Projet UE TADI

AMO – Apprentissage et application de l'apprentissage de dictionnaires parcimonieux en traitement d'images

Diana Mandache - Yanis Chemli - Iyed Trimèche
UPMC/Telecom ParisTech

Résumé Ce document présente et synthétise les résultats des travaux menés lors des TP du module AMO. Ils ont consisté dans un premier temps en l'apprentissage de dictionnaires pour la représentation d'images naturelles. Il a ensuite été proposé dans un deuxième temps de mettre en application les résultats d'apprentissages de dictionnaires parcimonieux à des problèmes classiques tel que le débruitage ou bien l'inpainting.

On a donc cherché à décomposer de manière parcimonieuse des patchs d'images, issues de la base « Kyoto », sur un ensemble redondant d'éléments de base, le but étant de modéliser chaque patch par une combinaison linéaire des éléments du dictionnaire visuel.

I Génération d'une base de patchs

Une étape préalable est l'extraction d'une base de patch de notre ensemble d'image.

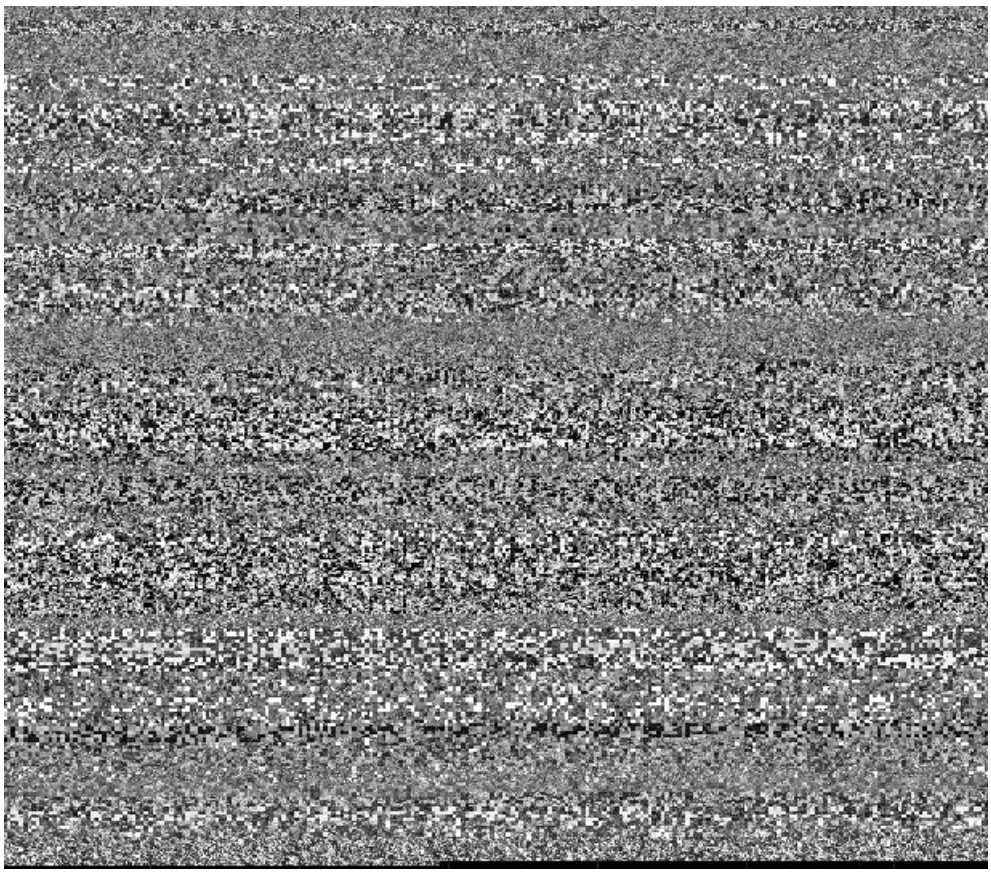

Figure 1Extraction des patches de la base "Kyoto"

Nous avons donc extrait tous les patchs de la base (55800) que nous avons ensuite centré et normalisé afin de travailler sur une même échelle de paramètres. Ce sont ces patchs que nous décomposerons de manière parcimonieuse.

II Apprentissage de dictionnaire parcimonieux

1) Représentation parcimonieuse

Avant d'aborder les aspects mis en pratique lors du TP, nous présentons dans la suite certains aspects théorique sur la représentation parcimonieuse des signaux.

Les techniques usuels de représentation des signaux (à n dimensions) utilisent en général une description des composantes sur une base orthonormée pour laquelle la représentation du signal est unique. La décomposition en séries de Fourrier est basée sur cette approche par exemple.

