

Introduction au Compressed sensing. Notions de mesures et de parcimonie¹

Guillaume Lecué²

Résumé

Le Compressed Sensing (CS) est un problème de traitement de données où les étapes d'acquisition et de compression se font en même temps ; lors de l'acquisition, par un procédé "intelligent" de prises de mesures. Un tel procédé permet de grands gains de performances en terme de stockage ainsi que de coût / vitesse d'acquisition. Un grand nombre de données réelles peuvent être réécrites dans le paradigme du CS. Dans cette partie nous présentons quelques exemples concrets où le CS est une modélisation pertinente.

Bien sûr, un point clef du succès du CS est que la donnée stockée / observée doit être en mesure de restituer par des algorithmes rapides, stable et robuste le signal d'origine. Nous verrons que cette reconstruction peut se faire de manière efficace pour certains types de signaux qui ont une représentation "parcimonieuse" dans une base choisie.

La notion de parcimonie est centrale dans la théorie du CS. L'objectif de cette première partie est de présenter cette notion et de montrer qu'elle est naturellement présente dans des données réelles.

Dans cette première partie, on introduit donc deux notions clefs : mesures et parcimonie. La formalisation / modélisation du problème du CS et d'autres problèmes annexes est aussi donnée.

1 Introduction

Le *Compressed sensing* peut se traduire par "acquisition comprimée". Cela signifie que l'acquisition et la compression d'un signal se font en une seule étape.

Avant le CS, la compression des données était une étape à part de celle de l'acquisition. Par exemple, en photographie numérique, la première étape est de prendre la photo c'est-à-dire d'exposer la plaque photo-sensible de l'appareil photos à la lumière de la scène à photographier. Cette donnée est stockée au format RAW : la scène est pixelisée et le niveau de couleur de chaque pixel est stocké dans une grande matrice. Pour les appareils photos actuels, le stockage d'une photo en format RAW occupe un espace mémoire de l'ordre du Méga-octet. Ceci constitue la première étape : celle de l'**acquisition**. La deuxième étape s'effectue généralement à l'aide d'un CPU (soit en interne à l'appareil photos, soit sur PC après transfert des fichiers RAW). Elle consiste à lancer un algorithme de compression sur le fichier RAW associé à l'image. Cette deuxième étape est la **compression** du signal qui permet de diviser la place de stockage jusqu'à 25, soit une centaine de kilo-octet de stockage pour une photo au format compressé au lieu des quelques Méga-octet au format RAW. Le format JPEG est une extension classique des photos compressées suivant un certain algorithme basé sur la transformation des images (par blocs) dans une base de cosinus et à une quantification des coefficients obtenus. La compression n'est utile que

1. Notes du cours du 26 janvier 2016

2. CNRS, CREST, ENSAE. Bureau E31, 3 avenue Pierre Larousse. 92 245 Malakof. Email : guillaume.lecue@ensae.fr.

s'il est possible de reconstruire l'image d'origine à partir de celle compressée de manière rapide et efficace : c'est-à-dire sans trop de perte – moralement, l'œil humain ne doit pas voir la différence entre l'image d'origine et l'image reconstruite. C'est le cas de la compression JPEG, qui malgré une perte d'information due (entre autres) à l'étape de quantification des coefficients de la DCT (discrete cosine transform), permet une reconstruction d'images très fidèle à celles d'origine.



FIGURE 1 – Les deux étapes dans l'approche classique : 1) acquisition 2) compression.

Contrairement à l'approche classique, le CS propose de réaliser les deux étapes précédentes d'acquisition et de compression en une seule étape ; d'où la traduction française, d'acquisition comprimée. L'objectif est donc de prendre des mesures “intelligentes” du signal qu'on souhaite connaître (ou estimer) de telle sorte qu'on puisse reconstruire ce signal seulement à partir d'un petit nombre de ces mesures.

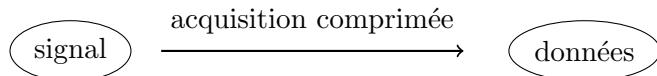


FIGURE 2 – En CS, l'acquisition et la compression se font en une seule et même étape.

La notion de **mesure** est donc centrale en CS. Une grande partie du travail théorique qui a eu lieu (et qui continue encore) en recherche sur le CS porte sur la construction de mesures. En CS, on ne considère que des mesures linéaires des signaux. Si x est le signal que nous souhaitons connaître, on effectuera alors des mesures de la forme

$$y_i = \langle x, X_i \rangle \quad (1.1)$$

où :

1. i : le numéro de la mesure,
2. X_i : i -ième vecteur de mesure,
3. y_i : i -ième mesure ou observation.

On notera par m le nombre de mesures. On ne s'intéressera qu'à la reconstruction de signaux dans des espaces de dimension finie. On suppose alors que x appartient à \mathbb{R}^N (ou \mathbb{C}^N – pour simplifier la présentation, on ne considère que le cas de \mathbb{R}^N ici). On peut alors réécrire les mesures observées sous forme matricielle :

$$y = Ax \quad (1.2)$$

où :

1. $y \in \mathbb{R}^m$: vecteur de mesures,
2. $A \in \mathbb{R}^{m \times N}$: matrice de mesure dont les lignes sont les vecteurs de mesure X_i^\top ,

$$A = \begin{pmatrix} X_1^\top \\ \vdots \\ X_m^\top \end{pmatrix} \quad (1.3)$$

3. $x \in \mathbb{R}^N$: signal à reconstruire.

L'objectif étant d'effectuer le moins de mesures possible – pour minimiser le coût de stockage ou le coût de l'acquisition – on aura toujours

$$m < N \quad (1.4)$$

c'est-à-dire le nombre de mesures m sera en général beaucoup plus petit que la dimension N de l'espace ambiant (c'est l'espace où vivent les signaux à reconstruire). C'est le cadre classique des **statistiques en grandes dimensions** : le nombre d'observations m est plus petit que la dimension N de l'espace des paramètres à estimer (le signal x est perçu comme le paramètre indexant le modèle statistique).

Pour l'instant, le problème en CS s'énonce de la manière suivante : retrouver $x \in \mathbb{R}^N$ à partir de $y (= Ax)$ et de A . D'un point de vue algébrique, on est donc amener à résoudre un problème d'équations linéaires à m équations et N inconnues (les coordonnées de x) alors que $m < N$ (et même, en général, $m \ll N$). La Figure 1 représente le problème comme un problème “hautement sous-déterminé” : il y a beaucoup plus d'inconnues que d'équations. Il y a donc plusieurs solutions $x \in \mathbb{R}^N$ telles que $y = Ax$ et donc, sans hypothèse supplémentaire sur x , on ne pourra pas reconstruire x à partir de y et A dès que $m < N$.

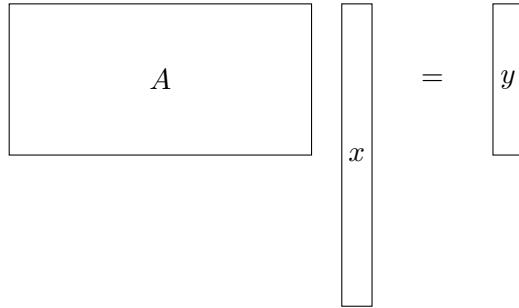


FIGURE 3 – Trouver x à partir de $y = Ax$ et A : le Compressed Sensing peut se voir comme un problème de résolution d'un système linéaire hautement sous-déterminé.

L'hypothèse centrale en CS est que le signal x à reconstruire est **parcimonieux** – on emploiera aussi l'anglicisme “**sparse**” – c'est-à-dire que la majorité des coordonnées de x sont nulles. Ce qui revient à dire que la taille du support de x (c'est l'ensemble des indices des coordonnées non-nulles de x) est petit par rapport à N .

Définition 1.1 Soit N et s deux entiers tels que $1 \leq s \leq N$. Soit $x \in \mathbb{R}^N$. Le **support de x** est l'ensemble

$$\text{supp}(x) = \{j \in \{1, \dots, N\} : x_j \neq 0\}$$

où pour tout $1 \leq j \leq N$, x_j est la j -ième coordonnée de x . La taille du support de x est notée

$$\|x\|_0 = |\text{supp}(x)|$$

c'est le nombre de coordonnées non nulles dans x . On dit qu'un vecteur est **s -sparse** (ou **parcimonieux**) quand $\|x\|_0 \leq s$ et on note par Σ_s l'ensemble de tous les vecteurs s -sparse de \mathbb{R}^N :

$$\Sigma_s = \{x \in \mathbb{R}^N : \|x\|_0 \leq s\}.$$

On fait donc l'hypothèse que le signal x est s -sparse. On verra par la suite des exemples pour lesquels cette hypothèse est bien vérifiée par des signaux “concrets”.

Remarque 1.2 Dans la définition 1.1, la base canonique joue un rôle central : la notion de parcimonie est définie par rapport à la base canonique. Cette définition s'étend à n'importe quelle base ou même à des dictionnaires (ensemble de signaux sans structure particulière). Si $\mathcal{F} = \{f_1, \dots, f_p\}$ est un ensemble d'éléments de \mathbb{R}^N et $x \in \mathbb{R}^N$, on dit que x est s -sparse dans le dictionnaire \mathcal{F} quand il existe $J \subset \{1, \dots, p\}$ tel que

$$|J| \leq s \text{ et } x = \sum_{j \in J} \theta_j f_j.$$

Ainsi x s'écrit de manière concise dans \mathcal{F} : comme une combinaison linéaire de seulement s termes de \mathcal{F} . On parle alors de **parcimonie structurée**.

