

Pour exploiter pleinement les bases de données, il convient de réduire la taille des données tout en apportant une valeur ajoutée à l'exploitation thématique de l'information. Dans ce projet, nous souhaitons apporter de nouvelles pistes d'amélioration afin de réduire la taille de la base des données et assurer une extraction thématique de l'information utile alors qu'elle n'était pas exploitable à partir des données initiales.

La redondance est présente dans une image à cause de la corrélation entre les pixels. La transformation de l'image en une représentation où les coefficients sont peu corrélés réduit ce type de redondance.

Les représentations parcimonieuses consistent à approcher un signal, mis sous la forme d'un vecteur, par une combinaison linéaire de quelques colonnes seulement (atomes), d'une matrice appelée **dictionnaire**. Le signal est alors représenté par un vecteur parcimonieux contenant seulement quelques coefficients non nuls.

1 Apprentissage sur un ensemble de signaux

L'apprentissage du dictionnaire sur des données d'entraînement permet de l'adapter à un type de données particulières et ainsi d'améliorer la qualité de la représentation de ces données et de rendre les représentations plus parcimonieuses.

Le nombre d'atomes K dans un dictionnaire n'est pas limité, on choisit souvent un nombre d'atomes K pour le dictionnaire supérieur ou égal à la dimension des atomes ($K \geq n$). Dans ce cas, le dictionnaire est dit surcomplet ("**overcomplete**") et le dictionnaire obtenu aura donc certains atomes corrélés. De manière générale, plus le nombre d'atomes K augmente, plus les atomes du dictionnaire sont redondants, cela entraîne que beaucoup d'entre eux peuvent ne pas être utilisés et on obtient des représentations encore plus parcimonieuses.

Cependant, un dictionnaire contenant un grand nombre d'atomes est difficile à stocker et sera difficile à manipuler pour la recherche des coefficients. Il faut aussi tenir compte du coût en calculs pour la recherche de la représentation parcimonieuse des signaux et la mise à jour du dictionnaire qui augmente en fonction du nombre d'atomes K du dictionnaire.

L'utilisation de dictionnaires appris a eu un grand succès ces dernières années. L'idée principale utilisée est que les atomes du dictionnaire peuvent capturer la structure de l'image. L'image est traitée comme un ensemble de patches (blocks d'images) et on apprend un dictionnaire de patches sur un ensemble d'images non bruitées. On effectue alors la décomposition de patches bruités à l'aide du dictionnaire appris.



Livrable : Apprentissage d'un dictionnaire D

1. Coder l'algorithme d'apprentissage d'un dictionnaire sur-complet.
2. Afficher le dictionnaire (en termes d'image).

2 Compressive Sensing (CS) Measurement

Selon la théorie du compressive sensing, une image peut être recouverte (reconstruite) à partir de peu de mesures linéaires, si cette image suffisamment parcimonieuse dans un certain dictionnaire. Supposons qu'on a une image 2D, \mathbf{X} image de taille $\mathbf{N}_1 \times \mathbf{N}_2$

Nous convertissons cette image en un vecteur colonne \mathbf{x} de taille $\mathbf{N} = \mathbf{N}_1 \mathbf{N}_2$ en concaténant les colonnes de \mathbf{X} puis nous considérons \mathbf{M} mesures de \mathbf{x} , $\mathbf{y} \in \mathbb{R}^{\mathbf{M}}$ de telle sorte

$$\mathbf{y} = \Phi \mathbf{x} \quad (1)$$

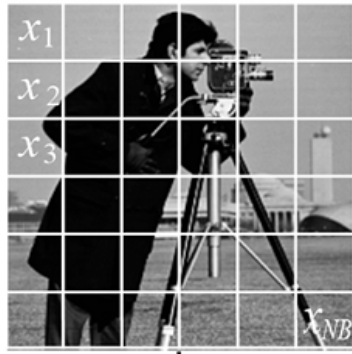
où Φ est une $\mathbf{M} \times \mathbf{N}$ matrice aléatoire de mesures. En pratique, la grande valeur de \mathbf{N} rend l'implémentations du CS pour une image occupe beaucoup de mémoire et s'avère complexe.

2.1 Image blocking

L'image \mathbf{X} est partitionnée en \mathbf{N}_B blocs, \mathbf{X}_j , $j = 1, \dots, \mathbf{N}_B$ de même taille $\mathbf{B} \times \mathbf{B}$, sans chevauchement et permettant le recouvrement de l'image initiale.

