THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
UFR des sciences fondamentales et appliquées
XLIM-SIC
(Diplôme National - Arrêté du 7 août 2006)

École doctorale : Sciences et ingénierie pour l'information, mathématiques - S2IM (Limoges)
Secteur de recherche : Traitemement du signal et des images

Présentée par :
Sylvain Rousseau

Détection de points d'intérêts dans une image multi ou hyperspectral par acquisition compressée

Directeur(s) de Thèse :
Philippe Carré, David Helbert

Soutenue le 02 juillet 2013 devant le jury

Jury :

Président	**Michel Berthier**	Professeur des Universités, Université de La Rochelle
Rapporteur	**Grégoire Mercier**	Professeur des Universités, Institut Télécom-Bretagne
Rapporteur	**Jocelyn Chanussot**	Professeur des Universités, INP de Grenoble
Membre	**Philippe Carré**	Professeur des Universités, Université de Poitiers
Membre	**David Helbert**	Maître de conférences, Université de Poitiers
Membre	**Michel Berthier**	Professeur des Universités, Université de La Rochelle
Membre	**Jacques Blanc-Talon**	Chargé de mission, DGA/DS/QIS, Bagneux

Pour citer cette thèse :
Sylvain Rousseau. *Détection de points d'intérêts dans une image multi ou hyperspectral par acquisition compressée* [En ligne]. Thèse Traitemement du signal et des images . Poitiers : Université de Poitiers, 2013. Disponible sur Internet <http://theses.univ-poitiers.fr>

THÈSE
pour l'obtention du Grade de

DOCTEUR DE L'UNIVERSITÉ DE POITIERS

(Faculté des Sciences Fondamentales et Appliquées)

Diplôme National - Arrêté du 7 août 2006

École Doctorale : Sciences et Ingénierie pour l'Information, Mathématiques

Secteur de Recherche : Traitement du signal et des images

Présentée par :

Sylvain Rousseau

Détection de points d'intérêt par acquisition compressée dans une image multispectrale

---

Directeur de thèse : Philippe Carré
Co-directeur de thèse : David Helbert

Soutenue le 2 juillet 2013
devant la Commission d'Examen

JURY

Grégoire Mercier	Professeur, Institut Télécom-Bretagne	Rapporteur
Jocelyn Chanussot	Professeur, INP Grenoble	Rapporteur
Michel Berthier	Professeur, Université de La Rochelle	Examinateur
Jacques Blanc-Talon	Chargé de mission DGA/DS/QIS	Examinateur
Philippe Carré	Professeur, Université de Poitiers	Examinateur
David Helbert	Maître de conférence, Université de Poitiers	Examinateur

Table des matières

Page

Guide de lecture **1**

1 Introduction **3**
1.1 Gradient de Di Zenzo et métriques riemanniennes 4
1.1.1 Gradient de Di Zenzo 4
1.1.2 Métriques 5
1.1.3 Projection sur un axe quelconque 7
1.1.4 Projection sur un sous-espace 8
1.1.5 Applications aux images multispectrales 9
1.2 Acquisition compressée 11
1.2.1 Principes 12
1.2.2 Mesures et reconstruction 13
1.2.3 Les algorithmes de reconstruction 14

2 Algorithmes de type Bregman **19**
2.1 L'itération de Bregman 20
2.1.1 Sous-différentielle 21
2.1.2 Distance de Bregman 21
2.1.3 L'itération de Bregman 22
2.1.4 Cas où $H(u) = \|Au - f\|_2^2$ 24
2.2 Algorithme de Bregman scindé 26
2.2.1 Opérations de seuillage 26
2.2.2 Technique de Bregman scindé 27
2.3 Bregman scindé contraint 29
2.4 Applications du Bregman scindé 31
2.4.1 Application au débruitage TV/L_2 anisotrope 31
2.4.2 Application au débruitage TV/L_2 isotrope 34
2.4.3 Application à la reconstruction TV d'une image 41
2.5 Conclusion 45

3 Détection compressée de signatures **47**
3.1 Détection de signature simple 48
3.2 Détection compressée de signature 51
3.2.1 Modèle de mesures 52
3.2.2 Détection compressée 53
3.2.3 Formulation du problème 55
3.3 Mise en œuvre et illustrations 58
3.4 Conclusion 64

4 Détection compressée de motifs **65**
4.1 Spectralisation suivant un motif 66
4.1.1 Motif 66
4.1.2 Motif normalisé 67
4.1.3 Opérateur de décalage 67
4.1.4 Spectralisation d'une image en niveaux de gris 69
4.1.5 Spectralisation d'une image multispectrale 71
4.2 Mesures sur une image spectralisée 72
4.2.1 Reconstruction de mesures naïves 73
4.2.2 Reconstruction de mesures glissées 74
4.2.3 Motif de mesures glissées optimal 75
4.3 Mise en œuvre et illustrations 80
4.3.1 Mise en œuvre 80
4.3.2 Tests sur l'image `peppers` 81
4.3.3 Tests sur l'image réelle `Bâche` 85
4.4 Conclusion 87

5 Matrices circulantes partiellement orthogonales **89**
5.1 Quelques définitions 90
5.1.1 Matrices circulantes 90
5.1.2 Matrices de Fourier 91
5.1.3 Opérateurs $\bullet$ et $\dagger$ 92
5.2 Matrices circulantes orthogonales 93
5.3 Matrices circulantes partiellement orthogonales 94
5.3.1 Analyse et synthèse 94
5.3.2 Exemple pratique 97
5.3.3 Exemple de « somme nulle » 99
5.4 Matrices circulantes par blocs, à blocs circulants partiellement orthogonales 100
5.4.1 Analyse et synthèse 100
5.4.2 Exemple pratique 105
5.5 Mise en œuvre 105

5.6 Applications 106
5.6.1 Sur l'image `peppers` 107
5.6.2 Sur l'image `bâche` 108
5.7 Conclusion 112

Conclusion **113**

Appendices **117**

A Produit de Kronecker **119**
A.1 Définition et propriétés immédiates 119
A.2 Application au calcul de la norme TV 120

Guide de lecture

Une image est une description du réel. D'abord monochrome, c'est à dire représentée en niveaux de gris, elle est ensuite couleur. Elle est composée de trois images monochromatiques chacune décrivant l'intensité de la scène perçue dans une bande passante précise. Avec l'évolution des technologies des dernières décennies, la conception de capteurs de plus en plus performants nous a permis de décrire une même scène pour différentes longueurs d'onde. Le nombre de bandes va d'une dizaine pour l'imagerie multispectrale classique jusqu'à plusieurs centaines pour les capteurs les plus sophistiqués. Le nombre de bandes croissant ainsi que la complexité des images nécessitent la conception de nouveaux algorithmes et l'élaboration de nouvelles méthodes de modélisation de ces données.

Cette thèse, financée par la DGA, a pour problématique la détection d'éléments particuliers dans une scène hyperspectrale. La quantité sans cesse croissante de données multispectrales et leur caractère extrêmement redondant suggèrent qu'une réduction de la dimension est possible. Dans ce but, l'utilisation de la géométrie riemannienne est la première approche envisagée dans cette thèse. Elle offre un point de vue différent qui permet de développer de nouveaux algorithmes [DZ86, SKM98].

D'un autre côté, il est question depuis un certain nombre d'années de changer le modèle d'acquisition des données plutôt que d'essayer de trouver des algorithmes pouvant gérer les grandes quantités de données générées par l'acquisition classique. C'est l'objet de l'acquisition compressée introduite par Candès, Tao et Donoho [CRT06, Don06] qui propose de faire l'acquisition d'une image en prenant un nombre de mesures réduits sur elle. L'acquisition génère donc moins de données mais elles sont mélangées et leur exploitation nécessite la résolution d'une minimisation complexe. D'abord appliquée aux images en niveaux de gris, elle est généralisée aux données multispectrales [DB09, LSKZ11]. Il existe également des mises en œuvre pratiques comme la *single pixel camera* ou le système CASSI [GJB+07].

Le premier chapitre commence par présenter la première approche envisagée. Elle s'appuie sur l'utilisation d'outils de géométrie riemannienne pour réaliser de la détection de contours dans une image multispectrale. Le tenseur de structure d'ordre 2 de Di Zenzo est réinterprété avec des concepts issus de la géométrie

riemannienne. Nous enchaînons par l'introduction des principes de l'acquisition compressée qui constituera notre deuxième approche. Nous présentons les problèmes que posent l'acquisition compressée notamment en termes de reconstruction par minimisation.

Le deuxième chapitre est dévolu à l'étude d'une famille d'algorithmes qui repose sur la distance de Bregman. L'intérêt de ces algorithmes est qu'ils sont simples, flexibles et efficaces. Ils vont nous permettre de résoudre les différents problèmes de minimisations que nous rencontrerons.

Le troisième chapitre présente l'idée de départ qui est au cœur du travail de cette thèse : la minimisation de Guo et Osher qui réalise une détection de signature dans une image multispectrale à l'aide d'une minimisation L_1. Après avoir fixé un modèle de prise de mesures sur une image multispectrale, nous étudions comment adapter la minimisation de Guo et Osher pour qu'elle fonctionne avec des données compressées et proposons une minimisation qui réalise la détection compressée de signature dans une image multispectrale. Nous comparons ensuite les résultats de la nouvelle minimisation avec la minimisation originale.

Le quatrième chapitre propose une extension de la détection compressée de signature qui réalise une détection de motif. Cette extension repose sur une transformation formelle de l'image appelée *spectralisation*. Nous mettons en évidence les problèmes posés par une telle approche avec notamment la reconstruction de mesures. Nous soulignons enfin la nécessité de choisir des mesures appropriées pour permettre une reconstruction efficace. Nous illustrons enfin la détection compressée de motif sur une image couleur et sur une image multispectrale.

Enfin, le dernier chapitre propose une nouvelle méthode pour générer des matrices circulantes partiellement orthogonales qui s'affranchit des contraintes de la méthode classique. À travers deux propositions, nous donnons une méthode de construction de telles matrices dans le cas des dimensions 1 et 2. L'utilisation de ces matrices permet d'illustrer la détection compressée de motif sur de plus grandes images.

Les principales contributions de cette thèse sont les suivantes. Nous présentons une nouvelle minimisation qui réalise la détection compressée de signature dans une image multispectrale en nous appuyant sur une minimisation originale de Guo et Osher. Nous introduisons une transformation formelle appelée *spectralisation* qui nous permet de proposer une minimisation réalisant une détection compressée de motif. Enfin, nous démontrons qu'il est possible de construire des matrices circulantes orthogonales de manière beaucoup moins contraignante que la méthode classique et nous donnons une méthode pour les générer.

Chapitre 1

Introduction

Sommaire

1.1 Gradient de Di Zenzo et métriques riemanniennes . . 4
1.1.1 Gradient de Di Zenzo 4
1.1.2 Métriques 5
1.1.3 Projection sur un axe quelconque 7
1.1.4 Projection sur un sous-espace 8
1.1.5 Applications aux images multispectrales 9
1.2 Acquisition compressée 11
1.2.1 Principes 12
1.2.2 Mesures et reconstruction 13
1.2.3 Les algorithmes de reconstruction 14

Une image multispectrale est une collection d'images monochromes de la même scène qui sont prises par différents capteurs, chacun sensible à une longueur d'onde différente. Chaque image monochrome est appelée une bande. Les images couleur constituent l'exemple le plus simple d'images multispectrales. Il s'agit de trois images monochromes prises à trois longueurs d'onde différentes : le rouge, le vert et le bleu. Une modélisation naturelle est de considérer l'image multispectrale comme un cube de données avec deux dimensions spatiales et une dimension spectrale comme représenté à la figure 1.1. La détection de points d'intérêt ou d'objets particuliers dans une image multispectrale est une question ouverte. De nombreuses techniques et outils ont été utilisés pour s'attaquer au problème.