Trouver les “bonnes bases” ou “bons dictionnaires” dans lesquels certains types de signaux (images, vidéo, etc.) ont des représentations parcimonieuses a été et est encore l'objet de nombreuses recherches en théorie de l'approximation. Par exemple, il s'avère que certaines bases d'ondelettes sont pour de nombreux types de données “réelles” de bonne bases de ce point de vue. Cet aspect du traitement du signal ne sera pas étudié dans ce cours (on renvoie le lecteur intéressé au livre [6] ou aux numerical tours de [7]). On ne va donc considérer dans ce qui suit que la notion de parcimonie non-structurée, c'est-à-dire celle définie par rapport à la base canonique comme dans la Définition 1.1. Le cas général revient à faire la transformation suivante $x = F\theta$ où $F \in \mathbb{R}^{N \times p}$ est la matrice dont les colonnes sont données par les $f_j, j = 1, \dots, p$ et $\theta \in \mathbb{R}^p$ est un vecteur s -sparse et donc à remplacer la matrice de mesure A par AF . La parcimonie de x dans \mathcal{F} s'exprime donc par la parcimonie de θ dans la base canonique.

Remarque 1.3 Les trois points faibles de l'approche en deux étapes “acquisition-puis-compression” sont :

1. La phase d'acquisition nécessite le stockage de N données, alors que pour les problèmes qui nous intéressent N est typiquement grand (problème de stockage)
2. On doit calculer l'ensemble des N coordonnées du signal dans la base de compression alors qu'on ne va en garder qu'une petite proportion (problème de temps de calcul inutilement utilisé)
3. la localisation des grands coefficients dans la base de compression doit être stocké pour chaque image.

Dans ce qui suit nous présentons plusieurs problèmes concrets où les données peuvent être traduites dans le cadre du CS. Nous commençons cependant par donner dans la section suivante une formalisation du problème de CS et deux variantes importantes (données bruitées et signal “approximativement sparse”).

2 Formalisation du problème en compressed sensing

Dans la section précédente, les notions de *mesure* et de *parcimonie* ont été introduites. La parcimonie est une hypothèse fondamentale sur les signaux à reconstruire et les mesures sont dans certains cas au libre choix du statisticien et dans d'autres cas, “imposées” par la nature. Dans le premier cas où le statisticien choisit les mesures, c'est-à-dire les vecteurs de mesures X_i introduits dans (1.1), on peut parler d'un plan d'expérience et donc d'une certaine stratégie de choix de mesures. L'objectif étant de pouvoir reconstruire le signal (parcimonieux) à partir d'un nombre minimum de mesure. Dans ce cas, on peut alors énoncer le problème de CS de la manière suivante.

Problème numéro 1 en CS : Construire, en nombre minimal, des vecteurs de mesures X_1, \dots, X_m tels que pour tout vecteur $x \in \Sigma_s$, on puisse reconstruire efficacement x à partir de $y (= Ax)$ et de $A \in \mathbb{R}^{m \times N}$, la matrice de mesures définie dans (1.3)

Il reste donc à définir ce qu'on entend par "efficacement". Ici, et pour l'instant, une reconstruction efficace de x est valide lorsqu'on peut implémenter un algorithme qui marche effectivement càd qui ne demande pas un temps et une puissance de calcul trop importante. On donnera dans la suite une définition plus précise des algorithmes qu'on s'interdit d'implémenter en pratique (les algorithmes NP-hard) et de ceux qui sont particulièrement apprécié (programmation linéaire ou Semi-Definite Programming). Il y a donc une composante "computationnelle" très importante dans le CS, ceci constitue un autre problème centrale au CS.

Problème numéro 2 en CS : Etant donnés des vecteurs de mesures X_1, \dots, X_m , construire un algorithme de reconstruction efficace de tout vecteur $x \in \Sigma_s$, à partir de $y (= Ax)$ et de $A \in \mathbb{R}^{m \times N}$ (définie dans (1.3)). C'est à dire une application

$$\Delta : \mathbb{R}^m \mapsto \mathbb{R}^N \quad (2.1)$$

qui s'implémente de manière efficace et telle que

$$\text{pour tout } x \in \Sigma_s, \Delta(Ax) = x. \quad (2.2)$$

(il y a ici un léger abus de notation car l'algorithme de reconstruction Δ utilise A , on devrait donc noter $\Delta(Ax, A)$ en lieu de $\Delta(Ax)$).

Nous reviendrons sur le problème d'efficacité des algorithmes dans les sections suivantes concernant les algorithmes de minimisation ℓ_0 et ℓ_1 . Pour l'instant, le point important est que les mesures sont à choisir de telle sorte à avoir besoin du moins de mesures possible pour pouvoir reconstruire (efficacement) n'importe quel signal x s -sparse. On insiste sur le fait qu'on souhaite être capable d'identifier m vecteurs de mesures qui permettent de reconstruire (efficacement) tous les vecteurs $x \in \Sigma_s$ uniquement à partir de $y (= Ax)$ et A où A est la même matrice de mesure.

On utilisera donc le même ensemble de vecteurs de mesure $\{X_1, \dots, X_m\}$ pour tous les signaux s -sparse x . Cette notion d'uniformité sur Σ_s joue un rôle important pour déterminer le nombre minimal de mesures m permettant la reconstruction efficace sur Σ_s .

Il y a donc principalement deux problèmes en CS :

1. un problème de choix des vecteurs de mesure X_1, \dots, X_m ,
2. un problème de construction d'algorithmes de reconstruction de signaux s -sparse à partir d'un système d'équations hautement sous-déterminé : $\Delta : \mathbb{R}^m \mapsto \mathbb{R}^N$ tel que pour tout $x \in \Sigma_s, \Delta(Ax) = x$.

Souvent ces deux problèmes sont traités en même temps ; càd, on construit des vecteurs de mesure de telle sorte qu'un algorithme donné satisfasse (2.2).

Remarque 2.1 Le compressed sensing est un exemple de problème de statistiques en grandes dimensions car le nombre d'observations – ici appelées mesures – est bien plus petit que la dimension de l'objet qu'on souhaite estimer, ici N . Intuitivement, la "vraie" dimension d'un vecteur s -sparse est s et non N – ce serait réellement le cas si on connaissait le support de x . Dans le cas d'un problème d'estimation d'un objet de dimension s le nombre minimal d'observation qu'il faut

obtenir pour que l'estimation ne soit pas triviale est au moins s . On aimerait donc pour le problème de CS que le nombre minimal de mesures suffisant pour reconstruire tout vecteur s -sparse soit aussi de l'ordre de s auquel il faudrait, dans certains cas, ajouter un coût parce que le support du signal est inconnu. Intuitivement, on travaillera dans le contexte $s \ll N$ et “idéalement” $m \sim s$ – ceci nous donne une idée de l'ordre de grandeur attendu pour le nombre de mesures.

Dans les deux sous-sections qui suivent, on introduit deux variantes “plus réaliste” au problème du compressed sensing précédemment introduit.

2.1 Données bruitées – procédures robustes

Dans beaucoup de situation, l'observation des données est bruitées. On observe alors des mesures linéaires d'un signal x auquel s'est ajouté un bruit. On modélise ces données par :

$$y_i = \langle X_i, x \rangle + \xi_i \text{ pour tout } i = 1, \dots, m \quad (2.3)$$

où ξ_i est le terme de bruit sur la i -ième observation. En statistiques, il est courant de modéliser les bruits ξ_1, \dots, ξ_m comme une famille de m variables aléatoires, généralement indépendantes et de même distribution, et dans beaucoup de cas cette distribution est Gaussienne centrée de variance (inconnue) σ . En notation matricielle, on observe

$$y = Ax + \xi \quad (2.4)$$

où $y = (y_i)_{i=1}^m$, A est la matrice de mesure définie dans (1.3) et $\xi = (\xi_i)_{i=1}^m$ est le vecteur des bruits.

Dans le cas, de données bruitées, il n'est plus possible de reconstruire exactement x à partir de $y = (y_i)_{i=1}^m$ et A comme dans (2.2). On cherchera donc à estimer (toujours efficacement – dans un sens *computational*) le signal x . On souhaitera que cette reconstruction se fasse de manière robuste selon la définition suivante.

Définition 2.2 *On dit qu'un algorithme de reconstruction (ou de décompression) $\Delta : \mathbb{R}^m \mapsto \mathbb{R}^N$ est robuste (au bruit) d'ordre s quand pour tout $x \in \Sigma_s$ et tout vecteur bruit $\xi \in \mathbb{R}^m$, on a*

$$\|x - \Delta(Ax + \xi)\|_2 \leq c_0 \|\xi\|_2$$

où c_0 est une constante absolue.

L'idée derrière la propriété de robustesse est que s'il n'y a pas de bruit – càd $\xi = 0$ – alors Δ est bien un algorithme de reconstruction exacte sur Σ_s car $\Delta(Ax) = x$ pour tout $x \in \Sigma$. Mais si les données sont bruitées alors on souhaite quand même assurer que la reconstruction de x par $\Delta(Ax + \xi)$ fasse une erreur au plus de l'ordre de grandeur du bruit mesurée ici par la norme ℓ_2^m .