Un signal y de taille N peut être représenté sur une base orthonormale B par la projection de ses composantes x sur B :

$$y = Bx$$

B étant une base, il y unicité de la décomposition de y sur cette base.
On a donc directement :

$$x = B^{-1}y$$

Dans le cas de la représentation parcimonieuse, on cherche à décomposer notre signal sur un dictionnaire redondant, qui contient un nombre d'atomes supérieur à la dimension du signal. Cette décomposition introduit dans le nouvel espace de représentation un grand nombre de valeurs nuls.
Un signal y de taille N peut alors être représenté sur un dictionnaire redondant D (ou famille génératrice liée) de K vecteurs de taille N (K>>N) par :

$$y = Dx$$

Lorsque l'on cherche une représentation approchée su signal de départ, on utilise un critère de différence entre le signal et sa représentation. On va en pratique chercher à minimiser la quantité :

$$||y - Dx||^2$$

2) Apprentissage de dictionnaire parcimonieux

Le problème de l'apprentissage de dictionnaire parcimonieux revient donc à optimiser sous forme matricielle :

$$min_{D,X}\left\{||Y - DY||^2\right\} \; avec\ la\ contrainte : \; \forall \, i \, \|x_i\|_0 \leq T_0$$

Avec :
- Y (de taille n x N patch) la matrice des données d'entrée regroupant dans ses vecteurs colonne chaque patch.
- D (de taille n x K) la matrice des éléments du dictionnaire qu l'on cherche à apprendre
- X (de taille K x N) la matrice de projection de yk sur D.

En pratique, l'optimisation est réalisée par la version optimisée de l'algorithme K-SVD. Cet algorithme consiste en une optimisation alternée du problème en considérant D fixé puis X fixé. Le problème peut être réécrit de la façon suivante :

$$f(D) = \|Y - DX\|_F^2 = \left\| Y - \sum_{j=1}^{K} d_j x_T^j \right\|_F^2 = \left\| \left(Y - \sum_{j \neq k}^{K} d_j x_T^j \right) - d_k x_T^k \right\|_F^2 = \|E_k - d_k x_T^k\|_F^2$$

La première étape de l'algorithme de sparse coding a été fournie. C'est dans cette étape où sont calculés les projections des données d'entrés sur le dictionnaire.

La deuxième étape, que nous avons mis en œuvre, consiste en la mise à jour du dictionnaire qui a préalablement été initialisé par des patchs sélectionnés aléatoirement.

De façon globale, la méthode prend en entrée la matrice des données Y (196 x 20 000 dans notre cas), le dictionnaire initial (196 x 441) et apprendra en 40 itérations.

Nous avons donc dans un premier temps effectué des premiers tests sur un exemple jouet. Le but a été de générer des données de synthèses en utilisant un dictionnaire avec une parcimonie k = 2, d'apprendre un nouveau dictionnaire à partir des données générées et enfin, de comparer le dictionnaire appris au dictionnaire initial.

On génère 1326 exemples et nous appelons la fonction K-SVD mise en place pour apprendre le dictionnaire sur quelques de ces données sélectionnées aléatoirement.

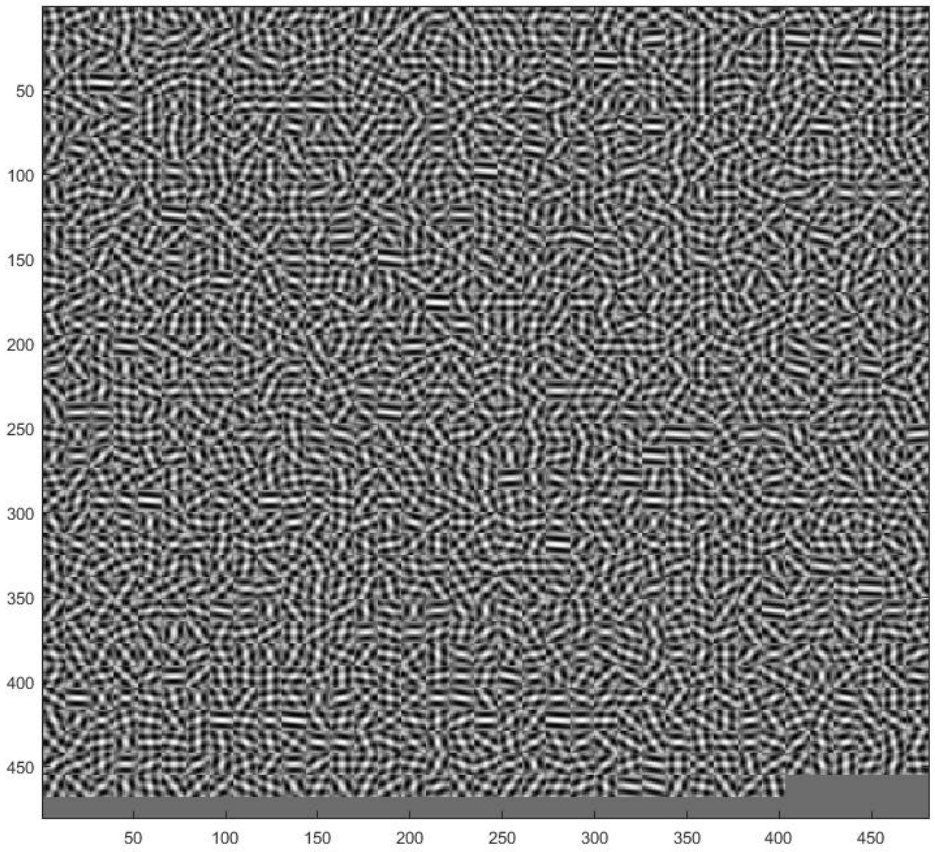

Figure 2 Données Y générées à partir du dictionnaire "Gabor"