Ce chapitre présente la première approche que nous avons envisagée pour notre problématique. Nous proposons d'utiliser le formalisme de la géométrie riemannienne et de la notion de métrique pour réaliser une détection de contours dans une image multispectrale. Ce travail a fait l'objet d'une publication dans l'article [RHCBT10].

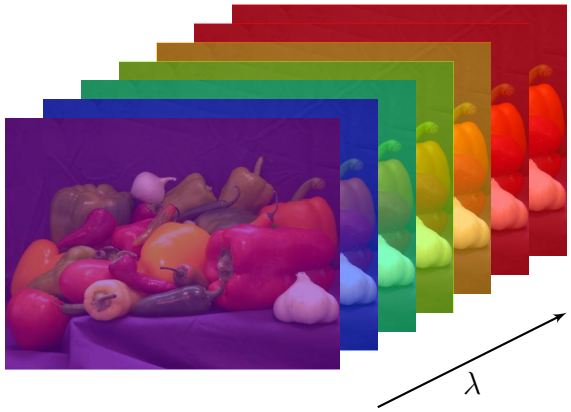


Figure 1.1 – Représentation des différentes bandes d'une image multispectrale

Ensuite, nous présentons la deuxième approche qui s'appuie sur l'acquisition compressée. D'abord appliquée aux cas des images monochromes puis à celui des images couleur, l'acquisition compressée est plus récemment appliquée à l'imagerie multispectrale. Nous en présentons les principes et nous formulons les problèmes de minimisation qui en découlent.

1.1 Gradient de Di Zenzo et métriques riemanniennes

Un des premiers articles utilisant des outils de géométrie riemannienne est l'article de Di Zenzo [DZ86]. Cet article propose de calculer le gradient d'une image multispectrale vue comme un champ de vecteurs. Classiquement, le calcul du gradient d'une image multispectrale peut être effectué de manière marginale : il s'agit d'appliquer des techniques éprouvées sur des images en niveaux de gris à chacune des bandes de l'image puis de combiner les résultats. L'inconvénient majeur est que les différents gradients n'interagissent pas et que même si les résultats sont combinés, nous perdons de l'information. Pour parer à cet inconvénient une solution est d'utiliser une généralisation du gradient d'une image en niveaux de gris.

1.1.1 Gradient de Di Zenzo

L'article [DZ86] présente donc le calcul de gradient en dimension quelconque : c'est le premier pas vers les outils riemanniens. Nous modélisons donc une image m-dimensionnelle comme une fonction f de $D \subset \mathbb{R}^2$ vers $\mathbb{R}^m$. L'ensemble $f(D)$ est alors une sous-variété de $\mathbb{R}^m$ si le jacobien est partout de rang 2 et que f est

injective.

$$\begin{array}{rccc} f: & D & \longrightarrow & \mathbb{R}^m \\ & (x,y) & \longmapsto & (f^1(x,y),\ldots,f^m(x,y)) \end{array}$$

En suivant Di Zenzo, nous introduisons le gradient,

$$T = \begin{pmatrix} \|f_1\|^2 & f_1 \cdot f_2 \\ f_1 \cdot f_2 & \|f_2\|^2 \end{pmatrix},$$

où $f_1 = (\frac{\partial f^1}{\partial x}, \ldots, \frac{\partial f^m}{\partial x})$ et $f_2 = (\frac{\partial f^1}{\partial y}, \ldots, \frac{\partial f^m}{\partial y})$. Nous remarquons qu'il peut être écrit de la manière suivante,

$$J^T J \quad \text{où} \quad J^T = \begin{pmatrix} \frac{\partial f^1}{\partial x} & \cdots & \frac{\partial f^m}{\partial x} \\ \frac{\partial f^1}{\partial y} & \cdots & \frac{\partial f^m}{\partial y} \end{pmatrix}.$$

Cette écriture nous permettra de voir que le tenseur de Di Zenzo est un cas très particulier où l'on choisit une métrique euclidienne.

1.1.2 Métriques

Nous avons donc un tenseur d'ordre 2 en chaque point de l'image. Pour faire une détection de contours, il suffit alors de maximiser ce tenseur. Nous remarquons ensuite que le fait d'avoir un tenseur d'ordre 2 en chaque point du domaine D définit une métrique sur la variété D. En fait, toute la modélisation de Di Zenzo peut être vue comme un calcul de métriques et de *pullback* de métriques sur des variétés riemanniennes [GHL04].

Nous pouvons en effet adapter la fonction f ci-dessus et considérer maintenant la fonction φ :

$$\begin{array}{rccc} \varphi: & D & \longrightarrow & \mathbb{R}^{m+2} \\ & (x,y) & \longmapsto & (x,y,\varphi^1(x,y),\ldots,\varphi^m(x,y)) \end{array}$$

Le rajout des coordonnées dans l'espace d'arrivée permet de s'assurer que φ est un plongement. Cette propriété est utile car elle permet de transporter une métrique de l'espace d'arrivée à l'espace de départ. De manière formelle, si nous avons deux variétés riemanniennes M^m et N^n, un plongement $\varphi : M^m \longrightarrow N^n$ et une métrique riemannienne g sur N^n alors nous pouvons construire une métrique riemannienne h sur M^m appelée le *pullback* de g par φ que nous notons $\varphi^* g$ défini par,

$$\varphi^* g = J^T g J,$$

où J est le jacobien de φ.

Il suffit donc de définir une métrique sur l'espace d'arrivée et, par *pullback*, nous en aurons une sur l'espace D. Voici quelques exemples de choix de métriques menant entre autres aux tenseurs de Di Zenzo. Supposons que $m = 1$, nous avons donc une image en niveaux de gris I. Alors,

$$\begin{array}{rccc} \varphi : & D & \longrightarrow & \mathbb{R}^3 \\ & (x,y) & \longmapsto & (x,y,I(x,y)) \end{array}$$

et

$$J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ I_x & I_y \end{pmatrix},$$

où $\frac{\partial I}{\partial x}$ est noté I_x et $\frac{\partial I}{\partial y}$ est noté I_y. Si $\mathbb{R}^3$ est muni de la (pseudo-)métrique,

$$g = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

alors nous avons,

$$\varphi^* g = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}.$$

Nous retrouvons donc exactement le tenseur de Di Zenzo qui est donc obtenu en choisissant la métrique g suivante sur l'espace d'arrivée,

$$g = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Supposons maintenant que nous avons une image couleur c'est à dire $m = 3$. Le plongement s'écrit alors,

$$\begin{array}{rccc} \varphi : & D & \longrightarrow & \mathbb{R}^5 \\ & (x,y) & \longmapsto & (x,y,I^1(x,y),I^2(x,y),I^3(x,y))' \end{array}$$

où I^1, I^2, I^3 sont respectivement les composantes rouge, verte et bleue de l'image. Par différents choix de la métrique de départ, nous retombons sur différentes techniques. Ainsi, pour une approche marginale, il suffit de choisir la métrique suivante,

$$g = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{1.1}$$

Nous avons alors,

$$\varphi^* g = \begin{pmatrix} (I_x^1)^2 & I_x^1 I_y^1 \\ I_x^1 I_y^1 & (I_y^1)^2 \end{pmatrix},$$

ce qui correspond au tenseur de Di Zenzo appliqué à la composante rouge.

Une manière de combiner l'information est de faire les calculs sur la luminance. Cette technique correspond également au choix d'une métrique particulière.

$$g = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{1.2}$$

La métrique issue du *pullback* s'écrit,

$$\varphi^* g = \begin{pmatrix} 1 + \sum_{i=1}^{3} (I_x^i)^2 & \sum_{i=1}^{3} I_x^i I_y^i \\ \sum_{i=1}^{3} I_x^i I_y^i & 1 + \sum_{i=1}^{3} (I_y^i)^2 \end{pmatrix}.$$

Nous proposons maintenant d'exploiter ce formalisme pour réaliser des opérations de projection locales de l'image.

1.1.3 Projection sur un axe quelconque

Jusqu'à présent, en choisissant les métriques riemanniennes (1.1) ou (1.2), nous avons projeté l'image sur l'axe rouge ou l'axe des gris. D'une manière générale, supposons que nous voulons projeter sur un axe quelconque u. Supposons que $u = (a, b, c)^T$ est unitaire. Soit Q la forme quadratique telle que $Q = uu^T$. Nous avons alors,

$$\begin{aligned} Q(u) &= u^T Q u \\ &= u^T u u^T u \\ &= 1, \end{aligned}$$

car $uu^T = 1$ et pour tout vecteur v orthogonal à u, nous avons,

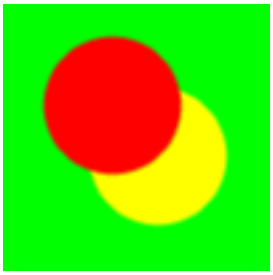
$$\begin{aligned} Q(v) &= u^T v u \\ &= 0, \end{aligned}$$

puisque $u^T v = 0$. La forme Q est donc la forme recherchée car elle projette orthogonalement sur l'axe u. En explicitant la forme Q car $u = (a, b, c)^T$ alors nous avons la métrique suivante,

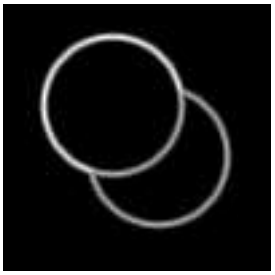
$$g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix}. \tag{1.3}$$

Nous pouvons alors retrouver les métriques (1.1) ou (1.2), en prenant respectivement $u = (1, 0, 0)^T$ et $u = \frac{1}{\sqrt{3}}(1, 1, 1)^T$.

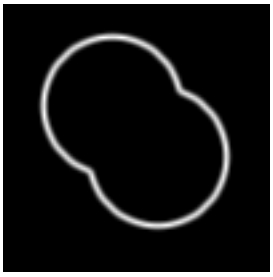
Nous allons maintenant illustrer l'usage de ces différentes métriques sur une image synthétique de test représentée à la figure 1.2a. Nous utilisons tout d'abord la métrique classique euclidienne. Tous les contours sont ainsi détectés comme le montre la figure 1.2b. La seconde métrique utilisée est une métrique marginale qui



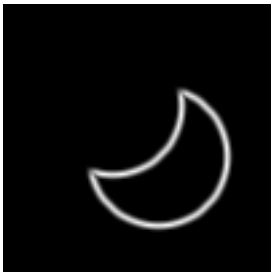
(a) Image test originale



(b) Métrique euclidienne



(c) Métrique marginale pour le rouge



(d) Métrique pour la luminance

Figure 1.2 – Détection de contours sur une image de test pour différentes métriques

projette sur l'axe rouge. Nous observons bien sur la figure 1.2c que la frontière entre le rouge et le jaune, qui ont la même composante rouge, n'est pas détectée. Nous retrouvons également la détection par luminance à la figure 1.2d où le rouge et le vert, qui ont même luminance, ne sont pas distingués. Nous proposons maintenant d'étendre cette approche à la projection sur un sous-espace quelconque.