Pour de nombreux problèmes concrets, la propriété de robustesse au bruit est une nécessité car même si les données sont bruitées on peut espérer que la magnitude du bruit (mesurée ici par $\|\xi\|_2$) ne sera pas trop grande par rapport à l'amplitude du signal (ici $\|x\|_2$). En statistique, $\|x\|_2 / \|\xi\|_2$ est appelée SNR – *signal-to-noise ratio*. Le SNR mesure la faisabilité du problème de dé-bruitage : quand le SNR est petit alors le signal est “noyé” dans le bruit, on aura donc très peu d'espoir de bien estimer x et inversement.

2.2 Signaux approximativement parcimonieux – procédures stables

Les signaux qu'on cherche à reconstruire ne sont pas, en général, exactement s -sparse mais seulement “proche” d'un signal s -sparse. On aimerait donc s'assurer que les procédures que nous avons construites (en pensant que les signaux à estimer sont exactement s -sparse) réagissent bien face à cette éventualité. Cette propriété est appelée la *stabilité*.

Définition 2.3 *On dit qu'un algorithme de reconstruction (ou de décompression) $\Delta : \mathbb{R}^m \mapsto \mathbb{R}^N$ est stable d'ordre s quand pour tout $x \in \mathbb{R}^N$, on a*

$$\|x - \Delta(Ax)\|_2 \leq c_0 \frac{\min_{z \in \Sigma_s} \|x - z\|_1}{\sqrt{s}}$$

où c_0 est une constante absolue.

Le choix des normes dans la propriété de stabilité est important. On voit que l'erreur de reconstruction “ $\|x - \Delta(Ax)\|_2$ ” est mesurée par rapport à la norme ℓ_2^N alors que l'erreur d'approximation “ $\min_{z \in \Sigma_s} \|x - z\|_1$ ” est mesurée par rapport à la norme ℓ_1^N . Idéalement, on aurait souhaité que l'erreur d'approximation soit aussi mesurée en norme ℓ_2^N cependant on peut montrer (on le verra en exercice) que s'il existe un algorithme de reconstruction Δ tel que pour tout $x \in \Sigma_s$, même $s = 1$, vérifiant $\|x - \Delta(Ax)\|_2 \leq c_0 \min_{z \in \Sigma_s} \|x - z\|_2$ alors nécessairement $m \geq c_1 N$ où c_1 est une constante absolue dépendant uniquement de c_0 . On ne peut donc pas avoir la propriété de stabilité pour une erreur d'approximation mesurée en norme ℓ_2^N sauf si le nombre de mesures est de l'ordre de grandeur de la dimension de l'espace – ce qu'on souhaite éviter à tout prix.

Il est facile de voir que le terme d'approximation satisfait :

$$\sigma_s(x)_1 := \min_{z \in \Sigma_s} \|x - z\|_1 = \|x - x_s^*\|_1$$

où x_s^* est le vecteur de Σ_s des s plus grandes coordonnées de x en valeur absolue c'est-à-dire tel que pour tout $1 \leq j \leq N$,

$$(x_s^*)_j = \begin{cases} x_j & \text{quand } j \in J_s \\ 0 & \text{sinon} \end{cases}$$

où $J_s = \{j_1, \dots, j_s\}$ et $|x_{j_1}| \geq |x_{j_2}| \geq \dots \geq |x_{j_s}| \geq x_j$ pour tout $j \notin J_s$. Ainsi le vecteur s -sparse le plus proche de x (pour la norme ℓ_1^N) est simplement un vecteur tronqué où seuls les s -plus grandes coordonnées de x ont été gardées.

La propriété de stabilité d'un algorithme assure que si x est bien s -sparse alors la reconstruction est bien exacte (càd $\Delta(Ax) = x$ pour tout $x \in \Sigma_s$) mais si le signal x n'est pas exactement s -sparse mais, seulement proche d'un vecteur s -sparse alors $\sigma_s(x)_1$ sera petit et dans ce cas la reconstruction $\Delta(Ax)$ ne sera pas loin de x quand Δ satisfait la propriété de stabilité. Comme dans le cas de la robustesse, l'erreur d'estimation est de l'ordre de l'erreur au modèle “idéal” (sans bruit et signal exactement sparse). Ici le cadre idéal n'est pas satisfait car le signal n'est pas exactement s -sparse ainsi on paie un prix qui est mesuré par cette erreur d'approximation $\sigma_s(x)_1$ vue que nos procédures auront été pensées et construites dans le cadre idéal même si on les adaptera aux cas plus réels de données bruitées et de signaux non exactement s -sparse.

Finalement, les seuls algorithmes qui auront un intérêt d'un point de vue pratique devront rassembler les deux propriétés de robustesse et de stabilité. Il sera en effet fréquemment rencontrés sur des données réelles la présence de bruit dans les mesures et des signaux approximativement parcimonieux. La définition suivante rassemble ces deux aspects.

Définition 2.4 On dit qu'un algorithme de reconstruction (ou de décompression) $\Delta : \mathbb{R}^m \mapsto \mathbb{R}^N$ est **stable et robuste d'ordre s** quand pour tout $x \in \mathbb{R}^N$ et tout $\xi \in \mathbb{R}^m$, on a

$$\|x - \Delta(Ax + \xi)\|_2 \leq c_0 \frac{\min_{z \in \Sigma_s} \|x - z\|_1}{\sqrt{s}} + c_0 \|\xi\|_2$$

où c_0 est une constante absolue.

Une séance du cours sera consacrée à l'étude d'algorithme satisfaisant ces deux propriétés. La suite de ces notes de cours d'introduction au CS portent sur des exemples concrets. La présentation de ces exemples mettra particulièrement l'accent sur les notions centrales de mesures (linéaires) et de parcimonie.

Remarque 2.5 Jusqu'à maintenant, nous avons considéré des mesures de la forme $y = Ax + \xi$ où A est la matrice de taille $m \times N$ dont les m vecteurs lignes sont donnés par les vecteurs de mesure X_i , $1 \leq i \leq m$. Il est cependant courant de rencontrer une autre normalisation pour laquelle la matrice de mesures est normalisée par $1/\sqrt{m}$ (cette normalisation a un intérêt en probabilité car elle fait apparaître des moyennes empiriques). Les données sont alors

$$y' = A'x + \xi'$$

où $y' = y/\sqrt{m}$, $A' = A/\sqrt{m}$ et $\xi' = \xi/\sqrt{m}$. Pour ce choix de normalisation, une procédure sera stable et robuste d'ordre s quand pour tout $x \in \mathbb{R}^N$ et tout $\xi' \in \mathbb{R}^m$, on a

$$\|x - \Delta(A'x + \xi')\|_2 \leq c_0 \frac{\min_{z \in \Sigma_s} \|x - z\|_1}{\sqrt{s}} + c_0 \frac{\|\xi'\|_2}{\sqrt{m}} \quad (2.5)$$

où c_0 est une constante absolue.

C'est la normalisation habituellement utilisée en statistiques. Dans ce cadre, il est courant de faire l'hypothèse que le bruit ξ' est un vecteur gaussien standard de \mathbb{R}^m . Pour cet modélisation, $\|\xi\|_2$ est, avec très grande probabilité (au moins $1 - 2 \exp(-c_0 m)$), de l'ordre de grandeur de \sqrt{m} , le terme $\|\xi'\|_2 / \sqrt{m}$ est donc constant dans (2.5). Ce qui est optimal quand $m \sim s \log(eN/s)$.

Quand on a plus d'observations que nécessaire (càd plus que $s \log(eN/s)$), on peut construire des procédures \hat{x} (Lasso ou Dantzig) qui vérifient, avec grande probabilité, que

$$\|x - \hat{x}\|_2 \lesssim \inf_{z \in \Sigma_s} \left(\|x - z\|_2 + \sqrt{\frac{\|z\|_0 \log(n)}{m}} \right).$$

Mais ce résultat n'est vrai que pour un seul signal $x \in \mathbb{R}^N$ à reconstruire alors que dans Définition 2.4, il est demandé que ce résultat soit vrai pour tout vecteur x .

3 Compressed sensing pour la photographie – exemple de l'appareil photo à un pixel

L'exemple qui nous a servit de fil conducteur pour introduire la notion d'acquisition comprimée “en une seule étape” dans la première section était celui de la prise de photo. On poursuit l'exposition de cet exemple dans cette section.

Les appareils photos que nous utilisons actuellement sont basés sur l'ancien paradigme en compression de données qui s'opère en deux étapes : 1) l'acquisition 2) la compression (cf. Figure 1).

Quel serait un appareil photo utilisant l'idée centrale du CS, celle de l'acquisition comprimée (en une seule étape) ?

Le but est donc de construire des vecteurs de mesures – en petit nombre – tels que les données stockées permettent de reconstruire la scène photographiée. Au lieu d'enregistrer le niveau de couleur pour chaque pixel de la plaque photosensible de l'appareil, on va faire une mesure linéaire de ces niveaux de couleur. Cela revient à faire une moyenne locale du niveau de couleur selon la résolution et la luminosité dont on dispose puis à faire une somme pondérée de tous ces niveaux de couleur. La Figure 4 montre un exemple de mesure d'une scène.



FIGURE 4 – Mesure linéaire d'une scène. Ici le vecteur x qu'on souhaite reconstruire est une scène moyennisée localement – c'est l'image de gauche. C'est donc ici une matrice. La mesure X_i est l'image de droite où le niveau de noir indique le poids affecté au pixel auquel il est associé.