Nous imposons 100 itérations et estimons le dictionnaire suivant (1er ligne) :

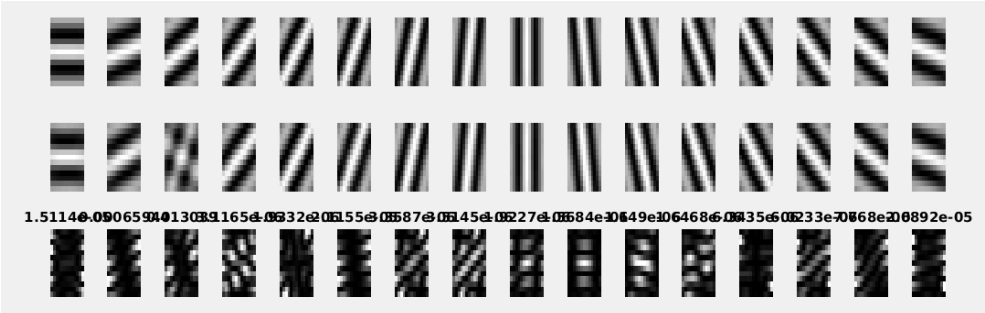

Figure 3 Comparaison des élements des dictionaires estimé et vérité

L'erreur entre le dictionnaire estimé et le dictionnaire ayant servie à la génération des données est donné par la courbe suivante :

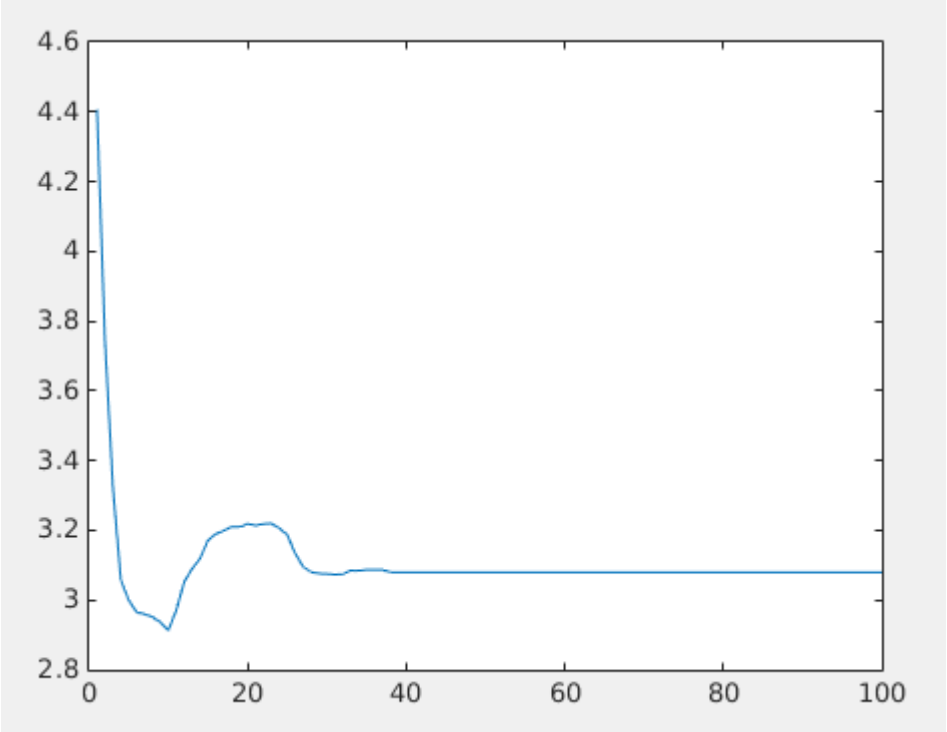

Figure 4 Courbe d'erreur d'estimation du dictionnaire au cours des itérations du K-SVD

On observe qu'à partir d'une trentaine d'itérations, une stabilisation de l'erreur qui, quantitativement parlant met lumière le fait que le nouveau dictionnaire est proche du dictionnaire initial.

Il est a noté que connaissant le niveau de parcimonie réel, nous ne sommes pas assurés de la convergence de l'algorithme vers une solution optimale car nous avons opéré une représentation parcimonieuse approchée des données qui dépend d'un seuil mais surtout d'une initialisation aléatoire du dictionnaire. Pour limiter ce problème on fera plusieurs initialisations et nous prendrons celle qui fait intervenir l'erreur minimale.