1.1.4 Projection sur un sous-espace

Jusqu'alors, nous avons uniquement envisagé des métriques qui projetaient sur un axe précis que ce soit sur l'axe rouge pour une approche marginale ou sur l'axe des gris pour une approche par la luminance. La projection sur un axe précis est trop réductrice pour des images qui contiennent beaucoup de bandes. Une question naturelle est donc de trouver une forme quadratique qui projette sur un sous-espace donné. Dans le cas d'une image couleur, nous cherchons donc une forme quadratique Q qui est dégénérée selon une direction et euclidienne sur l'espace orthogonal à cette direction. Soit u cette direction dégénérée et v, w tels que (u, v, w) soit une base orthonormale. Soit Q une forme quadratique telle que $Q = vv^T + ww^T$. Alors nous avons, $Q(u) = 0$ et $Q(v) = Q(w) = 1$. La forme Q est donc bien celle que nous cherchons. Nous avons également la relation,

$$uu^T + vv^T + ww^T = I_3,$$

car (u, v, w) est une base orthonormale. Ainsi, Q s'écrit plus simplement,

$$Q = I_3 - uu^T.$$

Par exemple, dans le cas de la projection sur le plan de chrominance, nous supposons $u = \frac{1}{\sqrt{3}}(1, 1, 1)^T$. Nous avons alors,

$$uu^T = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

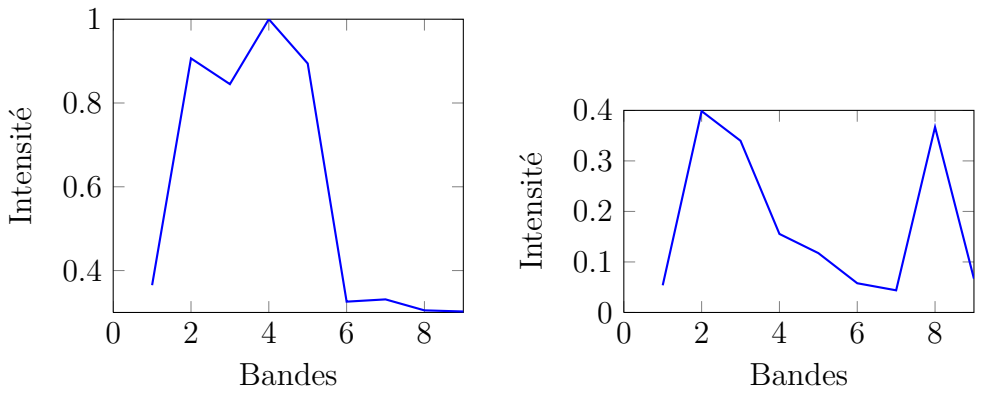
et donc,

$$g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \oplus \frac{1}{3}\begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}. \tag{1.4}$$

Finalement, nous pouvons décrire facilement une métrique qui projette sur un sous-espace donné dès lors que nous connaissons une base orthonormale de ce sous-espace.

1.1.5 Applications aux images multispectrales

Nous allons appliquer le résultat obtenu à la section précédente à la détection adaptative de contours dans une image multispectrale. Nous utilisons l'image AVIRIS, `Moffett Field` dont nous extrayons la région `Mowry` ainsi que 9 bandes. Nous voudrions effectuer une détection de contours sans faire apparaître la frontière qui sépare deux zones. Une zone est caractérisée par une signature spectrale



(a) Signature du banc de sable (b) Signature de l'eau

Figure 1.3 – Signatures de l'eau et des bancs de sable

précise. La figure 1.3 montre les deux signatures que nous voulons amalgamer. Nous proposons donc de projeter sur l'orthogonal de l'espace engendré par ces deux signatures. Supposons donc (u_1, u_2) soit une base orthonormale de l'espace engendré par les deux signatures. Nous appliquons alors l'algorithme avec la métrique,

$$Q = I_9 - u_1 u_1^T - u_2 u_2^T.$$

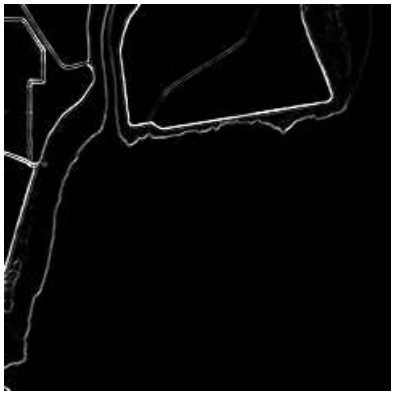
Cette forme quadratique est équivalente à la norme euclidienne de la projection orthogonale sur l'orthogonal de l'espace engendré par s_1 et s_2.

La figure 1.4 illustre cette projection. L'image d'origine est une image multispectrale d'une côte. L'océan est représenté en noir et les bancs de sable sont en jaune. Les deux signatures prises sur l'image d'origine 1.4a sont celle du banc de sable et celle de l'océan. La figure 1.4b montre le résultat lorsque nous choisissons une métrique euclidienne. Tous les contours apparaissent. En prenant la métrique de projection d'après les deux signatures, nous avons une détection de contours qui ne distingue plus l'océan des bancs de sable comme le montre la figure 1.4c.

Cet algorithme généralise l'algorithme original de Di Zenzo dans le sens où il permet de changer la métrique en chaque point de l'image alors que celui de Di Zenzo était limité à une métrique constante et euclidienne. Dans tout ce qui précède, nous avons fixé une métrique constante qui ne dépend pas du pixel considéré. Rien ne nous empêcherait de fixer une métrique précise pour chaque pixel. Dès lors, deux problèmes se posent. Comment fixer cette métrique qui dépend du pixel considéré ? Cela suppose un calcul préliminaire pour chaque pixel. Le deuxième problème soulevé est la quantité de données en jeu. La taille des images considérées est déjà un problème en soi, auquel il faut rajouter une métrique pour chaque pixel, c'est à dire la donnée d'une matrice $n \times n$ où n est le nombre de bandes. La dernière remarque que l'on peut faire est que même si la modélisation proposée généralise



(a) Image originale en fausses couleurs



(b) Détection par la métrique euclidienne



(c) Détection par projection

Figure 1.4 – Projection sur un sous-espace dans une image multispectrale

élégamment les approches marginales ou par luminance, l'algorithme résultant consiste en deux étapes. La première est une projection de chaque pixel sur un sous-espace fixé suivie de l'algorithme de Di Zenzo classique.

Ce premier développement nous a permis d'introduire les objets étudiés ainsi que la problématique de détection d'éléments particuliers dans une image multispectrale en fonction d'une signature. Nous présentons maintenant notre deuxième approche dans la détection de points d'intérêt dans une image multispectrale avec une acquisition compressée.

1.2 Acquisition compressée

L'acquisition compressée [1] est une discipline récente de la théorie du signal et de l'information. Elle est apparue à la suite des articles fondateurs de Candès, Tao et Donoho [CRT06, Don06]. L'idée principale derrière l'acquisition compressée est que l'échantillonnage et la compression sont réalisés en même temps. Selon le paradigme classique de l'acquisition de données représenté à la figure 1.5, la première étape est généralement un échantillonnage à un taux plus grand ou égal au taux de Nyquist. Cependant, la plupart du temps les données recueillies sont redondantes ou ont une structure particulière. Ces données brutes sont donc l'objet d'une étape de compression à l'issue de laquelle les données peuvent être transmises ou stockées. Lors d'une utilisation ultérieure, ces données sont décompressées et nous pouvons ensuite les utiliser.

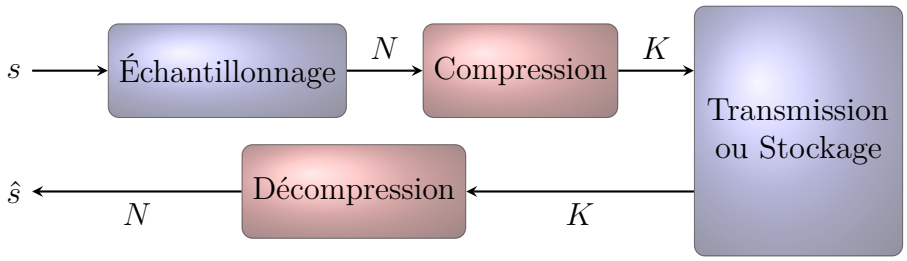


Figure 1.5 – Paradigme classique

Prenons l'exemple d'une photo d'un million de pixels prise par un appareil à photo. Dans l'étape d'échantillonnage, le capteur va acquérir un million de données. Avant d'être stockée, l'image ainsi acquise est compressée. Nous tirons parti du fait que l'image se représente bien dans une base d'ondelettes par exemple. En effet, la plupart des coefficients d'ondelettes sont ignorés car seulement une petite partie d'entre eux sont significatifs. Nous stockons donc non pas l'image entière mais une petite partie des coefficients d'ondelettes. Ce processus en deux temps peut paraître contradictoire : nous générons un nombre important de données avant d'en supprimer la plupart lors d'une opération coûteuse de compression. De ce constat est née l'acquisition compressée.

1.2.1 Principes

L'acquisition compressée propose de réaliser les deux opérations d'acquisition et de compression d'un seul coup. Elle consiste en une prise de mesures en nombre limité du signal à acquérir. Le signal qui nous intéresse est le plus souvent un

1. *compressed sensing* ou *compressive sensing* en anglais.

vecteur d'un espace vectoriel de grande dimension. Les mesures sont alors des produits scalaires du vecteur qui nous intéresse avec des vecteurs de mesure. Le nombre de ces mesures est très inférieur à la dimension du vecteur. Nous avons alors un nombre infini de vecteurs pouvant correspondre aux mesures prises. Nous avons cependant une information supplémentaire sur ce vecteur : comme précisé précédemment, il possède une base de représentation parcimonieuse et nous pouvons espérer qu'il y a un nombre réduit de vecteurs qui vérifient les mesures tout en ayant une représentation parcimonieuse dans une certaine base. C'est d'ailleurs le problème qu'il faut résoudre pour reconstruire le signal à partir des mesures prises. Nous recherchons le vecteur le plus parcimonieux possible vérifiant les mesures.

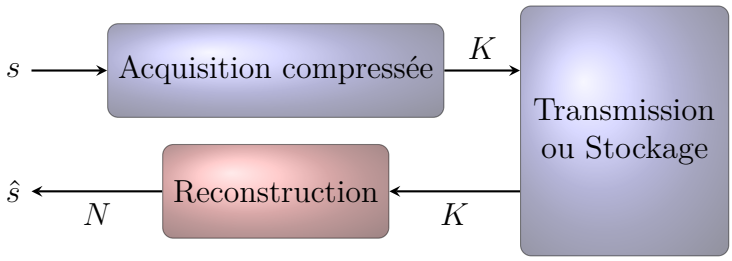


Figure 1.6 – Paradigme de l'acquisition compressée

Cette opération de reconstruction est un problème difficile. Il faut trouver des algorithmes efficaces calculant la solution.