Dans l'approche classique, l'étape d'acquisition revient à faire des mesures comme montré dans la Figure 4 sauf que le vecteur de mesure vaut 0 partout sauf en un pixel où il vaut 1. Il y a donc autant de mesures que de pixels – d'où la grande taille des fichiers au format RAW.

Pour la construction d'un appareil photo utilisant l'idée d'acquisition comprimée, on va effectuer plusieurs mesures du même “type” que ceux de la Figure 4. On va donc construire plusieurs vecteurs de mesures et récupérer le produit scalaire entre la scène observée et le vecteur de mesure. Dans la Figure 5, on stock les mesures y_1, \dots, y_m . L'objectif étant de pouvoir reconstruire la scène de départ (pixelisée au niveau de résolution donnée) à partir des mesures y_i .

$$y_1 = \langle \text{Image}, \text{Matrice} \rangle , \dots , y_m = \langle \text{Image}, \text{Matrice} \rangle$$

FIGURE 5 – m mesures d'une scène.

Il est en particulier intéressant de remarquer que les valeurs y_1, \dots, y_m sont des combinaisons linéaires de niveaux de couleurs (ou de noir et blanc) ; ce sont donc des valeurs réelles qui peuvent être enregistrées avec une plaque photo sensible ne comportant qu'un seul point lumino-sensible (contrairement aux plaques photo-sensibles comportant actuellement plusieurs millions de points lumino-sensibles). Cette observation est à l'origine de la construction par la RICE University de “l'appareil photo à un pixel”.

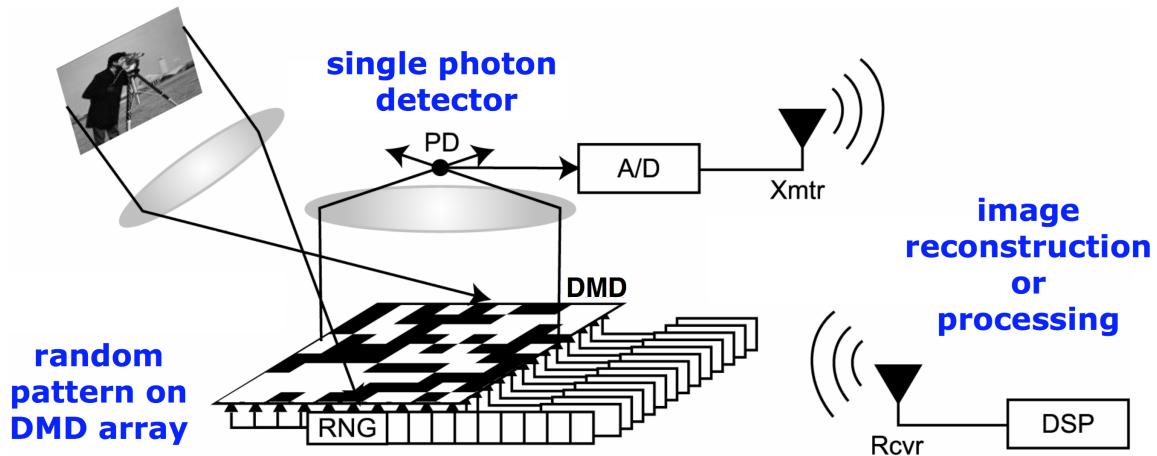


FIGURE 6 – Appareil photo à un pixel de la RICE University. Le vecteur de mesure est construit par un tableau de DMD (*digital micromirror device*) qui s’oriente de manière aléatoire. La mesure y_i est collectée par un seul détecteur de photons.

La construction des vecteurs de mesure dans l’appareil photo à un pixel se fait par un tableau de micro-miroir qui, en s’orientant, laisse plus ou moins passer la lumière. C’est la somme de tous ces niveaux de lumière plus ou moins atténués qui est ensuite envoyé à l’unique détecteur photo-sensible. Un point clef dans la construction des vecteurs de mesure est la présence de l’aléatoire.

On verra en effet plus tard qu’une construction efficace des vecteurs de mesures passe (pour l’état actuel du niveau de connaissances) par des constructions de mesures aléatoires. C’est donc la stratégie utilisée par la RICE University pour le choix d’orientation de tous les micros miroirs. La Figure 7 propose un zoom sur ces micro-miroirs.

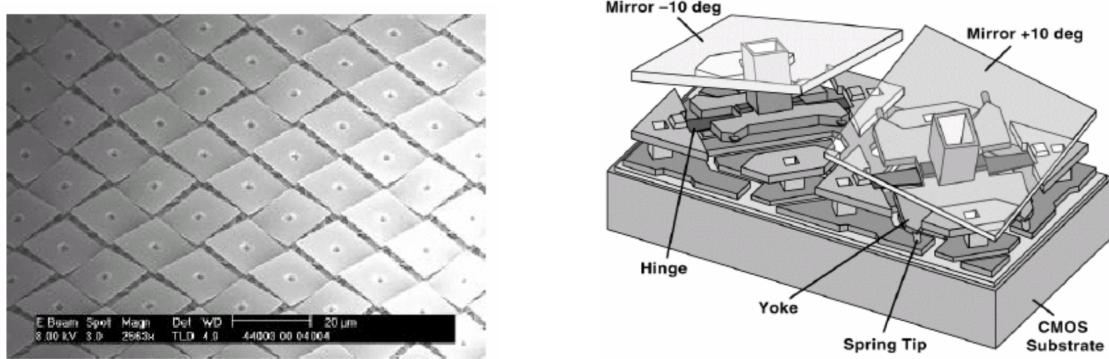


FIGURE 7 – Zoom sur les micro-miroirs utilisés dans l’appareil photo à un pixel de la RICE University. L’orientation et donc le niveau d’opacité de ces miroir est obtenu de manière aléatoire pour la construction de vecteurs de mesure aléatoires.

Il semble donc possible de construire un appareil photo utilisant l’idée de l’acquisition comprimée en une seule étape. Le problème maintenant est de savoir s’il fonctionne. Pour cela, une hypothèse clef en CS est l’hypothèse de parcimonie : le signal aussi compliqué qu’il puisse être ici peut-il être décrit par une structure de petite dimension ? C’est la question typique qu’on se

pose en CS et plus généralement en statistiques en grandes dimensions. Ici, on se demande si les images peuvent être décrites de manière concise – c'est avec seulement un petit nombre de réels ?

Cette question est bien plus ancienne que le problème du CS. De nombreux résultats ont été obtenus dans les années 80/90 notamment grâce à l'introduction des ondelettes. L'idée centrale des ondelettes est de proposer des bases qui sont à la fois localisées en temps et en fréquence (contrairement à la base de Fourier qui n'est localisée qu'en fréquence). Les ingénieurs et chercheurs en traitement du signal ont réussi à construire des bases dans lesquelles les coefficients du développement des photos "réelles" sont en grande majorité très petits. Formellement, si on représente une image comme un tableau de nombre $f \in \mathbb{R}^N$ – où N est ici de la forme $N = N_1 \times N_2$ (N_1 étant le nombre de lignes et N_2 étant le nombre de colonnes) – alors on peut construire des bases orthonormales $(\psi_j)_{j=1}^N$ de \mathbb{R}^N telles que si on développe f dans ces bases,

$$f = \sum_{j=1}^N \langle f, \psi_j \rangle \psi_j$$

alors les coefficients $(\langle f, \psi_j \rangle)_{j=1}^N$ du développement sont dans leur grande majorité petits (relativement aux autres coefficients qui sont eux plus grands et qui portent presque toute l'information intéressante). Si bien que si on ne garde que les s plus grands coefficients dans le développement de f

$$f_s^* = \sum_{j \in J_s} \langle f, \psi_j \rangle \psi_j$$

alors l'image rendue par f sera en général indistinguable de celle associée à f_s^* pour l'œil humain si on prend s de l'ordre de 10Ko pour une image de taille $N \sim 1Mo$ (une bonne illustration de ce phénomène se trouve par exemple dans la Figure 1.1 du livre [4] où seulement 1% des plus grands coefficients d'ondelette sont gardés entre l'originale et l'image compressée qui sont néanmoins de même qualité pour l'œil humain). Donc, aussi surprenant que cela puisse paraître (vue la complexité et la diversité a priori des photos), les photos peuvent être bien approchées par des objets de "petite" dimension. L'hypothèse de parcimonie est donc "approximativement" vérifiée. S'il existe des procédures stables au sens de la Définition 2.3 alors on pourra reconstruire les images obtenues par l'appareil photo à un pixel. Etant donné que la prise de mesures sera très certainement bruitées, on demandera aussi à cet algorithme d'être robuste au bruit.

Les bases utilisées pour le traitement d'images qui permettent une écriture parcimonieuse des images sont les bases d'ondelettes. En quelques mots et pour faire simple, elles sont construites à partir d'une fonction mère, par exemple la fonction de Haar :

$$\Psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Les vecteurs de la base d'ondelette sont ensuite obtenues par translations / dilations de l'ondelette mère :

$$\Psi_i^j(x) = \sqrt{2^j} \Psi(2^j x - i), \quad i = 0, 1, \dots, n-1; j = 0, 1, \dots, \log_2 n - 1$$

pour $n = N_1 = N_2$.