III K-SVD Application sur des données réelles

Nous avons dans la suite appliqué le même principe que précédemment mais sur des images réelles. Nous avons appris par l'algorithme K-SVD appliqué sur des patchs issus de la base « Kyoto », un dictionnaire.

Linéairement, nous avons chargé la base des 55 800 patchs centres et normalisé. Nous avons ensuite appliqué K-SVD sur 20 000 patchs aléatoires et imposé un niveau de parcimonie de 10 et 40 itérations. Le résultat du dictionnaire appris est le suivant :

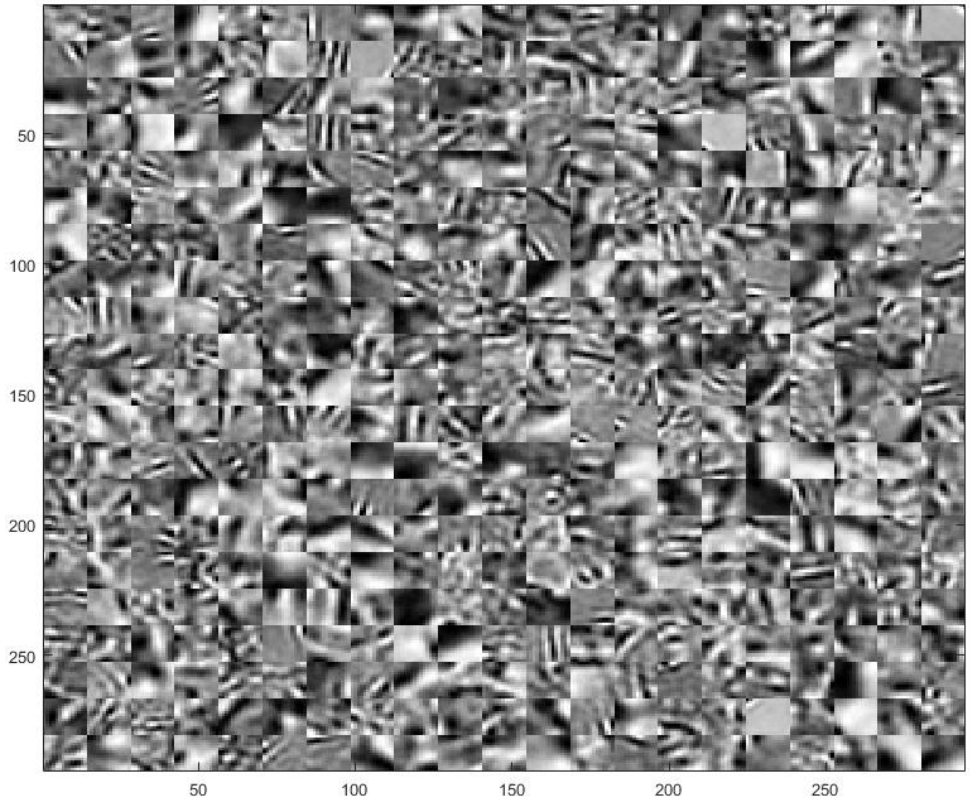

Figure 5 Dictionnaire appris à partir de la base "Kyoto"

IV Applications

Dans cette partie, nous utilisons les méthodes d'apprentissage parcimonieux pour des applications de débruitage et d'inpainting.

1) Reconstruction d'une image à partir de patchs

Dans cette partie du TP, il a été proposé de reconstruire une image de la base Kyoto avec différents dictionnaires. Nous avons utilisé le dictionnaire appris précédemment, un dictionnaire correspondant à la transformée en cosinus discret redondante et dictionnaire appris sur l'image à reconstruire.

Figure 6 Exemple d'une image à reconstruire à partir du dictionnaire

Ainsi, sur une image, nous avons extrait des patchs selon une méthode de fenêtre glissante. Ces patchs sont centrés et normalisés. Sur ces patchs, nous apprenons un nouveau dictionnaire spécifique à l'image. Nous imposons 5 itérations et un niveau de parcimonie de 10.

Le but est alors de reconstruire l'image originale avec les trois dictionnaires. Pour cela, on calcule X, la matrice de projection des données. A partir de cette matrice, on reconstruit les patchs pr comme étant égale à DX.

Voici les résultats de reconstruction obtenues pour les différents dictionnaires sur différentes images :