1.2.2 Mesures et reconstruction

Nous allons d'abord préciser ce que nous entendons par mesures sur un signal. Si le signal à mesurer est représenté par un vecteur s, une mesure sur s est la donnée d'un produit scalaire de ce vecteur avec un vecteur de mesure prédéfini f. Lorsque nous faisons plusieurs mesures, nous pouvons regrouper tous ces produits scalaires sous la forme d'un produit matriciel. Nous avons alors une matrice de mesures F et le résultat de toutes ces mesures dans un vecteur m,

$$m = Fs.$$

Le vecteur m est tout ce dont nous disposons pour reconstruire le signal s. Comme m est de taille très inférieure à s, il existe une infinité de signaux s qui vérifient $m = Fs$. Cependant, nous pouvons faire des hypothèses sur le signal s. Par exemple, nous pouvons disposer d'une représentation parcimonieuse de s dans une base c'est à dire d'une matrice carrée Ψ telle que,

$$s = \Psi x \quad \text{avec } x \text{ parcimonieux.}$$

Nous pouvons alors formuler le problème de reconstruction sous la forme d'une minimisation faisant intervenir la norme L_0 qui compte le nombre d'entrées non nulle.

$$\arg\min_x \|x\|_0 \quad \text{tel que} \quad m = F\Psi x. \tag{1.5}$$

Par exemple, la matrice Ψ de la figure 1.7 représente une matrice DCT [2]. Le

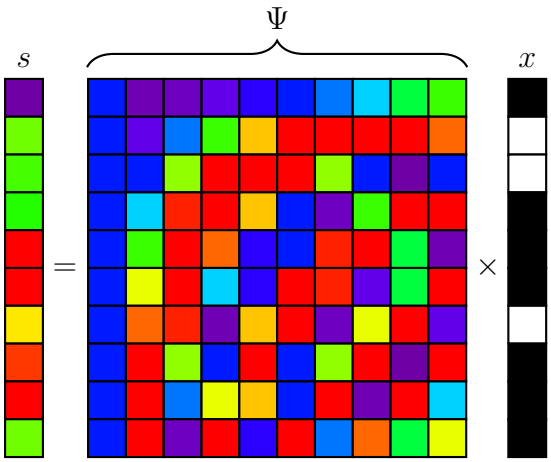


Figure 1.7 – Représentation parcimonieuse de s par une matrice DCT

vecteur s s'écrit comme la transformée d'un vecteur 3-parcimonieux. Le vecteur s est alors 3-parcimonieux dans la base de représentation Ψ.

Nous pouvons également supposer que le signal s est la superposition de signaux appartenant à un dictionnaire Ψ connu. Le signal s s'écrit alors,

$$s = \Psi x \quad \text{avec } x \text{ parcimonieux.}$$

Cette écriture est une généralisation du cas précédent car le dictionnaire peut contenir un nombre quelconque d'atomes : la matrice Ψ peut avoir beaucoup plus de colonnes que de lignes. La problème de minimisation est le même. La figure 1.8 représente la décomposition d'un signal s dans un dictionnaire qui est la réunion de deux bases.

Nous pouvons enfin supposer que le signal s possède une transformée parcimonieuse c'est à dire qu'il existe une matrice Ψ telle que,

$$\Psi s \text{ est parcimonieux.}$$

2. Transformée en cosinus discrète ou *Discrete cosine transform* en anglais.

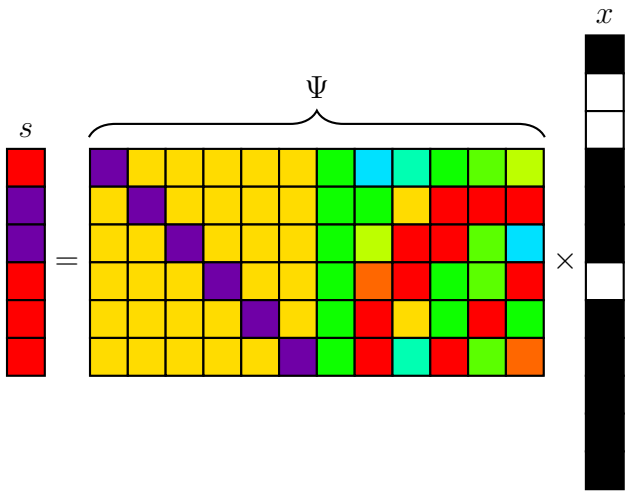


Figure 1.8 – Représentation dans un dictionnaire DCT/Dirac

Le problème s'écrit alors,

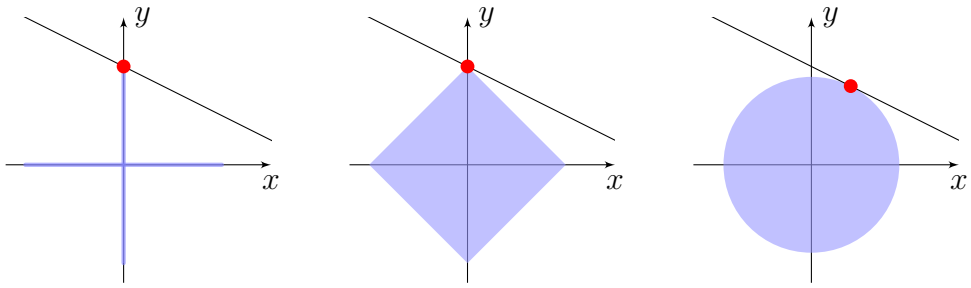
$$\arg\min_{s} \|\Psi s\|_0 \quad \text{tel que} \quad m = Fs. \tag{1.6}$$

On constate que tous ces problèmes de reconstruction reposent sur la norme L_0. La résolution de tels problèmes est difficile du fait de la nature combinatoire de la norme L_0. Plusieurs algorithmes ont été proposés pour résoudre ce type de minimisation. Nous proposons de les introduire brièvement avant de justifier le choix de l'un d'entre eux.

1.2.3 Les algorithmes de reconstruction

De multiples algorithmes ont été suggérés pour résoudre les minimisations (1.5) et (1.6) et leurs dérivées. Nous avons d'une part les algorithmes gloutons qui sont des algorithmes itératifs qui à chaque itération trouvent une solution locale optimale. Parmi ces algorithmes, le *matching pursuit (MP)* [MZ93] proposé par Mallat et Zhang en 1993 consiste à sélectionner à chaque itération l'élément du dictionnaire le plus en corrélation avec le signal puis de réitérer ce procédé avec le vecteur résiduel. Une amélioration consiste à modifier rétroactivement la contribution des éléments sélectionnés dans le dictionnaire : c'est le *orthogonal matching pursuit (OMP)* [PRK93], voir [TAG07, BD08, DMA97, DET04] pour plus de détails. Des algorithmes reposant sur la même idée sont ensuite proposés : le *Stagewise OMP (StOMP)* [DTDS12], le *Regularized OMP (ROMP)* [NV09] et le *CoSaMP* [NT08].

D'autre part, nous avons les algorithmes convexes qui consistent à trouver des minimisations convexes qui approchent les problèmes (1.5) et (1.6). En effet, une approximation connue depuis assez longtemps de la norme L_0 est la norme L_1. La figure 1.9 illustre ce fait. Dans les trois cas, nous avons un même ensemble de solutions représenté par une droite et la solution de la minimisation représenté par un point rouge. La figure 1.9a montre la minimisation de la norme L_0. La minimisation L_1 à la figure 1.9b reconstruit bien la même solution alors que la minimisation L_2 à la figure 1.9c donne une solution grossière.



(a) Solution L_0 (b) Solution L_1 (c) Solution L_2

Figure 1.9 – Reconstruction selon différentes normes

L'approche la plus simple consiste donc à remplacer la norme L_0 par une norme L_1. Les problèmes (1.5) et (1.6) deviennent alors respectivement,

$$\arg\min_x \|x\|_1 \quad \text{tel que} \quad m = F\Psi x, \tag{1.7}$$

et,

$$\arg\min_s \|\Psi s\|_1 \quad \text{tel que} \quad m = Fs. \tag{1.8}$$

Le problème (1.7), connu sous le nom de *Basis Pursuit (BP)* est introduit par Chen, Donoho et Saunders dans [CDS98]. L'intérêt de la norme L_1 réside dans le fait que, contrairement à la norme L_0, elle est convexe. Le problème (1.7) se réduit alors à un simple problème de programmation linéaire. De nombreuses minimisations sont alors envisagées. L'introduction de bruit dans les mesures suggère de considérer plutôt les problèmes,

$$\arg\min_x \|x\|_1 \quad \text{tel que} \quad \|m - F\Psi x\|_2 \leqslant \sigma, \tag{1.9}$$

et,

$$\arg\min_x \|x\|_1 + \frac{\lambda}{2} \|m - F\Psi x\|_2^2 . \tag{1.10}$$

De même, nous dérivons de la minimisation (1.8) les problèmes,

$$\arg\min_s \|\Psi s\|_1 \quad \text{tel que} \quad \|m - Fs\|_2 \leqslant \sigma, \tag{1.11}$$

et,

$$\arg\min_{s} \|\Psi s\|_1 + \frac{\lambda}{2} \|m - Fs\|_2 . \tag{1.12}$$

Ces minimisations ont donné lieu au développement de différents algorithmes. Les problèmes de minimisations de la variation totale d'une image proposés dans [ROF92] pour le débruitage d'images sont des problèmes de la forme (1.10) ou (1.12) et sont résolus dans [ROF92] en les traduisant en des équations aux dérivées partielles. Les problèmes (1.7) et (1.10) peuvent se traduire en programme linéaire. L'algorithme du simplexe ou la méthode du point intérieur sont alors mis à profit pour les résoudre dans [CDS98]. Les autres problèmes nécessitent des outils plus compliqués comme l'optimisation sur le cône du second ordre [3] utilisée dans la boîte à outils ℓ_1`-magic` réalisée par Candès et Romberg [CR05].

Dans les chapitres qui suivent, nous serons confrontés à des problèmes du type (1.11) avec une contrainte supplémentaire de positivité pour s et des fonctions Ψ qui iront de la simple identité qui nous ramène à la norme L_1 ou à un calcul de variation totale et même un mélange des deux. Les problèmes que nous rencontrerons seront également de grandes dimensions. Pour toutes ces raisons, nous présentons dans le chapitre suivant les algorithmes de type Bregman qui sont faciles à implanter et assez flexibles pour résoudre la plupart des problèmes de minimisation auxquels nous serons confrontés.

3. *Second-order cone programming, (SOCP)* en anglais.

Chapitre 2

Algorithmes de type Bregman

Sommaire

2.1 L'itération de Bregman 20
2.1.1 Sous-différentielle 21
2.1.2 Distance de Bregman 21
2.1.3 L'itération de Bregman 22
2.1.4 Cas où $H(u) = \|Au - f\|_2^2$ 24
2.2 Algorithme de Bregman scindé 26
2.2.1 Opérations de seuillage 26
2.2.2 Technique de Bregman scindé 27
2.3 Bregman scindé contraint 29
2.4 Applications du Bregman scindé 31
2.4.1 Application au débruitage TV/L_2 anisotrope 31
2.4.2 Application au débruitage TV/L_2 isotrope 34
2.4.3 Application à la reconstruction TV d'une image 41
2.5 Conclusion 45

Le chapitre précédent aboutit sur un problème de minimisation qui va être au cœur du travail réalisé dans cette thèse. Dans les chapitres suivants, plusieurs variantes de ce problème seront tour à tour envisagées suivant la nature et les contraintes du problème. En effet, nous serons amenés à changer la fonctionnelle pour améliorer les résultats, à ajouter une contrainte de positivité sur la solution ou à intégrer la contrainte dans la fonctionnelle. Une autre contrainte sera l'efficacité de l'algorithme qui devra fonctionner en un temps raisonnable avec des problèmes de plusieurs milliers de variables. Dans ce chapitre, nous allons présenter une classe d'algorithmes appelés algorithmes de Bregman répondant à ces différentes contraintes. Les algorithmes de type Bregman présentent plusieurs avantages. Premièrement, la simplicité de leur mise en œuvre. Ensuite, la flexibilité qu'ils

apportent. Et enfin, l'efficacité dans la résolution de problèmes avec un nombre important de variables.

