrow \infty$ .

**while**  $dl > \epsilon$  **do**

**M step** Mise à jour des paramètres (7), (8), (9) et des dimensions (10).

**E step** Calcul de  $t_{ik}$  (6).

Mise à jour de la log-vraisemblance

$$l = \sum_{i=1}^n \log \sum_{k=1}^K \pi_k p(y_i; \theta_k)$$

et calcul de l'erreur relative avec la vraisemblance de l'itération précédente  $dl = |l - lex|/|l|$ .

$lex \leftarrow l$ .

**end while**

calcul des patches débruités  $\{x_1, \dots, x_n\}$  par l'espérance conditionnelle ;

agrégation des patches  $x_i$  pour obtenir l'image débruitée  $v$ .

$K$	5	10	15	20	30	40
PSNR	37.39	38.19	38.45	38.59	38.72	38.83
$K$	50	70	100	140	200	400
PSNR	38.91	38.97	39.05	<b>39.07</b>	39.06	39.01

TABLE 1 – Performance du débruitage (évalué en PSNR) en fonction du nombre  $K$  de groupes dans HDMI sur l'image *Simpson* avec  $\sigma = 10$ .

précis par rapport aux données jusqu'à un certain point (environ  $K = 140$ ). Enfin, le PSNR décroît lentement lorsque le nombre de classe devient très grand, ceci s'explique par le fait que le nombre de paramètres du modèle explose par rapport au nombre de patches. En conclusion, puisque que le temps de calcul est linéaire en  $K$ , nous proposons d'utiliser un nombre de classes de l'ordre de  $K = 40$  qui est un bon compromis entre temps de calcul et performance puisque HDMI atteint déjà les résultats de l'état-de-l'art dans ce cas.

**Performance de la méthode à  $K$  fixé.** La figure 1 illustre sur l'image *simpson* la dimension des groupes et les types de structures encodées par chaque groupe. Les résultats obtenus sont comparés à l'état-de-l'art dans le tableau 2, pour notre algorithme HDMI nous avons mis deux colonnes : la première donne les résultats pour  $K = 40$  la seconde pour  $K = 90$ . Une visualisation des résultats pour un bruit  $\sigma = 30$  est donnée par la figure 2, nous pouvons constater que pour la plupart des images les textures sont mieux retrouvées par la méthode HDMI, par exemple au niveau de la nappe ou du pantalon dans *barbara* ou des pavés ou des volets dans *alley*. Ces résultats montrent que la méthode HDMI peut rivaliser avec l'état-de-l'art que ce soit en terme de PSNR ou visuellement.

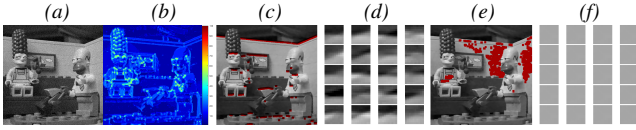


FIGURE 1 – Une illustration de la méthode. (a) l’image *simpson* bruité  $\sigma = 20$ , (b) la carte des dimensions du groupe majoritaire de chaque patch, (c) et (e) l’image débruité avec en rouge les patches d’un même groupe majoritaire de dimensions 13 et 0, (d) et (f) des patches générés par le modèle local de ces groupes.

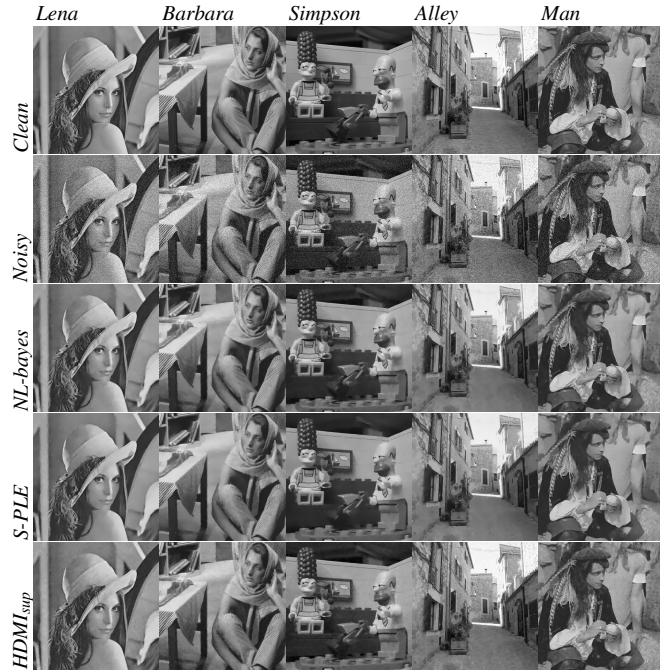


FIGURE 2 – Résultats comparatifs sur *Lena*, *Barbara*, *Simpson*, *Alley* et *Man* avec  $\sigma = 30$ . Pour chaque colonne, de haut en bas : l’image originale, l’image bruitée, NL-Bayes [5], SURE-PLE [6], HDMI. Les images doivent être comparées à pleine résolution sur la version électronique de l’article.