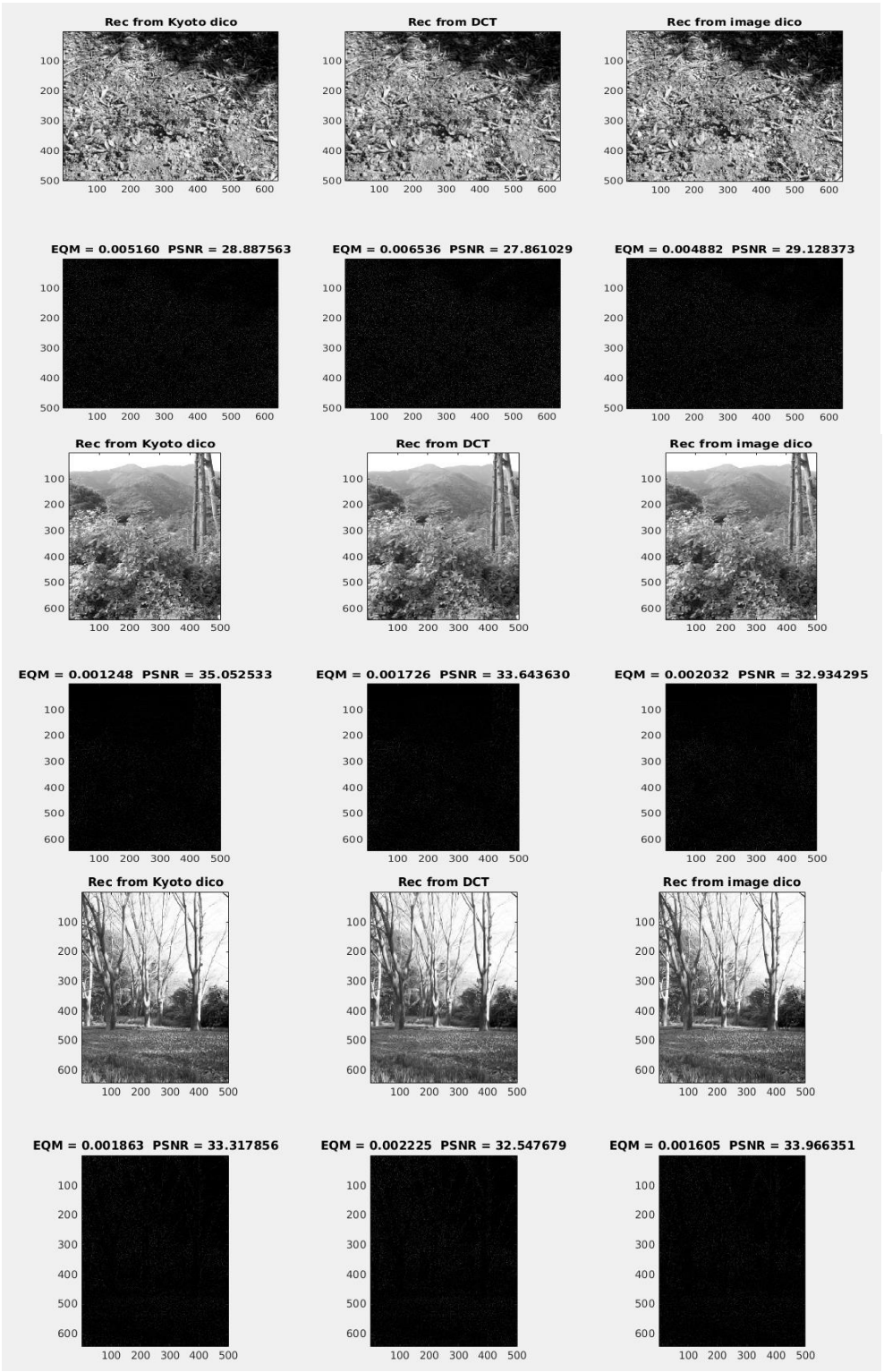

Figure 7 Des exemples d'images reconstruites à partir de plusieurs dictionnaires appris

Pour comparer les résultats de reconstruction, nous avons utilisé la métrique PSNR qui rend compte de la distorsion entre des images. Il mesure la proximité de l'image reconstruite par rapport à l'original au niveau du signal (, il ne prend pas en compte la qualité visuelle de reconstruction).

Ainsi on note que les qualités de reconstruction selon les dictionnaires sont proches. Cependant, il semble que la reconstruction par DCT soit la moins fidèle à l'image originale (PSNR plus faible). Mais cela ne semble pas être une vérité absolue. Nous avons synthétisé les résultats pour plus images (10) dans le graphique suivant :