Le point de départ est l'introduction de la distance de Bregman dans [Bre67]. L'auteur l'utilise pour généraliser une projection sur un convexe fermé et montre qu'on peut résoudre un problème d'optimisation convexe à contrainte linéaire en projetant successivement sur des convexes fermés à l'aide de cette distance. Cette idée est reprise et développée dans [CL81] et dans [Kiw97] où la notion de distance de Bregman généralisée est introduite. Elle permet de définir une distance de Bregman même lorsque la fonction utilisée n'est pas différentiable. Des algorithmes inspirés de Bregman et appliqués au traitement du signal apparaissent dans [Çet91, Çet89]. Les algorithmes de Bregman sont ensuite appliqués avec succès au difficile problème de la minimisation TV/L_2 issu du modèle de Rudin, Osher et Fatemi (ROF) dans [OBG+06]. Des applications au débruitage et au défloutage d'images sont aussi proposées. Un second papier [YOGD08] applique ces mêmes algorithmes à l'acquisition compressée avec le problème de poursuite de la base [1]. L'itération de Bregman consiste en une suite de problèmes sans contrainte qui sont pris en charge par d'autres algorithmes. Enfin, l'introduction de la technique de Bregman scindé [2] dans [GO09] tire parti des algorithmes de Bregman développés jusqu'alors et d'une technique de scindement introduite dans [WY07]. Elle accroît encore l'intérêt de l'itération de Bregman. En effet, non seulement le recours à d'autres algorithmes de minimisation n'est plus nécessaire grâce à cette opération de scindement mais le Bregman scindé permet de résoudre des minimisations aux fonctionnelles beaucoup plus compliquées qui peuvent alors mélanger des normes TV, L_1 ou de Besov. Des applications à l'acquisition compressée sur des problèmes plus compliqués que celui de la poursuite de la base sont mis en avant dans [GO09]. La convergence des méthodes de Bregman y est également démontrée. Celle des algorithmes de type Bregman scindé est démontrée dans [COS09].

Dans ce chapitre, nous allons nous intéresser à des problèmes de minimisation qui englobent tous ceux vus au chapitre précédent. Ils sont de la forme suivante,

$$\arg\min_u J(u) \quad \text{tel que} \quad H(u) = 0, \tag{2.1}$$

ou sous la forme non contrainte correspondante,

$$\arg\min_u J(u) + \lambda H(u). \tag{2.2}$$

Les fonctions J et H sont positives et convexes et H est supposée différentiable.

1. *Basis pursuit* en anglais.
2. *Split Bregman* en anglais.

2.1 L'itération de Bregman

Avant de voir l'algorithme de Bregman en détail, nous commençons par introduire quelques notions nécessaires.

2.1.1 Sous-différentielle

La sous-différentielle est une généralisation de la notion classique de différentielle lorsque nous considérons une fonction convexe. En effet, lorsqu'une fonction convexe n'est plus différentiable et ce sera le cas assez souvent notamment dû à l'usage de la norme L_1, on voudrait quand même avoir un gradient « acceptable » pour cette fonction.

Plus précisément, soit J une fonction scalaire convexe et positive de $\mathbb{R}^n$ dans $\mathbb{R}$. Nous appellons sous-différentielle ou sous-gradient de J en $u \in \mathbb{R}^n$ et nous noterons $\partial J(u)$ l'ensemble :

$$\{p \in \mathbb{R}^n, \forall v \in \mathbb{R}^n, J(v) \geqslant J(u) + \langle p, v - u \rangle\} \tag{2.3}$$

Si la fonction J est différentiable en u, la sous-différentielle $\partial J(u)$ est réduite au singleton formé par la différentielle. Si u est un minimum local de la fonction J, alors $0 \in \partial J(u)$.

2.1.2 Distance de Bregman

La distance de Bregman a été introduite par Bregman dans [Bre67]. Ce n'est pas une distance au sens propre du terme puisque qu'elle n'est pas symétrique et ne satisfait pas l'inégalité triangulaire. Néanmoins, elle généralise certaines distances telles que la distance Euclidienne ou la distance de Mahalanobis.

Définition 2.1. Soit J une fonction scalaire convexe de $\mathbb{R}^n$ dans $\mathbb{R}$, u, v des éléments de $\mathbb{R}^n$ et $p \in \partial J(v)$ une sous-différentielle de J en v. La distance de Bregman de u à v selon J est définie par la quantité,

$$D_J^p(u, v) = J(u) - J(v) - \langle p, u - v \rangle. \tag{2.4}$$

Il s'agit en fait de l'erreur commise entre $J(u)$ et son approximation de Taylor à l'ordre 1 depuis v comme l'illustre la figure 2.1.

Même si la « distance de Bregman » n'est pas une distance, elle vérifie plusieurs propriétés intéressantes. Comme J est convexe, nous avons toujours $D_J^p \geqslant 0$. De plus, pour tout élément w appartenant au segment $[u; v]$, nous avons $D_J^p(u, v) \geqslant D_J^p(w, v)$.

Exemple 2.1 ([BMDG05])**.** Quelques exemples de distances usuelles qui sont en fait des distances de Bregman.

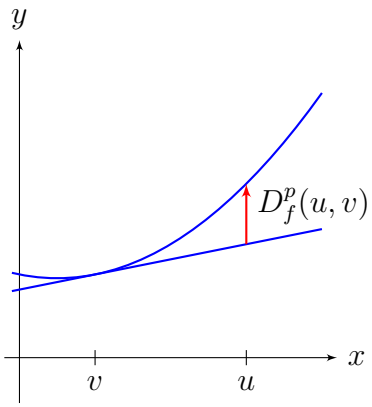


Figure 2.1 – Distance de Bregman

– La distance euclidienne est une distance de Bregman pour la fonction J suivante,

$$J(u) = \|u\|_2^2.$$

En effet,

$$D_J^p(u,v) = \|u\|_2^2 - \|v\|_2^2 - \langle p, u-v\rangle.$$

Or $\partial J(v)$ est réduit au singleton $\{2v\}$ donc $p = 2v$ et,

$$\begin{aligned} D_J^p(u,v) &= \|u\|_2^2 - \|v\|_2^2 - \langle u-v, 2v\rangle \\ &= \|u-v\|_2^2. \end{aligned}$$

– La distance de Mahalanobis est obtenue pour la fonction J suivante,

$$J(u) = u^T A u,$$

nous avons alors,

$$D_J(u,v) = (u-v)^T A (u-v).$$

– La distance de Kullback-Leibler entre deux distributions de probabilités discrètes p et q est obtenue au moyen de la fonction convexe,

$$J(p) = \sum_i p_i \log_2 p_i,$$

nous avons alors,

$$D_J(p,q) = KL(p\|q).$$

2.1.3 L'itération de Bregman

Nous allons à présent nous intéresser au problème (2.1) que nous rappelons ici.

$$\arg\min_u J(u) \quad \text{tel que} \quad H(u) = 0. \tag{2.5}$$

La technique de Bregman, comme suggéré dans [OBG+06] consiste à intégrer la contrainte à la minimisation et remplacer la fonction J par une distance de Bregman. Nous itérons ensuite la minimisation obtenue ce qui nous donne la suite de problèmes,

$$u^{k+1} = \arg\min_u D_J^{p^k}(u, u^k) + \lambda H(u), \tag{2.6}$$

avec $p^k \in \partial J(u^k)$.

Les p_k peuvent être calculés itérativement. En effet, si nous posons $\tilde{J}^k(u) = D_J^{p^k}(u, u^k) + \lambda H(u)$, comme u^{k+1} est un minimum de la fonction $\tilde{J}^k$, 0 est un sous-gradient de $\tilde{J}^k$ et donc,

$$0 \in \partial \tilde{J}^k(u^{k+1}) = \partial J(u^{k+1}) - p^k + \lambda \nabla H(u^{k+1}). \tag{2.7}$$

Pour l'itération suivante, il nous faut $p^{k+1} \in \partial J(u^{k+1})$. Nous choisissons donc d'après (2.7),

$$\begin{aligned} p^0 &\in \partial J(u^0), \\ p^{k+1} &= p^k - \lambda \nabla H(u^{k+1}). \end{aligned}$$

L'algorithme prend donc la forme suivante,

$$\begin{cases} u^{k+1} = \arg\min_u D_J^{p^k}(u, u^k) + \lambda H(u) & \text{(2.8a)} \\ p^{k+1} = p^k - \lambda \nabla H(u^{k+1}) & \text{(2.8b)} \\ p^0 \in \partial J(u^0) & \text{(2.8c)} \end{cases}$$

Le théorème suivant justifie l'introduction de l'itération de Bregman (2.6) pour résoudre le problème (2.1).

Théorème 2.1 ([OBG+06])**.** *En se plaçant dans les conditions rappelées en début de chapitre c'est à dire J positive et convexe et H positive, convexe et différentiable. Nous avons les propriétés de convergence suivantes,*

- *La suite $(H(u^k))_k$ est décroissante : $H(u^{k+1}) \leqslant H(u^k)$.*
- *Elle converge vers le minimum $H(u^*)$ de H et :*

$$H(u^k) \leqslant H(u^*) + \frac{J(u^*)}{k}$$

– *La suite* $D_J^{p^k}(\bar{u}, u^k)$ *tend vers zéro.*

À mesure que nous itérons l'algorithme (2.8), u^k converge vers la vraie solution en terme de distance de Bregman et la contrainte est réalisée puisque $H(u^k)$ tend vers son minimum.

Néanmoins, la minimisation (2.8a) reste difficile à résoudre car même si la contrainte a disparu, la fonctionnelle mêle les fonctions J et H. Nous allons voir qu'en faisant des suppositions sur H, l'algorithme (2.8) va devenir beaucoup plus simple.

2.1.4 Cas où $H(u) = \|Au - f\|_2^2$

Nous supposons à présent que $H(u) = \|Au - f\|_2^2$, ce qui est le cas dans les problèmes rencontrés au chapitre précédent. La minimisation (2.1) devient alors,

$$\arg\min_u J(u) \quad \text{tel que} \quad Au = f. \tag{2.9}$$

Le terme $\nabla H(u^{k+1})$ apparaissant dans (2.8b) s'écrit alors,

$$\nabla H(u^{k+1}) = A^T(Au^{k+1} - f).$$

L'itération se réécrit alors, [3]

$$\begin{cases} u^{k+1} = \arg\min_u D_J^{p^k}(u, u^k) + \dfrac{\lambda}{2}\|Au - f\|_2^2 & \text{(2.10a)} \\ p^{k+1} = p^k - \lambda A^T(Au^{k+1} - f) & \text{(2.10b)} \\ p^0 \in \partial J(u^0) & \text{(2.10c)} \end{cases}$$

Du fait notamment que la contrainte H est la norme L_2 d'une fonction affine en u, des simplifications s'opèrent. L'algorithme (2.10) se réécrit de la manière beaucoup plus simple suivante,

$$\begin{cases} u^{k+1} = \arg\min_u J(u) + \dfrac{\lambda}{2}\|Au - f^k\|_2^2 & \text{(2.11a)} \\ f^{k+1} = f^k + f - Au^{k+1} & \text{(2.11b)} \end{cases}$$

Cette équivalence est montrée dans [OBG+06], nous la reformulons ici dans la proposition suivante.