Image Blocking.....

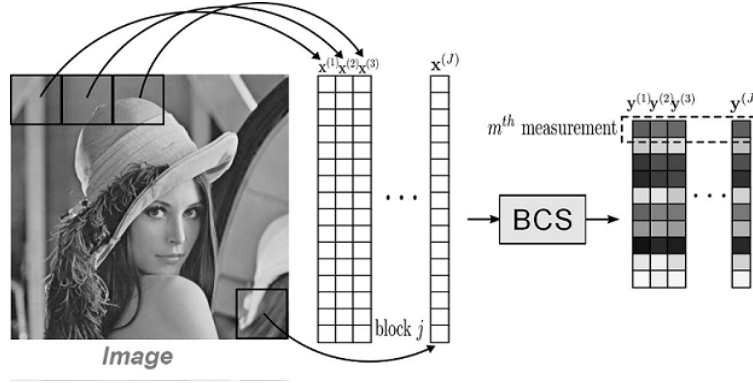
Original Image



2.2 Encoder – Blocked Compressive Sensing (BCS)

Chaque bloc \mathbf{X}_j mis sous forme d'une représentation vectorielle \mathbf{x}_j , $j = 1, \dots, \mathbf{N}_B$ est encodé par une matrice de mesure appropriée, Φ_B de taille $\mathbf{M} \times \mathbf{B}^2$:

$$\forall j = 1, \dots, \mathbf{N}_B, \mathbf{y}_j = \Phi_B \mathbf{x}_j \quad (2)$$



L'ensemble de ces mesures et leur allocation dans l'image est transmis ou stocké avec la matrice de mesure utilisée afin de décoder et reconstruire l'image initiale.

Livrable : Mesures

?

 Coder des différentes matrices de mesures avec des différents ratio de mesures.

3 Image Reconstruction

Connaissant les mesures \mathbf{y}_j , $j = 1, \dots, \mathbf{N}_B$ et Φ on se propose de déterminer ("approcher") \mathbf{x}_j . La technique de reconstruction de La projection de Landweber consiste à déterminer \mathbf{x}_j telle que $\|\mathbf{x}_j - \Phi_B \mathbf{y}_j\|_2^2$ soit la plus petite possible tout en ayant une solution parcimonieuse dans un certain domaine (ici on s'intéressera au domaine de la transformée en cosinus discrète DCT2D). Il s'agit donc d'un problème de minimisation sous contrainte:

$$\begin{cases} \min (f(\mathbf{x}_j) = \|\mathbf{x}_j - \Phi_B \mathbf{y}_j\|_2^2) \\ \text{sous contrainte } \{\mathbf{x} \text{ parcimonieux}\} \end{cases}$$

La résolution de ce type de problème peut se résoudre de façon itérative: on minimise dans un premier lieu f (sans prendre compte de la contrainte) et on projette la solution obtenue de façon à trouver une approximation parcimonieuse. La technique du SPL est une sorte du gradient projeté.

1. L'optimisation sans contrainte: f est une fonction quadratique. À l'itération k , supposons connus $\mathbf{x}_j^{(k)}$, l'itéré $\mathbf{x}_j^{(k+1)}$ se calcule par la descente du gradient à pas constant:

$$\mathbf{x}_j^{(k+1)} = \mathbf{x}_j^{(k)} + \Phi_B^T (\mathbf{y}_j - \Phi_B \mathbf{x}_j^{(k)})$$

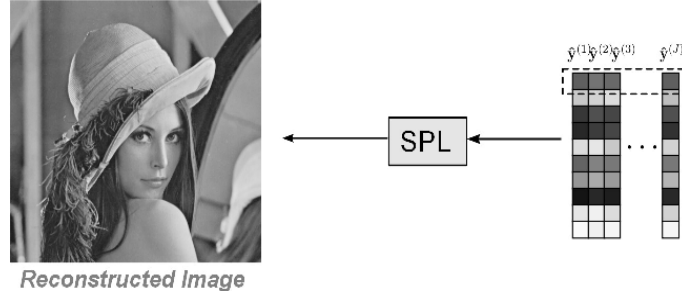
2. La projection dans un domaine de parcimonie: à partir des vecteurs $\mathbf{x}_j^{(k+1)}$, on reconstruit l'image $\mathbf{X}_1^{(k+1)}$ et cette dernière n'est pas forcément parcimonieuse, la seconde étape consiste à rendre notre image solution parcimonieuse:

- (a) on applique une transformée DCT à $\mathbf{X}_1^{(k+1)}$ pour obtenir une image $\Gamma_1^{(k+1)}$.
- (b) On applique un seuillage fort pour améliorer la parcimonie de $\Gamma_1^{(k+1)}$, on obtient $\Gamma_2^{(k+1)}$.