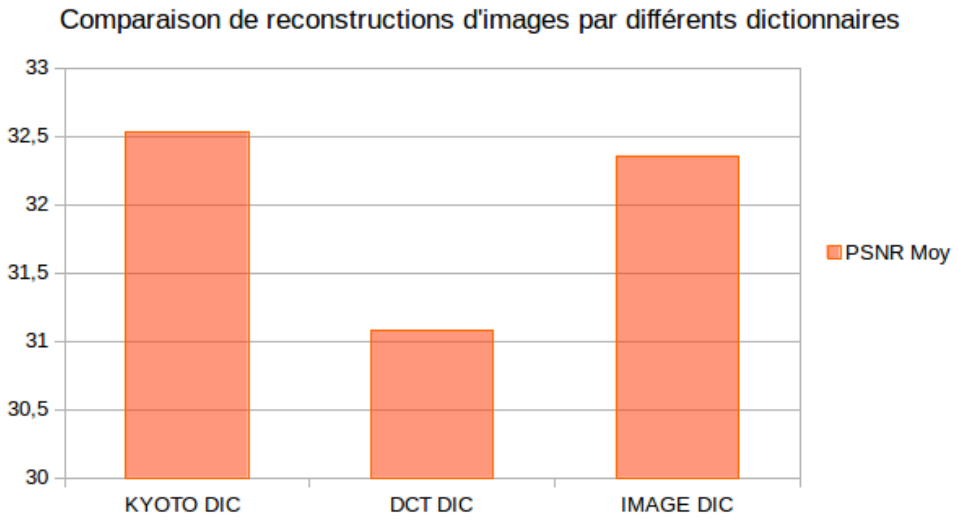

Figure 8 Synthèse des PSNR sur 10 images

2) Débruitage d'images

Nous avons par la suite appliqué la méthode au débruitage d'images. De la même façon, nous allons apprendre un dictionnaire parcimonieux spécifique à l'image à traiter. Nous avons calculé un dictionnaire de 26 élément, avec un niveau de parcimonie de 5. Les patches d'apprentissage (de taille 8) seront générés comme précédemment par une fenêtre glissante.

L'idée est d'apprendre le dictionnaire sur une image non bruitée et d'utiliser ce dictionnaire pour débruiter chacun des patches bruités (le bruit, gaussien, est ici maîtrisé afin d'effectuer des mesures qualitatives). On cherche alors une erreur de reconstruction et non pas une reconstruction à proprement

parlé.

La fonction de débruitage, par une opération de sparse coding, calcule la matrice de projection X satisfaisant l'équation :

$$\hat{x_i} = \arg\min_{x_i} ||x_i||_0 \quad \text{tel que :} \quad ||Y - DX||_F^2 \leq T_0$$

Le but est ensuite de reconstruire par moyennage les patches reconstruits.

Ainsi, pour l'image « Lena » on a le résultat suivant :

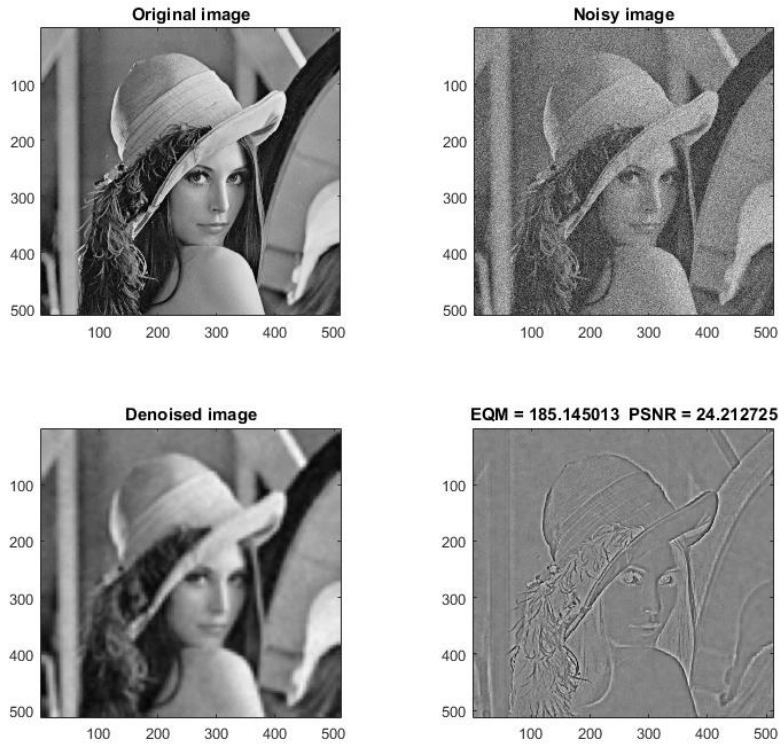

Figure 9 Débruitage de Lena

L'erreur mesuré par le PSNR est de 24,2. On observe une image bien débruité avec un léger flou. On peut comparer à un filtrage par un noyaud gaussien :