Proposition 2.2. *Les algorithmes (2.10) et (2.11) sont équivalents si et seulement si* $p^0 = \lambda A^T(f^0 - f)$.

3. Le coefficient λ est remplacé par $\frac{\lambda}{2}$ pour plus de commodité lorsqu'il s'agira de différencier.

Démonstration. Il suffit de montrer que les deux fonctions à minimiser diffèrent d'une constante. Nous formons donc la différence,

$$\begin{aligned} D_J^{p^k}(u,u^k) + \frac{\lambda}{2}\|Au-f\|_2^2 - J(u) - \frac{\lambda}{2}\|Au-f^k\|_2^2 \\ = J(u) - J(u^k) - \langle p^k, u-u^k\rangle \\ + \frac{\lambda}{2}\left(\|Au\|_2^2 - 2\langle Au, f\rangle + \|f\|_2^2\right) \\ - J(u) \\ - \frac{\lambda}{2}\left(\|Au\|_2^2 - 2\langle Au, f^k\rangle + \|f^k\|_2^2\right), \end{aligned}$$

qui se réduit en enlevant les termes ne dépendant pas de u à

$$-\langle p^k, u\rangle + \lambda\langle Au, f^k - f\rangle.$$

Ainsi, les deux fonctions à minimiser diffèrent d'une constante si et seulement si $\lambda\langle Au, f^k - f\rangle = \langle p^k, u\rangle$ pour tout u c'est à dire $p^k = \lambda A^T(f^k - f)$. Il faut et il suffit donc que cette égalité soit vraie pour tout k. Nous allons montrer que si elle est vraie pour $k = 0$ alors elle est vraie pour tout k. Par récurrence, nous supposons que l'égalité est vraie au rang k, nous calculons alors,

$$\begin{aligned} p^{k+1} - \lambda A^T(f^{k+1} - f) &= p^k - \lambda A^T(Au^{k+1} - f) - \lambda A^T(f^k - Au^{k+1}) \\ &= p^k - \lambda A^T(f^k - f) \\ &= 0, \end{aligned}$$

où la première égalité est obtenue en injectant les égalités (2.10b) et (2.11b). D'où la propriété au rang $k+1$. ■

Remarque. Dans le cas où la fonction J atteint son minimum en 0, alors pour satisfaire l'égalité $p^0 = \lambda A^T(f^0 - f)$ et la condition $p^0 \in \partial J(u^0)$, il suffit de prendre $u^0 = 0$, $p^0 = 0$ et $f^0 = f$. Ce sera le cas par exemple lorsque nous prendrons $J(u) = \|u\|_1$, $J(u) = \|u\|_{\mathrm{TV}}$ ou plus généralement $J(u) = \|\phi u\|_1$ avec ϕ une fonction linéaire.

Un autre théorème de convergence figure dans [GO09] et vient compléter celui déjà établi dans le cas général.

Théorème 2.3 ([GO09])**.** *Supposons que A est linéaire et qu'il existe un itéré u^* tel que $Au^* = f$. Alors u^* est une solution du problème avec contraintes (2.9).*

Démonstration. Soit $\hat{u}$ une solution du problème (2.9). Nous avons alors $A\hat{u} = f$. L'itéré u^* est solution de (2.11). Nous avons donc,

$$J(\hat{u}) + \frac{\lambda}{2}\|A\hat{u} - f^*\|_2^2 \geqslant J(u^*) + \frac{\lambda}{2}\|Au^* - f^*\|_2^2,$$

d'où nous en déduisons, étant donné que $Au^* = f$ et $A\hat{u} = f$,

$$J(\hat{u}) \geqslant J(u^*).$$

Ainsi, u^* est une solution du problème de départ (2.9). ■

2.2 Algorithme de Bregman scindé

Dans cette section, nous nous intéressons à une technique dite de Bregman scindé qui combine une technique de scindement vue dans [WY07] et une utilisation des outils de type Bregman décrit à la section précédente.

La technique de scindement propose d'introduire d'une part une nouvelle inconnue afin de découpler le système et d'autre part une contrainte supplémentaire pour fixer le comportement de cette nouvelle inconnue. Le problème envisagé se décompose en deux minimisations plus simples dont l'une est un problème des moindres carrés et l'autre peut être résolue efficacement avec un simple seuillage.

Nous retrouvons cette idée de scindement dans [GO09], couplée avec les outils de type Bregman. En fait, nous avons toujours introduction d'une nouvelle inconnue mais la contrainte n'est pas simplement rajoutée comme un terme de pénalité dépendant d'un paramètre qui va spécifier le poids de cette contrainte dans la fonction à minimiser. Le formalisme de Bregman tel que présenté à la section précédente va prendre en charge cette contrainte. [GO09] propose une application à l'acquisition compressée. Avant d'introduire les algorithmes de Bregman scindé, nous introduisons quelques opérateurs qui vont nous permettre d'exprimer les solutions de sous-problèmes.

2.2.1 Opérations de seuillage

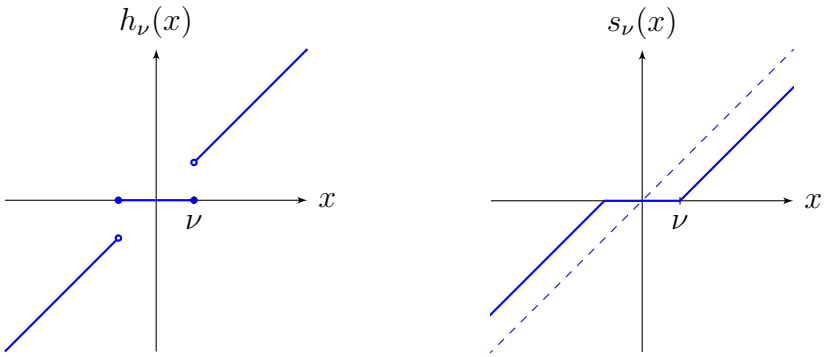
Il existe deux opérations classiques de seuillage : le seuillage fort [4] et le seuillage doux [5]. Le seuillage doux nous intéresse particulièrement ici car il nous permet d'exprimer directement la solution d'un problème de minimisation que nous allons rencontrer.

Seuillage fort

Le seuillage fort de paramètre ν est tel que toutes les valeurs $|x| \leqslant \nu$ passées en arguments sont mises à zéro. Ainsi,

$$h_\nu(x) = \begin{cases} 0 & \text{si } |x| \leqslant \nu \\ x & \text{sinon.} \end{cases}$$

4. *Hard thresholding* en anglais.
5. *Soft thresholding* ou *shrinkage operation* en anglais.



(a) Seuillage fort (b) Seuillage doux

Figure 2.2 – Seuillages fort et doux

Seuillage doux

Le seuillage doux est un opérateur qui apparaît dans [Don95] dans le cadre de débruitage à base d'ondelettes.

Le seuillage doux s_ν de paramètre ν est un seuillage en valeur absolue qui à la différence du seuillage fort a l'avantage d'être continu.

$$s_\nu(x) = \operatorname{sgn}(x)\max(0, |x| - \nu).$$

Propriétés et notations

Le seuillage doux vérifie une propriété qui nous sera utile par la suite.

Proposition 2.4. *Soit $x, \lambda, \nu \in \mathbb{R}$ alors nous avons,*
- $s_{\lambda\nu}(x) = \lambda s_\nu(\frac{x}{\lambda})$

Par ailleurs, nous allons être amenés à seuiller des vecteurs coordonnées par coordonnées. Nous adoptons les notations suivantes.

Notation 2.5. Si $x, \lambda \in \mathbb{R}$ et $y, \Lambda \in \mathbb{R}^n$ les notations $s_\lambda(y)$ et $s_\Lambda(y)$ auront le sens suivant.
- $s_\lambda(y)$ applique à chaque coordonnée de y un seuillage de paramètre λ c'est à dire,
$$(s_\lambda(y))_i = s_\lambda(y_i) \quad \text{pour tout} \quad i \in 1, \ldots, n.$$
- $s_\Lambda(y)$ applique à chaque coordonnée de y un seuillage de paramètre la coordonnée correspondante dans Λ c'est à dire,
$$(s_\Lambda(y))_i = s_{\Lambda_i}(y_i) \quad \text{pour tout} \quad i \in 1, \ldots, n.$$

2.2.2 Technique de Bregman scindé

La technique de Bregman scindé est décrite dans [GO09, WY07]. Elle consiste à découpler une expression par l'introduction d'une nouvelle variable. Cette variable est ensuite contrainte pour retrouver le problème initial. L'intérêt est que la nouvelle expression à optimiser comporte des termes indépendants. La contrepartie est l'introduction d'une contrainte dans un problème qui n'en comportait pas. La technique de Bregman permet justement de s'affranchir de cette nouvelle contrainte.

Les minimisations qui se prêtent à cette technique sont d'une forme assez générale. En reprenant les notations qui figurent dans [GO09],

$$\arg\min_u \|\phi(u)\|_1 + G(u), \tag{2.12}$$

où ϕ est un opérateur linéaire et G un opérateur différentiable. Nous pouvons choisir librement la fonction ϕ et la fonction G. Suivant les cas, nous pourrons faire du débruitage d'images à l'aide de la norme de variation totale ou réaliser un algorithme de type poursuite de la base utilisée en acquisition compressée.

En suivant [WY07], découplons cette minimisation en introduisant la variable d,

$$\arg\min_{u,d} \|d\|_1 + G(u) \quad \text{tel que} \quad d = \phi(u). \tag{2.13}$$

Nous avons rajouté une contrainte et une variable à minimiser mais l'expression à optimiser est plus simple puisque les deux termes sont devenus indépendants.

En suivant [GO09], nous pouvons ensuite appliquer la technique de Bregman et transformer cette minimisation contrainte en une série de minimisation sans contrainte. En effet, (2.13) est de la forme de (2.9) si nous posons,

$$\begin{aligned} J(u,d) &= \|d\|_1 + G(u), \\ H(u,d) &= \|d - \phi(u)\|_2^2. \end{aligned}$$

En suivant (2.10) nous trouvons,

$$\begin{cases} (u^{k+1}, d^{k+1}) = \arg\min_{(u,d)} D_J^{(p^k,q^k)}(u,d,u^k,d^k) + \dfrac{\lambda}{2}\|d - \phi(u)\|_2^2 & \text{(2.14a)} \\ \begin{pmatrix} p^{k+1} \\ q^{k+1} \end{pmatrix} = \begin{pmatrix} p^k \\ q^k \end{pmatrix} + \dfrac{\lambda}{2} B^T B \begin{pmatrix} u^{k+1} \\ d^{k+1} \end{pmatrix} \quad \text{avec} \quad B = \begin{pmatrix} -\phi & I \end{pmatrix} & \text{(2.14b)} \\ (p^0, q^0) \in \partial J(u^0, d^0) & \text{(2.14c)} \end{cases}$$

Puis,

$$\begin{cases} (u^{k+1}, d^{k+1}) = \arg\min_{(u,d)} J(u,d) + \dfrac{\lambda}{2}\|d - \phi(u) - f^k\|_2^2 & \text{(2.15a)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.15b)} \\ f^0 = 0 & \text{(2.15c)} \end{cases}$$

L'équation (2.15a) est une minimisation par rapport à deux variables u et d. Nous pouvons théoriquement la réaliser en faisant successivement des minimisations par rapport à u puis par rapport à d. Pour simplifier, nous procédons à une unique itération. L'itération (2.15) se décompose désormais en trois équations.

$$\begin{cases} u^{k+1} = \arg\min_u G(u) + \dfrac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 & \text{(2.16a)} \\ d^{k+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 & \text{(2.16b)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.16c)} \end{cases}$$

La première équation est un calcul des moindres carrés lorsque $G(u) = \frac{\mu}{2}\|Au - f\|_2^2$ qui peut être résolu très efficacement. La seconde minimisation est comme nous allons le voir dans la prochaine section une application de la propriété 2.7 et se résume donc essentiellement à un seuillage doux.