## 5 Conclusion

Nous avons présenté un algorithme de débruitage d’image entièrement fondé sur l’apprentissage d’un modèle statistique sur l’ensemble de ses patches. Nous avons mis en évidence les performances de cette méthode qui ne nécessite que peu de réglages de paramètres. En effet l’utilisateur ne doit fournir en entrée que trois données : la variance  $\sigma$  du bruit, la taille des patches et le nombre de groupe  $K$ . Notons que nous avons présenté ici la version supervisée de HDMI et que nous proposons dans [4] une version ne nécessitant par la connaissance de  $\sigma$ .

Pour finir, notons que le temps de calcul pour l’apprentissage du modèle sur l’ensemble des patches d’une image est relativement long (environ 12 minutes pour une image  $512 \times 512$ ) sur un ordinateur mono-CPU. Cependant, ce temps de calcul peut diminuer considérablement si l’on apprend le modèle que sur un sous-ensemble des patches de l’image. Les probabilités a posteriori sont ensuite prédites pour tous les patches en vue du débruitage. La figure 3 montre qu’il est tout à fait envisageable d’apprendre le modèle sur 20% des patches ; le temps de calcul est alors d’environ 2 minutes.

TABLE 2 – Comparaison des résultats obtenus par NLBayes, SURE-PLE et HDMI pour  $K = 40$  et  $K = 90$ . Exprimés en PSNR (dB).

Image	$\sigma$	NLBayes	S-PLE	HDMI <sub>sup</sub>	
<i>Lena</i>	10	35.79	35.50	35.78	<b>35.83</b>
	20	32.86	32.58	32.82	<b>32.90</b>
	30	<b>31.19</b>	30.75	30.99	31.04
<i>Barbara</i>	10	34.91	34.21	34.77	<b>35.01</b>
	20	31.51	30.67	31.32	<b>31.61</b>
	30	<b>29.62</b>	28.47	29.31	29.49
<i>Simpson</i>	10	38.67	38.37	<b>38.80</b>	<b>38.98</b>
	20	34.65	34.21	<b>34.74</b>	<b>34.91</b>
	30	32.21	31.44	<b>32.33</b>	<b>32.50</b>
<i>Alley</i>	10	32.45	32.14	32.40	<b>32.47</b>
	20	28.90	28.57	<b>29.03</b>	<b>29.07</b>
	30	26.89	26.61	<b>27.31</b>	<b>27.39</b>
<i>Man</i>	10	<b>34.07</b>	33.76	33.85	33.91
	20	<b>30.63</b>	30.31	30.44	30.47
	30	<b>28.81</b>	28.47	28.65	28.71

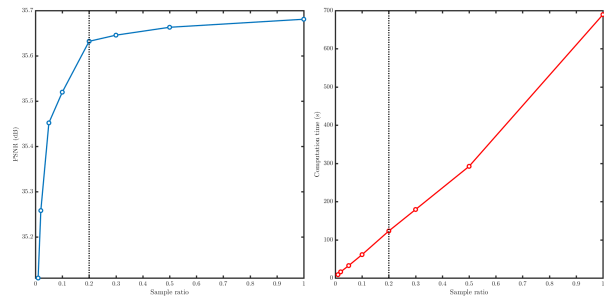


FIGURE 3 – *Gauche* : évolution du PSNR en fonction du taux d’échantillonnage. *Droite* : évolution du temps de calcul (s) en fonction du taux d’échantillonnage. La ligne verticale correspond à 20% des patches.

## Références

- [1] BOUYEYRON, C., GIRARD, S., & SCHMID, C., *High-dimensional data clustering*, Computational Statistics & Data Analysis, 2007.
- [2] BUADES, A., COLL, BARTOMEU, & MOREL, J. M., *A non-local algorithm for image denoising*, proceedings of CVPR’05, pp 60-65, 2005.
- [3] DABOV, K., FOI, A., KATKOVNIK, V. & EGIAZARIAN, K., *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.
- [4] HOUDARD, A., BOUYEYRON, C. & DELON, J., *High-Dimensional Mixture Models for Unsupervised Image Denoising (HDMI)*, preprint <https://hal.archives-ouvertes.fr/hal-01544249>, 2017
- [5] LEBRUN, M., BUADES, A., & MOREL, J. M., *A nonlocal bayesian image denoising algorithm*, SIAM Journal on Imaging Sciences, 2013.
- [6] WANG, Y. Q., & MOREL, J. M., *SURE guided Gaussian mixture image denoising*, SIAM Journal on Imaging Sciences, 2013.
- [7] WU, C.F., *On the convergence properties of the EM algorithm*, The Annals of Statistics, vol. 11, n. 1, pp. 95–103, 1983.
- [8] YU, G., SAPIRO, G. & MALLAT, S., *Solving inverse problems with piecewise linear estimators : from Gaussian mixture models to structured sparsity*, IEEE Transactions on Image Processing, vol. 21, n. 5, pp. 2481–99, 2012.