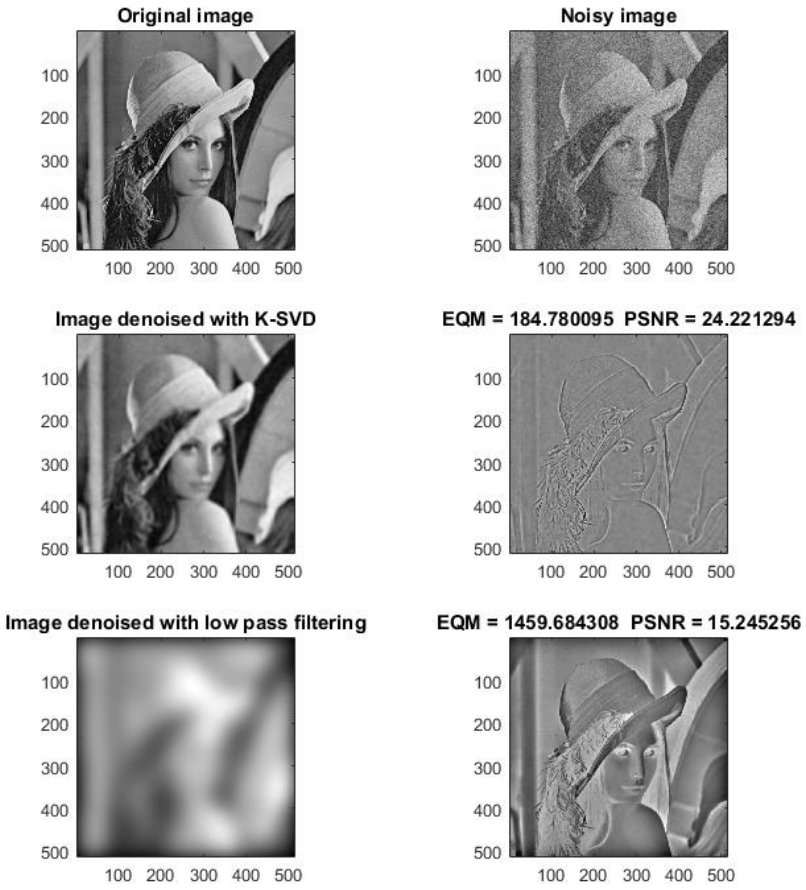

Figure 10 Comparaison avec un débruitage par noyau Gaussien

On note que le flou est beaucoup plus présent et que l'on perd en contraste. On a ces résultats car le filtre gaussien agis comme un filtre moyenneur (pondéré).

Nous avons ensuite appris un dictionnaire directement sur l'image bruité. On a comparé visuellement le dictionnaire généré sur l'image non bruité et celui généré sur l'image bruité :

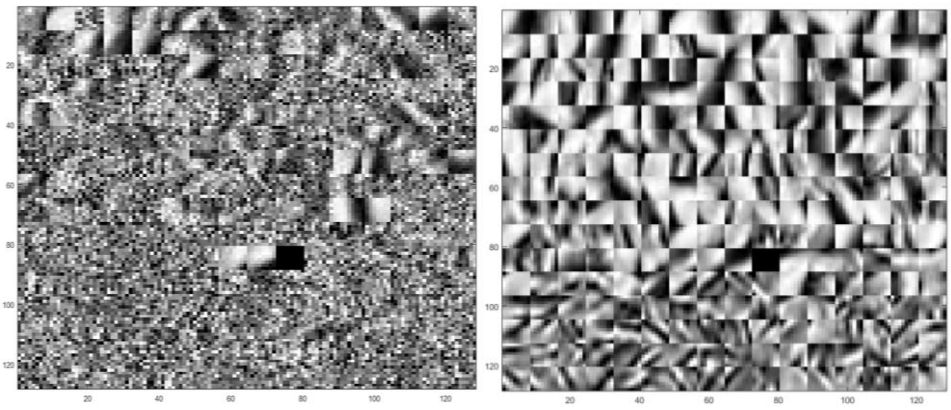

Figure 11 A gauche le dictionnaire de Lena bruité à droite celui de Lena non bruité

Les deux dictionnaires sont visuellement très différents mais on observe que certaines similarités au niveau de quelque composantes.

Le débruitage consiste alors à minimiser l'équation précédente.

On obtient ces résultats de débruitage suivant :

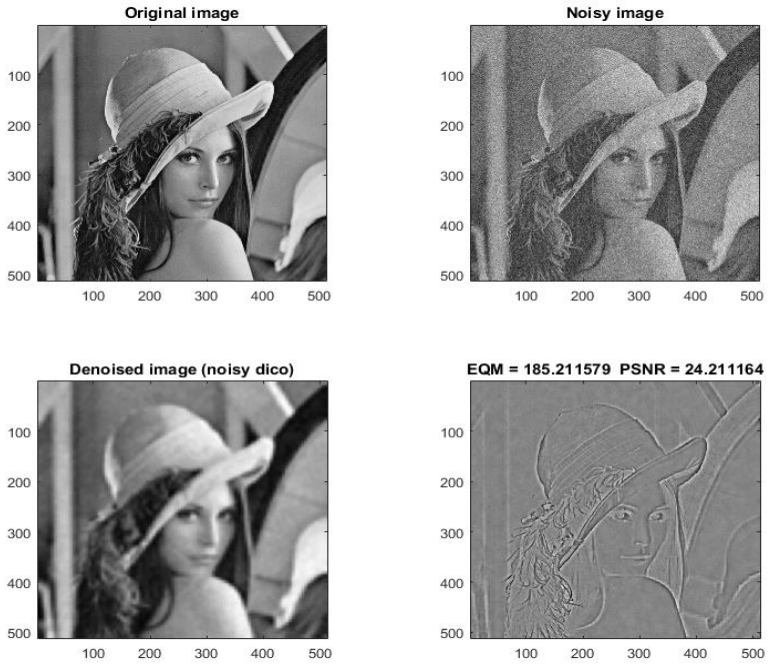

Figure 12 Débruitage de Lena à l'aide d'un dictionnaire appris sur l'image bruitée

Malgré une différence visuelle entre les deux dictionnaires le débruitage semble correcte par rapport au précédent. Le PSNR est de 24,2 quasiment similaire à ce qu'on a calculé. Il est donc pertinent d'apprendre le dictionnaire directement sur l'image bruitée.