Dans de nombreux problèmes, nous supposerons que $G(u) = \frac{\mu}{2}\|Au - f\|_2^2$. Nous avons alors la minimisation suivante.

$$\arg\min_u \|u\|_1 + \frac{\mu}{2}\|Au - f\|_2^2.$$

par l'*algorithme de Bregman scindé non contraint*, nous avons,

$$\begin{cases} u^{k+1} = \arg\min_u \dfrac{\mu}{2}\|Au - f\|_2^2 + \dfrac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 & \text{(2.17a)} \\ d^{k+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 & \text{(2.17b)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.17c)} \end{cases}$$

2.3 Bregman scindé contraint

La technique de Bregman a transformé une minimisation contrainte en une série de minimisations non contraintes. Or, la technique de Bregman scindé résout justement des minimisations non contraintes. Nous pouvons combiner les deux techniques pour résoudre le problème suivant,

$$\arg\min_u \|\phi(u)\|_1 \quad \text{tel que} \quad Au = b. \tag{2.18}$$

Nous commençons par transformer le problème précédent par une itération de Bregman. Nous avons alors, [6]

$$\begin{cases} u^{k+1} = \arg\min_u \|\phi(u)\|_1 + \dfrac{\mu}{2}\|Au - b^k\|_2^2 & \text{(2.19a)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.19b)} \end{cases}$$

6. Le paramètre λ qui figure dans 2.11a est remplacé par μ pour pouvoir appliquer la technique de Bregman scindé.

Il nous reste à résoudre à chaque itération la minimisation (2.19a). Or, il s'agit justement d'un problème que prend en charge le Bregman scindé par l'itération, [7]

$$\begin{cases} u^{l+1} = \arg\min_u \dfrac{\mu}{2}\|Au - b^k\|_2^2 + \dfrac{\lambda}{2}\|d^l - \phi(u) - f^l\|_2^2 & \text{(2.20a)} \\ d^{l+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{l+1}) - f^l\|_2^2 & \text{(2.20b)} \\ f^{l+1} = f^l + \phi(u^{l+1}) - d^{l+1} & \text{(2.20c)} \end{cases}$$

En combinant les deux, nous obtenons,

$$\begin{cases} \begin{cases} u^{l+1} = \arg\min_u \frac{\mu}{2}\|Au - b^k\|_2^2 + \frac{\lambda}{2}\|d^l - \phi(u) - f^l\|_2^2 \\ d^{l+1} = \arg\min_d \|d\|_1 + \frac{\lambda}{2}\|d - \phi(u^{l+1}) - f^l\|_2^2 \\ f^{l+1} = f^l + \phi(u^{l+1}) - d^{l+1} \end{cases} & \text{(2.21a)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.21b)} \end{cases}$$

Cet algorithme contient ainsi une itération imbriquée. En limitant la boucle intérieure à une seule itération, nous obtenons *l'algorithme de Bregman scindé contraint*,

$$\begin{cases} u^{k+1} = \arg\min_u \dfrac{\mu}{2}\|Au - b^k\|_2^2 + \dfrac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 & \text{(2.22a)} \\ d^{k+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 & \text{(2.22b)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.22c)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.22d)} \end{cases}$$

Cet algorithme fait plusieurs approximations. La minimisation portant sur les deux variables u et d est découplée et la boucle est limitée à une seule itération dans (2.20). De la même manière, une unique itération est réalisée dans (2.21). Pourtant, [COS09] montre que cet algorithme converge vers une solution du problème (2.18).

2.4 Applications du Bregman scindé

La méthode de Bregman scindé permet de résoudre des minimisations d'une forme assez générale. Dans l'expression (2.12), nous pouvons en effet choisir l'application ϕ qui doit être linéaire et la fonction G qui doit être différentiable. Dans les sous-sections suivantes, nous allons voir que différents choix pour ϕ et G mènent à la résolution :

7. L'indice d'itération est renommé car il est déjà utilisé par l'algorithme (2.19)

1. d'un problème classique de débruitage d'images de type TV/L_2, en commençant d'abord par une norme TV approchée puis par la vraie norme TV.
2. d'un problème hybride qui injecte la norme TV dans un problème classique de type poursuite de la base.

2.4.1 Application au débruitage TV/L_2 anisotrope

Dans cette sous-section, nous nous intéressons au problème classique de débruitage d'images selon le modèle de Rudin, Osher et Fatemi (ROF) introduit dans [ROF92]. La fonctionnelle utilisée est la norme de variation totale TV qui est la norme L_1 du gradient. Nous devons donc résoudre,

$$\arg\min_u TV(u) + \frac{\mu}{2}\|u - f\|_2^2.$$

Nous retrouvons cette minimisation dans (2.12) en choisissant ϕ de la manière suivante,

$$\begin{array}{rccc} \phi: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^{2n} \\ & u & \longmapsto & \begin{pmatrix} \nabla_x u \\ \nabla_y u \end{pmatrix}, \end{array}$$

et $G(u) = \frac{\mu}{2}\|u - f\|_2^2$.

Nous remarquons toutefois que nous avons $J(u) = \|\phi(u)\|_1 = \|\nabla_x u\|_1 + \|\nabla_y u\|_1$ qui calcule la somme en valeur absolue des gradients de notre image. Ce n'est pas vraiment la norme TV qui calcule normalement la norme L_2 du gradient alors que nous calculons ici la norme L_1. Nous parlerons alors de norme TV anisotrope [GO09]. L'utilisation de la vraie norme TV sera étudiée à la sous-section suivante.

En réécrivant l'algorithme de Bregman scindé non contraint (2.22) dans ce cas particulier, nous obtenons les équations suivantes,

$$\begin{cases} u^{k+1} = \arg\min_u \dfrac{\mu}{2}\|u - f\|_2^2 + \dfrac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 & \text{(2.23a)} \\ d^{k+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 & \text{(2.23b)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.23c)} \end{cases}$$

Il nous reste donc à résoudre les deux minimisations (2.23a) et (2.23b).

Résolution de (2.23a)

La minimisation,

$$\arg\min_u \frac{\mu}{2}\|u - f\|_2^2 + \frac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2,$$

porte sur une fonctionnelle différentiable. Une condition nécessaire est donc que le gradient soit nul,

$$\mu\langle u - f, \cdot\rangle - \lambda\langle d^k - \phi(u) - f^k, \phi(\cdot)\rangle = 0,$$

que l'on peut reformuler en identifiant $\mathbb{R}^n$ et son dual par le produit scalaire usuel. Nous obtenons ainsi,

$$\mu(u - f) = \lambda\phi^T(d^k - \phi u - f^k),$$

puis,

$$(\mu I + \lambda\phi^T\phi)u = \mu f + \lambda\phi^T(d^k - f^k). \tag{2.24}$$

En posant $D = \mu I + \lambda\phi^T\phi$ et en supposant que D est inversible, alors nous avons,

$$u^{k+1} = D^{-1}(\mu f + \lambda\phi^T(d^k - f^k)).$$

La principale difficulté pour calculer u_{k+1} est donc d'inverser une matrice qui peut être de taille conséquente. Nous pouvons envisager plusieurs stratégies :

- Nous calculons l'inverse de la matrice D une fois pour toutes et nous l'utilisons ensuite à chaque itération. Le coût est l'inversion d'une matrice potentiellement très grande. L'avantage est qu'une fois que cette inversion est réalisée, le calcul de u_{k+1} consiste essentiellement à une multiplication de matrice.
- À chaque itération, nous inversons le système 2.24 avec un algorithme itératif. Les méthodes de type Gauss-Seidel ou des gradients conjugués peuvent convenir ici.
- Nous tirons parti d'une structure particulière de la matrice D provenant d'une structure particulière sur la matrice ϕ. Par exemple, lorsque ϕ est circulante, le système (2.24) est facile à inverser.

Résolution de (2.23b)

La minimisation,

$$\arg\min_d \|d\|_1 + \frac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2,$$

mélange une norme L_1 et une norme L_2. En revanche, les coordonnées sont découplées. La solution s'écrit de manière très simple avec un seuillage doux. Nous pouvons le montrer en utilisant la notion de sous-gradient comme dans [HYZ07]. Nous proposons ici une démonstration en détails car les idées de cette démonstration seront réutilisées à la section suivante. Nous commençons tout d'abord par le lemme suivant.

Lemme 2.6. *Soit $x, y \in \mathbb{R}$ et $\lambda > 0$. Alors, le problème suivant,*

$$\arg\min_x |x| + \frac{\lambda}{2}(x-y)^2,$$

admet une unique solution qui s'écrit,

$$s_{\frac{1}{\lambda}}(y),$$

avec s l'opérateur de seuillage doux.

Démonstration. Nous remarquons tout d'abord que $s_{\frac{1}{\lambda}}(-y) = -s_{\frac{1}{\lambda}}(y)$. Sans nuire à la généralité, nous pouvons donc supposer que $y \geqslant 0$.

Nous avons par ailleurs $(|x| - y)^2 \leqslant (x-y)^2$ puisque $|x| \geqslant x$ et $y \geqslant 0$. Nous savons donc que le minimum est atteint pour $x \geqslant 0$. Pour $x \geqslant 0$, la fonctionnelle devient dérivable. Soit $\phi_y(x) = x + \frac{\lambda}{2}(x-y)^2$ cette fonctionnelle à minimiser. Nous avons $\phi'_y(x) = 1 + \lambda(x-y)$. Il y a alors deux cas possibles.

Si $y \geqslant \frac{1}{\lambda}$ alors la dérivée s'annule en un unique point $y - \frac{1}{\lambda}$. Le minimum est atteint en cette même valeur.

Si $y < \frac{1}{\lambda}$, la dérivée est toujours positive, le minimum est donc atteint en 0. Dans tous les cas nous avons $x = s_{\frac{1}{\lambda}}(y)$. ■

Vient ensuite la propriété qui nous intéresse.

Proposition 2.7. *Soit $x, y \in \mathbb{R}^n$, nous voulons résoudre le problème de minimisation suivant,*

$$\arg\min_x \|x\|_1 + \frac{\lambda}{2}\|x-y\|_2^2. \tag{2.25}$$

Le problème possède une unique solution qui s'écrit,

$$s_\nu(y) \quad \textit{avec} \quad \nu = \frac{1}{\lambda},$$

avec s l'opérateur de seuillage doux.

Démonstration. Le problème est séparable et se décompose en n problèmes identiques,

$$\|x\|_1 + \frac{\lambda}{2}\|x-y\|_2^2 = \sum_{i=1}^n |x_i| + \frac{\lambda}{2}(x_i - y_i)^2.$$

En appliquant le lemme 2.6 à chacun de ces problèmes, nous trouvons, $x_i = s_\nu(y_i)$ pour tout i que nous pouvons rassembler en $x = s_\nu(y)$ d'après les notations 2.5. ■

En appliquant la propriété 2.7 au problème (2.23b), nous trouvons,

$$d^{k+1} = s_{\frac{1}{\lambda}}(\phi(u^{k+1}) + f^k).$$

Nous sommes à présent en mesure d'écrire l'algorithme réalisant une minimisation TV anisotrope.