On peut indexer cette base de la manière suivante

$$w(2^j + i, x) = \Psi_i^j(x).$$

La famille $(w(k, \cdot) : 1 \leq k \leq n)$ est une base ortonormale de $L^2(\{0, \dots, n-1\})$. Pour obtenir une base de $L^2(\{0, \dots, n-1\}^2)$, on peut prendre le produit tensoriel de cette base. Ainsi, la transformée en ondelette d'un signal $f \in \{0, 1, \dots, n-1\}^2$ est en tout point $(x, y) \in \{0, \dots, n-1\}^2$,

$$f(x, y) = \sum_{k=0}^{n-1} \sum_{m=0}^{n-1} fw(k, m) \times w(k, x) \times w(m, y)$$

où pour tout $(k, m) \in \{0, \dots, n-1\}^2$,

$$fw(k, m) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \times w(k, x) \times w(m, y)$$

Dans la Figure 8, on observe que la grande majorité des coefficients d'ondelettes sont négligeables devant les quelques milliers de plus grands.

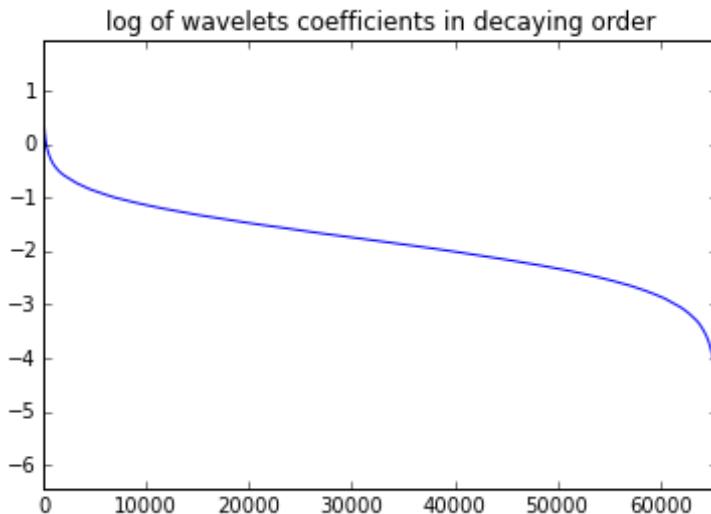


FIGURE 8 – Logarithme des coefficients d'ondelette d'une image ordonné par ordre décroissant. La grande majorité des coefficients sont très petits devant les 4.000 premiers.

On peut alors seuiller les plus petits coefficients à zéro et ne garder que la "poignée" de plus grands. C'est ce qui est fait dans la Figure 9.

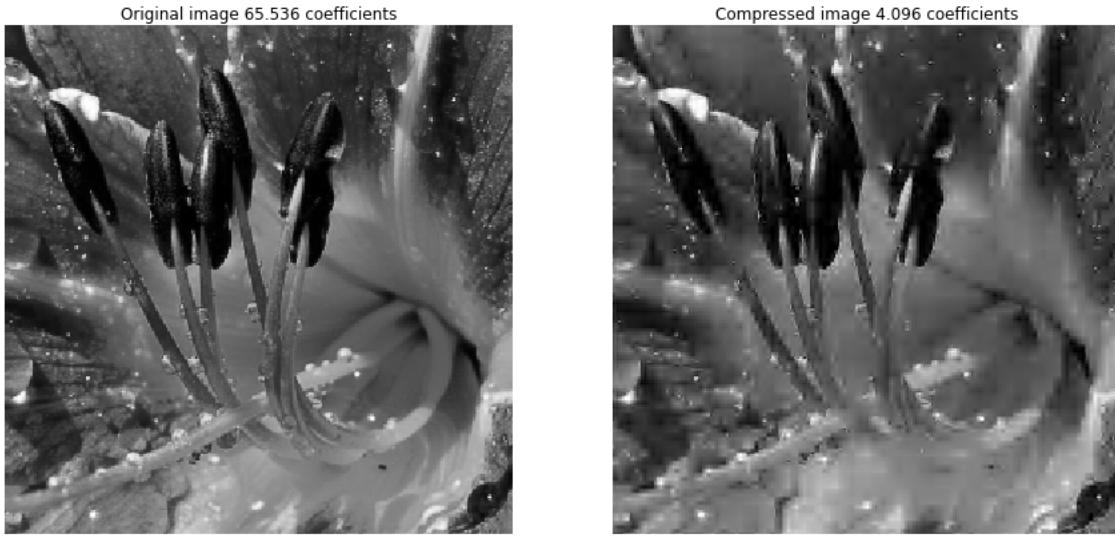


FIGURE 9 – A gauche, l'image d'origine. A droite, l'image obtenue par seuillage dur (seuls 16% des coefficients sont conservés).

L'hypothèse de parcimonie semble donc vérifiée. La construction de l'appareil photos à un pixel a donné les résultats de reconstruction donnés dans la Figure 10.

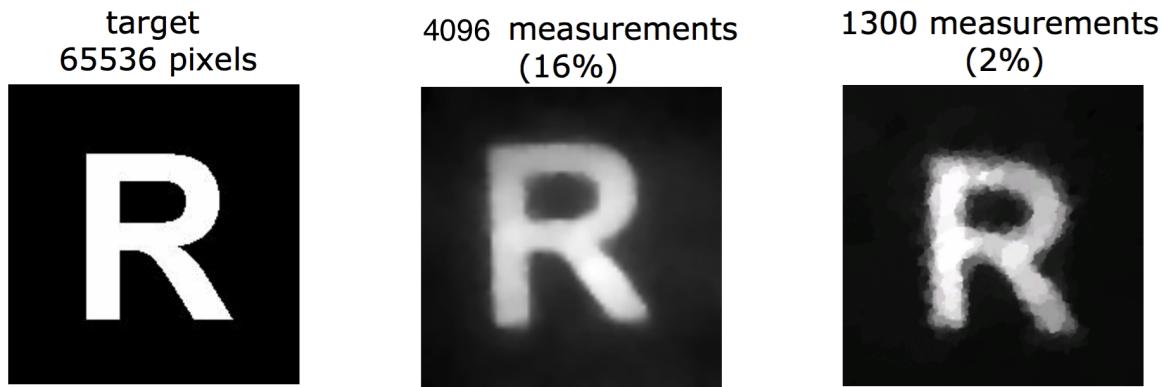


FIGURE 10 – Exemple de reconstruction d'images à partir de mesures de l'appareil photos à un pixel

4 Le problème du Compressed Sensing pour des mesures de Fourier partielles – notion de mesures incohérentes

Etant donné un signal $x \in \mathbb{C}^N$ et un ensemble $I \subset \{1, \dots, N\}$ de fréquences, il est en général impossible de reconstruire x uniquement à partir des coefficients de Fourier ($\widehat{x}_i : i \in I$) à moins que $I = \{1, \dots, N\}$ (car la transformée de Fourier est une bijection de \mathbb{C}^N).

Néanmoins, si x est s -sparse alors on retombe dans le cadre du Compressed Sensing : reconstruire un signal structuré (ici sparse dans la base canonique) à partir d'une “information incomplète” sur ce signal (ici un petit nombre de ses coefficients de Fourier).

Pour fixer les notations, on introduit la matrice de Fourier (transformée de Fourier discrète) :

$$\Gamma : \begin{cases} \mathbb{C}^N & \rightarrow \mathbb{C}^N \\ x & \rightarrow \widehat{x} \end{cases} \text{ où } \Gamma = \left(\frac{w^{(p-1)(q-1)}}{\sqrt{N}} \right)_{1 \leq p, q \leq N} \text{ et } w = \exp(-2\mathbf{i}\pi/N). \quad (4.1)$$

On note par $\bar{\Gamma}_1, \dots, \bar{\Gamma}_N$ les vecteurs lignes de Γ . On a donc $\widehat{x}_i = \langle \Gamma_i, x \rangle = \bar{\Gamma}_i^\top x$ pour tout $i = 1, \dots, N$. En particulier, l'ensemble les observations $\widehat{x}_i, i \in I$ sont bien des mesures linéaires du signal x .

Pour un sous-ensemble $I \subset \{1, \dots, N\}$ de fréquences, on dispose des données $(\widehat{x}_i : i \in I)$; en terme matriciel, on dispose de la donnée $\Gamma_I x$ où

$$\Gamma_I : \begin{cases} \mathbb{C}^N & \rightarrow \mathbb{C}^I \\ x & \rightarrow (\widehat{x}_i : i \in I). \end{cases}$$

C'est donc ici la matrice de mesures introduite dans (1.3) où les vecteurs de mesures sont les $\bar{\Gamma}_i$ pour $i \in I$ – le nombre de mesures est donc ici $m = |I|$. Dans ce cas, la matrice de mesures est obtenue en extrayant les lignes d'indices dans I de la matrice (carrée) de Fourier $\Gamma \in \mathbb{C}^{N \times N}$.

Le problème mathématique s'énonce ici de la manière suivante : sachant qu'on observe $m = |I|$ coefficients de Fourier d'un signal x vivant dans un espace de grande dimension N mais dont le support (dans la base canonique) est de petite taille s , quelles sont les conditions sur les trois paramètres s, m et N et quels sont les ensembles $I \subset \{1, \dots, N\}$ de fréquences à transmettre de telle sorte qu'il soit possible de reconstruire exactement tout signal s -sparse x seulement à partir de ces m coefficients de Fourier $\Gamma_I x$?