- (c) On reconstruit l'image à partir de $\Gamma_2^{(k+1)}$ en appliquant une transformée DCT inverse à $\Gamma_2(k+1)$ et on obtient une image $\bar{\mathbf{X}}_1^{(k+1)}$.
- (d) Ce seuillage permet d'améliorer la parcimonie mais peut nous éloigner de la bonne approximation recherchée, ainsi, on met à jour notre solution du problème de minimisation: pour chaque bloc vectorisé de $\bar{\mathbf{X}}_1^{(k+1)}$, on calcule

$$\bar{\mathbf{x}}_j^{(k+1)} = \bar{\mathbf{x}}_j^{(k)} + \Phi_{\mathbf{B}}^T (\mathbf{y}_j - \Phi_{\mathbf{B}} \bar{\mathbf{x}}_j^{(k)})$$

pour obtenir une image reconstruite $\mathbf{X}_2^{(k+1)}$.



L'algorithme de la projection de Landweber s'accompagne d'une phase de filtrage pour éviter les artefacts, on utilisera le filtre de Weiner(3,3) et s'initialise pour chaque blocs $j = 1, \dots, \mathbf{N}_{\mathbf{B}}$ comme suit

$$\mathbf{x}_j = \Phi_{\mathbf{B}}^T \mathbf{y}_j \quad (1)$$

3.1 Le seuillage

Le seuillage nous permettra à chaque itération de rendre la solution la plus parcimonieuse, on utilisera un seuillage fort universel (dans le traitement d'images). A chaque itération, on calcule:

$$\tau^{(k)} = \lambda \frac{\text{median}(|\Gamma_1^{(k)}|)}{0.6745} \sqrt{2 \ln(\mathbf{N}_1 \mathbf{N}_2)}$$

λ un facteur de contrôle fixé généralement à 6 et le critère de seuillage sera

$$\Gamma_2^{(k)} = \begin{cases} \Gamma_1^{(k)} & \text{si } \Gamma_1^{(k)} > \tau^{(k)} \\ 0 & \text{sinon} \end{cases}$$

On répète l'algorithme un nombre de fois, itermax=200 (par exemple) et s'arrête selon un critère d'arrêt

$$|\mathbf{D}^{(k+1)} - \mathbf{D}^{(k)}| < \text{Tolérance}$$

$$\text{où } \mathbf{D}^{(k)} = \frac{1}{\sqrt{\mathbf{N}_1 \mathbf{N}_2}} \|\mathbf{X}_2^{(k)} - \mathbf{X}_1^{(k)}\|$$

Livable : SPL

1. Coder l'algorithme BCS-SPL-DCT pour une itération.
2. Coder l'algorithme général (itermax iteration) avec critère d'arrêt.
3. Reconstruire l'image.

Algorithm BCS-SPL Reconstruction Algorithm

Input: The measurement matrix for the i -th block: φ_i , a constant used to control the convergence speed: λ , the measurement: y , signal: $x^{(r)}$, 2D transform: ψ

Output: signal: $x^{(r+1)}$

1: $\hat{x}^{(r)} = \text{Wiener}(x^{(r)})$

2: **for each** block i

3: $\hat{x}_i^{(r)} = \hat{x}_i^{(r)} + \varphi_i^T (y_i - \varphi_i \hat{x}_i^{(r)})$

4: **end for**

5: $\tilde{x}^{(r)} = \psi \hat{x}^{(r)}$

6: $\tilde{x}^{(r)} = \text{Threshold}(\tilde{x}^{(r)}, \lambda)$

7: $\bar{x}^{(r)} = \psi^{-1} \tilde{x}^{(r)}$

8: **for each** block i

9: $x_i^{(r+1)} = \bar{x}_i^{(r)} + \varphi_i^T (y_i - \varphi_i \bar{x}_i^{(r)})$

10: **end for**