3) Inpainting

Dans cette partie, il a été proposé d'utiliser le dictionnaire parcimonieux pour reconstruire des régions d'images où l'information est manquante. Nous effectuons l'apprentissage sur des données saine afin de reconstruire des données (patchs) corrompues. Le problème revient à minimiser la formule suivante :

$$\min_{x_i} \left\{ ||(y_i \circ M) - (D \circ M)x_i||_2^2 \right\} \quad \textbf{tel que :} \quad ||x_i||_0 \leq T_0$$

Le but est alors d'annuler toutes les ligne du dictionnaire correspondante aux composantes où l'information est manquante.

Nous apprenons un dictionnaire sur l'image saine suivant :

Figure 13 Image saine et dictionnaire appris sur cette image

Comme précédemment, on extrait des patchs que nous normalisons. On opère cette mise à zéro des composantes et on apprend un dictionnaire pour lesquelles toutes les lignes des données manquantes sont mis à zéro. Enfin on minimise l'équation précédente. On peut ainsi reconstruire les patchs comme dans l'exercice précédent.

Voici le type de résultat que l'on peut obtenir :

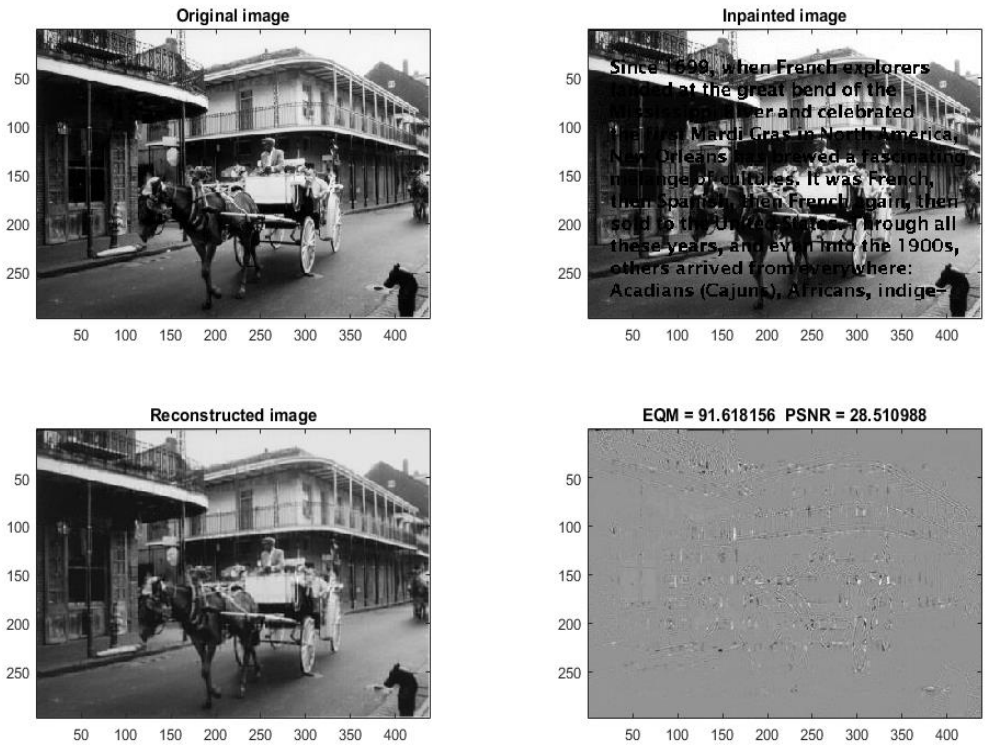

Figure 14 Résulta de reconstruction et mesure de l'erreur

De la même façon que précédemment, nous avons fait une mesure quantitative de la reconstruction à l'aide du PSNR.

IV Conclusion

La modélisation d'images comme combinaisons linéaires parcimonieuses d'atomes d'un dictionnaire est devenue un outil très populaire en traitement de l'image. Plusieurs techniques tante de calculer de tel dictionnaire. Les techniques les plus populaires abordent le problème via la minimisation d'une fonction de coût non-convexe, car il existe en théorie une infinité de solutions x au problème initiale y = Dx. L'objectif est alors de trouver la solution la plus parcimonieuse possible, c'est-a-dire celle présentant le plus faible nombre de valeurs non nulles dans x. En pratique, on cherche une approximation du signal et le problème devient :

$$\min_x ||x||_0, \text{ sous la contrainte } ||y - Dx||_2 \leq \epsilon$$

L'utilisation de l'algorithme K-SVD a fait ses preuves afin de résoudre ces problèmes. A partir de l'apprentissage de D on a vu qu'ont été capable de réaliser des opérations difficiles de reconstruction d'images, de débruitage ou bien d'inpainting.