On montrera dans la suite du cours qu'une sélection aléatoire des fréquences à transmettre permet de prendre m (le nombre de mesures) de l'ordre de s (la taille des supports des signaux) à des termes logarithmiques près.

4.1 Introduction à la notion de mesures incohérentes

La matrice de Fourier Γ peut ici être remplacée par toute matrice Γ telle que :

1. Γ est une isométrie : $\forall x \in \mathbb{C}^N, \|\Gamma x\|_2 = \|x\|_2$,
2. les entrées de Γ sont telles que $|\Gamma_{pq}| \leq C/\sqrt{N}$ (où C est une constante absolue). C'est-à-dire que si $\bar{\Gamma}_1, \dots, \bar{\Gamma}_N$ sont les vecteurs lignes de Γ et (e_1, \dots, e_N) est la base canonique de \mathbb{C}^N alors pour tout $1 \leq p, q \leq N$,

$$|\langle \Gamma_p, e_q \rangle| \leq \frac{C}{\sqrt{N}}. \quad (4.2)$$

La propriété (4.2) s'avère être un point clef dans le choix des mesures. Quand la propriété (4.2) est satisfaite on dit que les bases orthonormales $(\bar{\Gamma}_1, \dots, \bar{\Gamma}_N)$ et (e_1, \dots, e_N) sont incohérentes. Géométriquement, cela signifie qu'une base est faîte d'éléments diagonaux de l'autre (et réciproquement). Cette idée apparaît constamment en CS dans le choix des vecteurs de mesures : il "faut" que les vecteurs de mesure soient incohérents avec la base dans laquelle la parcimonie est exprimée. Ici la parcimonie est exprimée dans la base canonique, on va alors chercher des mesures incohérentes avec la base canonique. Si la parcimonie avait été exprimée dans la base de Fourier par exemple, alors on aurait pris des vecteurs de mesure de la base canonique. On verra plus tard qu'un point fort des mesures aléatoires est, qu'étant donné une base, avec grande probabilité le vecteur aléatoire est incohérent avec les éléments de cette base. Bien sûr n'importe quel vecteur aléatoire ne satisfiera pas cette propriété (par exemple des Dirac en la base canonique) mais pour

le cas typique d'un vecteur Gaussien (càd avec des entrées iid gaussiennes centrées réduites), on observera ce phénomène ainsi que pour d'autres exemples de vecteurs aléatoires.

La notion d'incohérence étend la dualité entre temps et fréquence ; un signal très localisé en temps sera très étalé en fréquence (c'est l'idée du principe d'incertitude qui sera détaillé plus tard dans le cours). Ici un objet sparse (dans la terminologie du traitement du signal, on dit *localisé* en temps) dans une base ou un dictionnaire $\{f_1, \dots, f_M\}$ doit être bien étalé (ce qui est l'opposé de sparse) dans le dictionnaire formé des mesures $\{X_1, \dots, X_m\}$. En particulier, les vecteurs de mesures $\{X_1, \dots, X_m\}$ ont une représentation bien étalée dans la base où est exprimée la parcimonie.

On peut définir formellement cette notion d'incohérence. Supposons que nous disposons de deux bases $\mathcal{B}_f := (f_1, \dots, f_N)$ et $\mathcal{B}_\Gamma := (\Gamma_1, \dots, \Gamma_N)$ de \mathbb{R}^N . Dans la base \mathcal{B}_f les signaux x que nous souhaitons reconstruire ont une représentation parcimonieuse ; \mathcal{B}_f est donc un bon choix de base pour représenter ces signaux. On l'appellera la base de représentation. La seconde base \mathcal{B}_Γ est utilisée pour les prises de mesure ; càd on va choisir les vecteurs de mesures dans \mathcal{B}_Γ mais comme on ne souhaite faire seulement qu'un petit nombre de mesures on n'en choisit que m parmi les N vecteurs dans \mathcal{B}_Γ . La base \mathcal{B}_Γ est appelée base de mesures.

Remarque 4.1 Si \mathcal{B}_f est une base orthonormale de \mathbb{R}^N et $\mathcal{B}_0 = (e_1, \dots, e_N)$ est la base canonique, alors tout signal x s'écrit dans \mathcal{B}_f et \mathcal{B}_0 par

$$x = \sum_{j=1}^N \langle x, f_j \rangle f_j = \sum_{j=1}^N x_j e_j.$$

On dit alors que $(x_j)_{j=1}^N$ est la représentation de x dans le domaine temporel et $(\langle x, f_j \rangle)_{j=1}^N$ est la représentation de x dans la \mathcal{B}_f -domaine. C'est la terminologie employée en traitement du signal qui étend le jargon introduit pour l'analyse de Fourier : quand \mathcal{B}_f est la base de Fourier (dans ce cas, l'espace des signaux est \mathbb{C}^N plutôt que \mathbb{R}^N) alors le \mathcal{B}_f -domaine est plus connu sous le nom de domaine de fréquences.

Définition 4.2 Soit $(\mathcal{B}_f, \mathcal{B}_\Gamma)$ un couple de base orthonormales de \mathbb{R}^N . La cohérence entre la base de représentation \mathcal{B}_f et la base de mesures \mathcal{B}_Γ est

$$\mu(\mathcal{B}_f, \mathcal{B}_\Gamma) = \max_{1 \leq p, q \leq N} |\langle \Gamma_p, f_q \rangle|.$$

On a toujours $\mu(\mathcal{B}_f, \mathcal{B}_\Gamma) \in [1/\sqrt{N}, 1]$. En effet, pour tout p, q , $|\langle \Gamma_p, f_q \rangle| \leq \|\Gamma_p\|_2 \|\mathbf{f}_q\|_2 = 1$ et par Parseval, pour tout p , $\sum_q |\langle \Gamma_p, f_q \rangle|^2 = 1$ donc nécessairement au moins un terme $|\langle \Gamma_p, f_q \rangle|^2$ de cette somme est plus grand que $1/N$.

En CS, on s'intéresse principalement aux couples de bases de représentation et de mesures ayant une faible cohérence, de l'ordre de $1/\sqrt{N}$. C'est en particulier le cas du couple (base canonique, base de Fourier) et c'est une des raisons pour laquelle la reconstruction de vecteurs s -sparses dans la base canonique peut se faire à partir seulement d'un petit nombre de leurs coefficients de Fourier.

Dans le cas inverse de bases ayant une forte cohérence ; par exemple, reconstruire des vecteurs s -sparse dans la base canonique par des mesures obtenues en choisissant des vecteurs de mesures aussi dans la base canonique est voué à l'échec car, il faudra $m = N$ mesures alors que pour le même problème seulement $m \sim s$ mesures de Fourier sont suffisantes. On renvoie le lecteur intéressé à l'article [2] pour des résultats théoriques liant nombre de mesures et la cohérence entre base de représentation et base de mesures.

5 Compressed sensing pour des données Radar

De nombreuses données Radar peuvent être interprétées comme des mesures linéaires d'une scène données. On peut alors dans certains cas utiliser les idées développées en CS pour traiter ses données et avoir une idée des signaux à émettre pour gagner en précision et en temps d'émission.

Dans cette section, on considère le cadre d'un radar au sol qui a pour but d'identifier les avions en vol dans son périmètre de surveillance. L'objectif du Radar est donc d'identifier la position et la vitesses de ces avions. Pour cela, le Radar balaie son espace de surveillance et émet des signaux pour déterminer la distance et la vitesse des cibles potentielles dans son champs de surveillance. Ainsi à partir de l'observation continue de la distance et de la vitesse de ces cibles, on peut déterminer leurs localisations et leurs vitesses.

A chaque émission d'un signal, le radar doit déterminer la position et la vitesse de chacune des cibles présentes. Il se trouve que la distance d'une cible peut être obtenue par la présence d'un décalage sur l'onde émise et la vitesse est obtenue par effet Doppler qui induit une modulation en fréquence de l'onde émise. Une cible peut donc être repérée (sa position et sa vitesse) par un décalage en temps et une modulation en fréquence : connaissant ce décalage en temps et cette modulation en fréquence on peut en déterminer la distance (au Radar) et la vitesse de la cible.

Une scène peut alors être décrite par quelques cibles induisant toutes un décalage en temps et une modulation en fréquence. On discrétise le décalage et la modulation (ce qui induira de résoudre un problème d'interpolation en post-traitement lorsque les "vrais" décalages et modulations ne se trouvent pas dans la grille discrète – on ne traitera pas de ce problème ici).

On introduit alors deux opérateurs $T_k : \mathbb{C}^N \mapsto \mathbb{C}^N$ et $M_l : \mathbb{C}^N \mapsto \mathbb{C}^N$ pour tout $0 \leq k, l \leq N-1$ tels que pour tout $z \in \mathbb{C}^N$ et tout $0 \leq j \leq N-1$

$$(T_k z)_j = z_{j-k \text{ mod } N} \text{ et } (M_l z)_j = \exp\left(\frac{2i\pi}{N} jl\right) z_j \quad (5.1)$$

où "mod N " signifie *modulo N* . L'opérateur T_k est un opérateur de décalage temporel ; il réalise un *shift* de longueur k . L'opérateur M_l réalise une modulation en fréquence qui est dû à l'effet Doppler d'une cible en mouvement.

Une scène peut donc être modélisée par un signal de la forme

$$x = \sum_{0 \leq k, l \leq N-1} x_{k,l} T_k M_l \in \mathbb{C}^{N \times N} \quad (5.2)$$

où les éléments non nuls du vecteur de coefficients $(x_{k,l})_{0 \leq k, l \leq N-1}$ donne les décalages temporels et les modulations de fréquence des cibles en présence (dont on peut en déduire leurs distances au Radar et leurs vitesses).

La parcimonie vient du fait que l'espace aérien est la plupart du temps dégagé et qu'au plus une dizaine de cibles est présente. Le signal x exprimé dans le dictionnaire $\mathcal{F} := \{T_k M_l : 0 \leq k, l \leq N-1\}$ est donc parcimonieux. On peut se donner une représentation matricielle des éléments de

ce dictionnaire :

$$T_k M_l = \begin{pmatrix} 0 & 0 & \cdots & 0 & \exp\left(\frac{2i\pi lk}{N}\right) & 0 & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & \exp\left(\frac{2i\pi l(k+1)}{N}\right) & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \ddots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \exp\left(\frac{2i\pi l(N-1)}{N}\right) \\ 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots \\ 0 & \exp\left(\frac{2i\pi l}{N}\right) & \cdots & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \cdots & \exp\left(\frac{2i\pi l(k-1)}{N}\right) & 0 & 0 & 0 & \cdots \end{pmatrix}$$

à partir de l'expression, pour tout $0 \leq p, q \leq N - 1$,

$$(T_k M_l)_{p,q} = \begin{cases} 0 & \text{si } p + k \bmod N \neq q \\ \exp\left(\frac{2i\pi lq}{N}\right) & \text{si } p + k \bmod N = q. \end{cases}$$

On voit donc que la signal x qu'on souhaite estimer ici est une matrice $N \times N$ parcimonieuse (lorsqu'elle est exprimée dans le dictionnaire \mathcal{F}). La démarche naturelle en CS serait alors de construire des "matrices" de mesures de taille $N \times N$ et d'observer le produit scalaire entre le signal x et ces matrices. Malheureusement, cela n'a pas de sens physique. Ici, comme dans beaucoup d'autres problèmes le choix des mesures est limité par le problème (et même dans certain problème – cf. Section 7 – on n'a pas le choix des mesures).

Dans le cas de signaux Radar, les observations (qui sont naturellement bruitées par l'environnement et les composants même du Radar mais qui pour simplifier l'exposition sont considérés comme sans bruit) sont de la forme

$$y = xg \tag{5.3}$$

où $g \in \mathbb{C}^N$ est un vecteur fait des impulsions émises par le Radar. Ce vecteur est laissé au libre choix du constructeur de Radar (dans les limites physiques du Radar). Le choix de g doit être fait de telle sorte que x puisse être reconstruit à partir de $y = xg$. Ici on n'a pas le choix sur le nombre de mesure qui est forcément N tandis que la dimension de x est N^2 . Un choix populaire est le vecteur *Alltop* $g \in \mathbb{C}^N$ tel que

$$g_l = \exp\left(\frac{2i\pi l^3}{N}\right) \text{ pour tout } l = 0, \dots, N - 1.$$

Remarque 5.1 *L'exemple du Radar, comme les exemples qui suivent, mets en avant une situation assez classiques en CS concernant le choix des vecteurs de mesures. Dans la modélisation proposée en Section 2, le statisticien a le choix des vecteurs de mesures – la notion d'incohérence entre vecteurs de mesures et base de représentation peut le guider dans son choix (et les cours suivant donneront d'autres idées sur le choix de ces vecteurs de mesures) – mais dans le cas du radar le choix de ces vecteurs de mesures est très limité, ici même le nombre de mesures est imposé par la physique du problème. Dans ce cas, il ne reste plus qu'à espérer que ces mesures soient "naturellement" adaptées au problème du CS en particulier que ces mesures sont suffisamment incohérentes avec la base de représentation.*

Suivant la remarque 5.1, on doit vérifier que les mesures obtenues pour ce problème de détection Radar ont une chance d'être incohérentes avec la base de représentation. Pour ce

faire, on va réécrire le vecteur observé y sous forme d'une mesure linéaire du vecteur sparse $X = (x_{k,l})_{0 \leq k,l \leq N-1}$ (qui est la représentation de x dans \mathcal{F}) par

$$y = xg = A_g X \text{ où } A_g \in \mathbb{C}^{N \times N^2}$$

et les N^2 colonnes de A_g sont les vecteurs $T_k M_l g \in \mathbb{C}^N$ pour tout $0 \leq k, l \leq N-1$. En particulier, si g est de la forme

$$g = \begin{pmatrix} \exp\left(\frac{2i\pi p_0 \times 0}{N}\right) \\ \exp\left(\frac{2i\pi p_1 \times 1}{N}\right) \\ \vdots \\ \exp\left(\frac{2i\pi p_j \times j}{N}\right) \\ \vdots \\ \exp\left(\frac{2i\pi p_{N-1} \times (N-1)}{N}\right) \end{pmatrix} = \left(\exp\left(\frac{2i\pi p_l \times l}{N}\right) \right)_{l=0}^{N-1}$$

où $p_0, p_1, \dots, p_{N-1} \in \{0, 1, \dots, N-1\}$ sont des fréquences émises par le Radar (laissé au choix de l'opérateur)

6 Compressed sensing pour la reconnaissance de visages

Dans cette section, on présente le problème de reconnaissance de visage qu'on écrira comme un problème de CS. L'approche utilisée ici est appelée *sparse representation-based classification (SRC)*. On pourra trouver une exposition détaillée de cette approche dans le chapitre 12 de [3].

On se donne un jeu de données du type $\mathcal{D} := \{(\phi_j, \ell_j) : 1 \leq j \leq N\}$ où :

1. $\phi_j \in \mathbb{R}^m$ est une représentation vectorielle de la j -ième image, par exemple, on peut penser à ϕ_j comme étant le vecteur (concaténé) des pixels d'une image de taille $p \times q$ et donc $m = pq$,
2. $\ell_j \in \{1, \dots, C\}$ est un label indiquant le numéro de l'individu représenté dans la j -ième image.

Un même individu peut être représentés plusieurs fois sous différents angles, différentes expositions de lumière, etc. dans la base de données \mathcal{D} .

On se donne une nouvelle image $y \in \mathbb{R}^m$ et on souhaite affecter (de manière automatique à partir des données dans \mathcal{D}) à y un label parmi $\{1, \dots, C\}$ indiquant le numéro de l'individu qui est représenté dans la nouvelle image y . C'est le problème de reconnaissance de visages qui peut être vu comme un problème de classification multi-classe.

Dans la suite, on va réécrire ce problème dans le cadre du CS. Des résultats (cf. [1, 5]) qui ont une grande influence dans la communauté de reconnaissance de visages ont observé que si le nombre d'images d'un même sujet était assez élevé et qu'il y avait suffisamment de diversité parmi ces photos (en termes d'exposition à la lumière et d'angles de prise de vue) alors ces images seront proches d'un espace linéaire de faible dimension dans l'espace de grande dimension m . Pour une certaine représentation ϕ des images cette dimension peut même être prise égale à 9 (cf. [1]). Ainsi, si le jeu de données \mathcal{D} est suffisamment riche en taille et varié en exposition et angles de prise de vue, alors on peut espérer que la nouvelle image y d'un sujet $i \in \{1, \dots, C\}$ (pour un certain i à déterminer) sera de la forme :

$$y \approx \sum_{j:\ell_j=i} \phi_j x_j. \tag{6.1}$$

On fait donc l'hypothèse que la nouvelle image (représentée par y) est une combinaison linéaire des images du sujet i déjà présentes dans le jeu de données \mathcal{D} . On peut réécrire cette hypothèse dans la formalisme du CS :

$$y = \Phi x + \xi \quad (6.2)$$

où ξ est vu comme un terme de bruit (dû à l'approximation linéaire de y par l'espace engendré par $\{\phi_j : \ell_j = i\}$) et

1. $\Phi := [\Phi_1 | \Phi_2 | \cdots | \Phi_C]$ où $\Phi_i = [\phi_j : \ell_j = i]$ pour tout $i \in \{1, \dots, C\}$. C'est-à-dire, on concatène les images de l'individu i présentes dans la base de données \mathcal{D} dans la matrice Φ_i – en particulier, s'il y a n_i photos de l'individu i dans \mathcal{D} alors Φ_i sera de taille $m \times n_i$. Ensuite on concatène les Φ_i pour $i = 1, \dots, C$ dans la matrice Φ qui est donc de taille $m \times N$.
2. le vecteur $x \in \mathbb{R}^N$ est de la forme par bloc

$$x = [\mathbf{0}^\top | \mathbf{0}^\top | \cdots | \mathbf{x}_i^\top | \mathbf{0}^\top | \cdots | \mathbf{0}^\top]^\top$$

où \mathbf{x}_i est la restriction de x au bloc de colonnes associé à Φ_i dans Φ .

On voit alors que le problème de reconnaissance de visage peut se mettre sous la forme du modèle bruité en CS vu que le signal x à retrouver est sparse dans la base canonique. Ici les vecteurs de mesures sont les lignes de Φ qui sont formées par les représentations des photos de la base de données.

Ici le problème qu'on souhaite principalement résoudre est la reconstruction du support de x qui permet d'identifier l'individu représenté dans y : si on construit un estimateur \hat{x} de x dont le support est principalement contenu dans le bloc associé à Φ_i dans Φ alors on pourra conclure que y représente l'individu i .

La construction de la matrice de mesure dépend donc des observations contenues dans la base de données \mathcal{D} . Soit cette base est fournies à l'avance et on n'a aucun moyen d'influencer sa construction. Soit, on peut la construire soit même comme dans les Figures 11 et Figure 12.

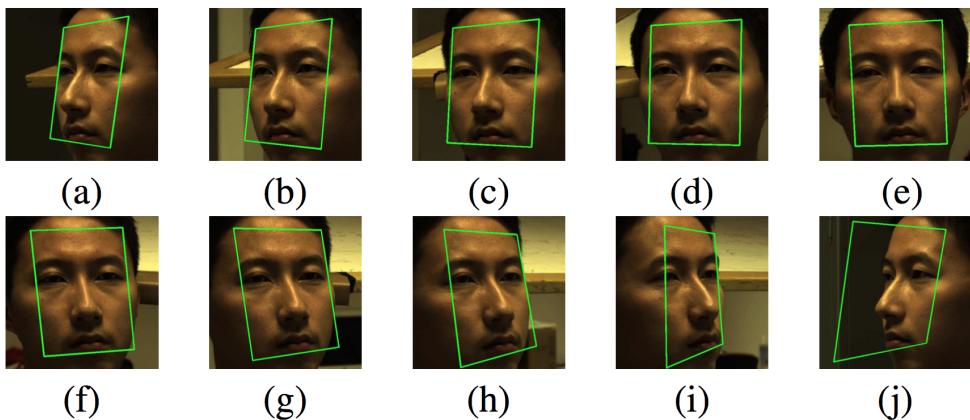


FIGURE 11 – Pour le problème de reconnaissance de visages, la construction de la base de données est équivalente à la construction de la matrice de mesures en CS : ici, prises de vue sous différents angles.



FIGURE 12 – Pour le problème de reconnaissance de visages, la construction de la base de données est équivalente à la construction de la matrice de mesures en CS : ici, prises de vue sous différentes luminosités.

7 Compressed sensing sur des données financières

Imaginons que nous ayons accès à la valeur d'un porte-feuille d'actions (ou d'autres actifs financiers) en temps réel, on note ces valeurs prises, par exemple, toutes les minutes, y_1, y_2, \dots . On suppose aussi que ce porte-feuille n'est pas très actif et que sur une semaine sa structure reste la même. Finalement, grâce aux données Bloomberg et Reuters, on a accès aux valeurs instantanées des actions du marché sur lequel évolue ce porte-feuille (cf. Figure 13).

			90 Save Defaults	90 News	90 Feedback		Global Commodity Prices	
	Movers	Units	Chg NY 14:30	Cal Spreads	Avg	Performance	%YTD	USD
1)	Energy	Units	2Day	Price	Net Chg	%Chg	Time	%YTD
10)	NYMEX WTI Crude	d \$/bbl	✓ ↗	88.70	-0.58	-0.65%	9:03	-10.25%
11)	ICE Brent Crude	d \$/bbl	✓ ↗	111.19	-0.51	-0.46%	9:03	+3.55%
12)	NYMEX Gasoline	d USD/gal	✓ ↗	273.43	-2.02	-0.73%	9:03	+1.79%
13)	NYMEX Heat Oil	d USD/gal	✓ ↗	306.70	-0.81	-0.26%	9:03	+4.50%
14)	ICE Gasoil	d \$/mt	✓ ↗	952.50	-1.00	-0.10%	9:03	+2.90%
15)	NYMEX Nat Gas	d \$/MMBtu	✓ ↗	3.756	+0.037	+0.99%	9:03	+25.66%
2)	Metals							
20)	Spot Gold	\$/t oz	✓ ↗	1732.10	+0.38	+0.02%	9:13	+10.68%
21)	Spot Silver	\$/t oz	✓ ↗	33.12	+0.00	+0.00%	9:13	+18.97%
22)	Spot Platinum	\$/t oz	✓ ↗	1575.63	-2.33	-0.15%	9:13	+13.00%
23)	Spot Palladium	\$/t oz	✓ ↗	642.60	+0.90	+0.14%	9:10	-1.61%
24)	LME 3mth Aluminium	d \$/mt	✓ ↗	1977.00 y	+26.00	+1.33%	11/19	-2.13%
25)	LME 3mth Copper	d \$/mt	✓ ↗	7804.00 y	+199.00	+2.62%	11/19	+2.68%
3)	Agriculture							
30)	CBOT Corn	d USD/bsh	✓ ↗	742.00	-0.50	-0.07%	9:03	+14.23%
31)	CBOT Wheat	d USD/bsh	✓ ↗	855.00	-2.75	-0.32%	9:02	+28.61%
32)	CBOT Soybeans	d USD/bsh	✓ ↗	1391.25	-3.50	-0.25%	9:03	+16.08%
33)	ICE Coffee	d USD/lb	✓ ↗	156.45	-0.95	-0.60%	9:03	-33.72%
34)	ICE Sugar	d USD/lb	✓ ↗	19.81	-0.13	-0.65%	9:03	-14.98%
35)	ICE Cotton	d USD/lb	✓ ↗	72.00	-0.06	-0.08%	9:02	-21.63%

FIGURE 13 – Données financières accessibles sur le site de *Bloomberg*. Ici la matrice de mesures est construite par les valeurs prises par les actions au fil du temps.

L'action j a pour cotation $x_{i,j}$ à l'instant i . Les données à notre disposition sont donc de la forme :

$$\begin{aligned}
 t = 1 : \quad & y_1 : \text{valeur du porte-feuille} \quad (x_{1,j})_{j=1}^N : \text{cotations des actions} \\
 t = 2 : \quad & y_2 : \text{valeur du porte-feuille} \quad (x_{2,j})_{j=1}^N : \text{cotations des actions} \\
 & \dots \\
 t = m : \quad & y_m : \text{valeur du porte-feuille} \quad (x_{m,j})_{j=1}^N : \text{cotations des actions}
 \end{aligned}$$

On cherche à savoir comment est structuré ce porte-feuille ; càd à identifier quelles sont les actifs qui le constituent et en quelle quantité.

On peut poser ce problème dans la cadre du CS. On note $y = (y_j)_{j=1}^m$ les valeurs prises par le porte-feuille et par $A = (x_{i,j} : 1 \leq i \leq m, 1 \leq j \leq N)$ la matrice des cotations. On souhaite alors retrouver un $x \in \mathbb{R}^N$ tel que $y = Ax$.

L'idée que le porte-feuille d'actions ne contient en fait que très peu d'action différentes se traduit directement en une hypothèse de parcimonie sur x : on s'attend à ce que x ait un petit support.

Reconstruire le support de x et estimer les valeurs prises par x sur son support vont donc résoudre le problème qu'on souhaite traiter ici.

8 Notations

Pour tout $p > 0$, on définit les normes (quand $p \geq 1$) et quasi-normes (quand $0 < p < 1$), ℓ_p^d sur \mathbb{R}^d par

$$\|x\|_p = \left(\sum_{j=1}^d |x_j|^p \right)^{1/p}$$

pour tout $x \in \mathbb{R}^d$. L'espace \mathbb{R}^d muni de la norme $\|\cdot\|_p$ est noté ℓ_p^d . La norme $\|\cdot\|_p$ sera parfois appelée norme ℓ_p ou ℓ_p^d . On identifie alors la norme et l'espace muni de cette norme lorsqu'il n'y a pas d'ambiguïté. La boule unité de l'espace ℓ_p^d est

$$B_p^d = \{x \in \mathbb{R}^d : \|x\|_p \leq 1\}$$

et la sphère unité de ℓ_p^d est notée

$$S_p^{d-1} = \{x \in \mathbb{R}^d : \|x\|_p = 1\}.$$

Pour tout $I \subset \{1, \dots, N\}$, on note par $P_I x$ le vecteur de \mathbb{C}^N qui coïncide avec x sur les coordonnées dans I et qui est nul sur le complément de I . Pour tout $T \subset \mathbb{C}^N$, $P_I T = \{P_I t : t \in T\}$.

L'ensemble de tous les vecteurs s -sparses de \mathbb{R}^N est noté

$$\Sigma_s = \{x \in \mathbb{R}^N : \|x\|_0 \leq s\}.$$

Si $A \in \mathbb{R}^{m \times N}$, et $I \subset \{1, \dots, m\}$, $J \subset \{1, \dots, N\}$ alors

1. $A_{I \cdot}$ est la matrice extraite de A où seules les lignes d'indice dans I ont été gardées
2. $A_{\cdot J}$ est la matrice extraite de A où seules les colonnes d'indice dans J ont été gardées
3. A_{IJ} est la matrice extraite de A où seules les lignes d'indice dans I et les colonnes d'indice dans J ont été gardées.

Références

- [1] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2) :218–233, Feb 2003.
- [2] Emmanuel Candès and Justin Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3) :969–985, 2007.
- [3] Yonina Eldar and Gitta Kutyniok. Compressed sensing : Theory and applications. *Cambridge University Press*, 2012.
- [4] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.
- [5] A.S. Georghiades, P.N. Belhumeur, and D. Kriegman. From few to many : illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6) :643–660, Jun 2001.
- [6] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier/Academic Press, Amsterdam, third edition, 2009. The sparse way, With contributions from Gabriel Peyré.
- [7] Gabriel Peyré. <http://www.numerical-tours.com>.