Data : $\phi, \mu, f, \lambda, err$
Result :
$f^0 = 0$;
$d^0 = 0$;
$D_{\text{inv}} = (\mu I + \lambda \phi^T \phi)^{-1}$;
while $\|u^{k+1} - u^k\|_2 > err$ **do**
  $u^k = u^{k+1}$;
  $u^{k+1} = D_{\text{inv}}(\mu f + \lambda \phi^T (d^k - f^k))$;
  $d^{k+1} = s_{\frac{1}{\lambda}}(\phi u^{k+1} + f^k)$;
  $f^{k+1} = f^k + \phi u^{k+1} - d^{k+1}$;
end

Algorithme 1 : Algorithme de minimisation TV anisotrope

2.4.2 Application au débruitage TV/L_2 isotrope

La section précédente illustre un algorithme qui minimise la norme TV anisotrope c'est à dire la somme en valeur absolue des gradients verticaux et horizontaux. Quitte à compliquer un peu la propriété 2.7, nous pouvons résoudre de manière assez similaire le même algorithme avec une vraie norme TV.

Nous nous intéressons donc à l'algorithme suivant,

$$\arg\min_u TV(u) + \frac{\mu}{2}\|Au - f\|_2^2. \tag{2.26}$$

La norme TV est cette fois-ci plus compliquée puisqu'elle s'écrit,

$$TV(u) = \sum_{i=1}^{n} \sqrt{(\nabla_x^i u)^2 + (\nabla_y^i u)^2}.$$

En suivant les notations de [GO09], nous posons,

$$\begin{aligned} d_x &= \nabla_x u, \\ d_y &= \nabla_y u, \\ \|(d_x, d_y)\|_2 &= \sum_{i=1}^{n} \sqrt{(\nabla_x^i u)^2 + (\nabla_y^i u)^2}. \end{aligned}$$

La minimisation se réécrit donc,

$$\arg\min_{u} \|(d_x, d_y)\|_2 + \frac{\mu}{2}\|Au - f\|_2^2 \quad \text{tel que} \quad \begin{cases} d_x = \nabla_x u \\ d_y = \nabla_y u \end{cases} . \tag{2.27}$$

De la même manière, (2.27) est de la forme de (2.9) si l'on pose,

$$J(u, d_x, d_y) = \|(d_x, d_y)\|_2 + \frac{\mu}{2}\|Au - f\|_2^2, \tag{2.28}$$

$$H(u, d_x, d_y) = \|d_x - \nabla_x(u)\|_2^2 + \|d_y - \nabla_y(u)\|_2^2. \tag{2.29}$$

Nous retrouvons une optimisation contrainte à laquelle nous pouvons appliquer le procédé de Bregman qui se traduit en une succession de minimisations sans contrainte,

$$\begin{cases} (u^{k+1}, d_x^{k+1}, d_y^{k+1}) = \arg\min_{(u,d_x,d_y)} J(u, d_x, d_y) + \frac{\lambda}{2}\|d_x - \nabla_x(u) - b_x^k\|_2^2 \\ \qquad\qquad + \frac{\lambda}{2}\|d_y - \nabla_y(u) - b_y^k\|_2^2 & \text{(2.30a)} \\ b_x^{k+1} = b_x^k + \nabla_x u^{k+1} - d_x^{k+1} & \text{(2.30b)} \\ b_y^{k+1} = b_y^k + \nabla_y u^{k+1} - d_y^{k+1} & \text{(2.30c)} \end{cases}$$

En suivant le même procédé qui nous permet de passer des équations (2.15) aux équations (2.16), nous obtenons,

$$\begin{cases} u^{k+1} = \arg\min_{u} \frac{\mu}{2}\|Au - f\|_2^2 + \frac{\lambda}{2}\|d_x^k - \nabla_x(u) - b_x^k\|_2^2 \\ \qquad\qquad + \frac{\lambda}{2}\|d_y^k - \nabla_y(u) - b_y^k\|_2^2 & \text{(2.31a)} \\ (d_x^{k+1}, d_y^{k+1}) = \arg\min_{(d_x,d_y)} \|(d_x, d_y)\|_2 + \frac{\lambda}{2}\|d_x - \nabla_x(u^{k+1}) - b_x^k\|_2^2 \\ \qquad\qquad + \frac{\lambda}{2}\|d_y - \nabla_y(u^{k+1}) - b_y^k\|_2^2 & \text{(2.31b)} \\ b_x^{k+1} = b_x^k + \nabla_x u^{k+1} - d_x^{k+1} & \text{(2.31c)} \\ b_y^{k+1} = b_y^k + \nabla_y u^{k+1} - d_y^{k+1} & \text{(2.31d)} \end{cases}$$

La minimisation (2.31a) est à nouveau un calcul des moindres carrés. En revanche, nous ne pouvons plus appliquer la propriété 2.7 pour résoudre la minimisation (2.31b) car il y a désormais deux termes quadratiques. Nous pouvons néanmoins contourner le problème en démontrant une variante de la propriété 2.7. Nous commençons tout d'abord par un lemme technique évoqué dans [WY07, GO09] que nous allons démontrer en deux temps en commençant par le lemme suivant.

Lemme 2.8. *Soit $x, y \in \mathbb{R}$. Le problème suivant,*

$$\arg\min_{x,y} \sqrt{x^2+y^2} + \frac{\lambda}{2}(x-a)^2 + \frac{\lambda}{2}(y-b)^2, \tag{2.32}$$

possède une unique solution qui s'écrit,

$$\begin{cases} x = \frac{a}{\sqrt{a^2+b^2}} s_{\frac{1}{\lambda}}(\sqrt{a^2+b^2}) \\ y = \frac{b}{\sqrt{a^2+b^2}} s_{\frac{1}{\lambda}}(\sqrt{a^2+b^2}). \end{cases}$$

Démonstration. Nous nous plaçons dans le plan euclidien représenté par la figure 2.3. Soit A le point de coordonnées (a, b), M le point de coordonnées (x, y) et $\bar{M}$ le projeté orthogonal de M sur la droite (OA). Nous avons donc $O\bar{M} \leqslant OM$ et

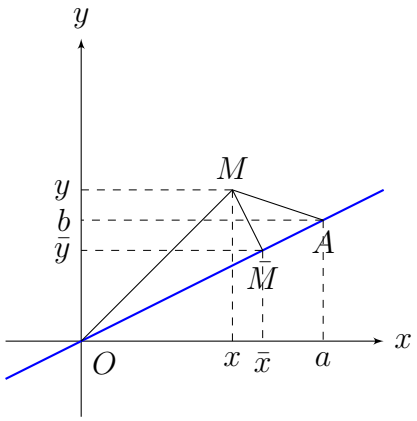


Figure 2.3

$A\bar{M} \leqslant AM$. D'autre part,

$$\begin{aligned} \sqrt{x^2+y^2} + \frac{\lambda}{2}(x-a)^2 + \frac{\lambda}{2}(y-b)^2 &= OM + \frac{\lambda}{2}AM^2 \\ &\geqslant O\bar{M} + \frac{\lambda}{2}A\bar{M}^2 \\ &\geqslant \sqrt{\bar{x}^2+\bar{y}^2} + \frac{\lambda}{2}(\bar{x}-a)^2 + \frac{\lambda}{2}(\bar{y}-b)^2. \end{aligned}$$

Ainsi, quelque soit le point M, son projeté orthogonal $\bar{M}$ sur la droite (OA) donne une fonctionnelle plus petite. Nous pouvons donc supposer que $M \in (OA)$. Nous avons donc $x = \alpha a$ et $y = \alpha b$. En remplaçant ces deux égalités dans (2.32), nous obtenons,

$$\begin{aligned} &\arg\min_{\alpha} \sqrt{a^2+b^2}|\alpha| + \frac{\lambda}{2}(a^2+b^2)(\alpha-1)^2 \\ &\quad = \arg\min_{\alpha} |\alpha| + \frac{\lambda}{2}\sqrt{a^2+b^2}(\alpha-1)^2. \end{aligned}$$

Cette dernière minimisation est une minimisation scalaire qui ne porte que sur le réel α. Nous avons donc réduit un problème vectoriel en un problème portant sur un scalaire. Il suffit ensuite d'appliquer le lemme 2.6 et nous avons,

$$\alpha = s_{\frac{1}{\Lambda}}(1) \quad \text{avec} \quad \Lambda = \lambda\sqrt{a^2+b^2}.$$

donc

$$\begin{cases} x = a\, s_{\frac{1}{\Lambda}}(1) \\ y = b\, s_{\frac{1}{\Lambda}}(1) \end{cases}. \tag{2.33}$$

D'après la propriété 2.4,

$$s_{\frac{1}{\Lambda}}(1) = \frac{1}{\sqrt{a^2+b^2}} s_{\frac{1}{\lambda}}(\sqrt{a^2+b^2}). \tag{2.34}$$

Nous avons donc finalement,

$$\begin{cases} x = \frac{a}{\sqrt{a^2+b^2}} s_{\frac{1}{\lambda}}(\sqrt{a^2+b^2}) \\ y = \frac{b}{\sqrt{a^2+b^2}} s_{\frac{1}{\lambda}}(\sqrt{a^2+b^2}) \end{cases}.$$

■

Nous complétons ensuite ce lemme par une propriété donnant la solution de (2.31b).

Proposition 2.9. *Soit $x, y, u, v \in \mathbb{R}^n$. Le problème d'optimisation suivant,*

$$\arg\min_{(x,y)} \|(x,y)\|_2 + \frac{\lambda}{2}\|x-u\|_2^2 + \frac{\lambda}{2}\|y-v\|_2^2, \tag{2.35}$$

a pour unique solution,

$$\begin{cases} x = u\, s_{\frac{1}{\Lambda}}(1) \\ y = v\, s_{\frac{1}{\Lambda}}(1) \end{cases},$$

avec $\Lambda = \lambda\sqrt{u^2+v^2}$.

Démonstration. Nous commençons par expliciter (2.35),

$$\|(x,y)\|_2 + \frac{\lambda}{2}\|x-u\|_2^2 + \frac{\lambda}{2}\|y-v\|_2^2 \tag{2.36}$$

$$= \sum_{i=1}^{n} \sqrt{x_i^2+y_i^2} + \frac{\lambda}{2}(x_i-u_i)^2 + \frac{\lambda}{2}(y_i-v_i)^2. \tag{2.37}$$

Les n termes de la somme sont indépendants, nous pouvons donc minimiser chaque terme indépendamment. Ainsi, pour tout i, d'après le lemme 2.8,

$$\begin{cases} x_i = u_i s_{\frac{1}{\lambda_i}}(1) \\ y_i = v_i s_{\frac{1}{\lambda_i}}(1) \end{cases},$$

avec $\lambda_i = \lambda\sqrt{u_i^2 + v_i^2}$. D'après les notations 2.5, nous pouvons regrouper les différents seuillages en un vecteur de seuillage ce qui donne,

$$\begin{cases} x = u\, s_{\frac{1}{\Lambda}}(1) \\ y = v\, s_{\frac{1}{\Lambda}}(1) \end{cases},$$

avec $\Lambda = \lambda\sqrt{u^2 + v^2}$. ■

Finalement, l'équation (2.31b) se résume aux deux seuillages suivants,

$$\begin{cases} d_x^{k+1} = (\nabla_x u^{k+1} + b_x^k) s_{\frac{1}{\Lambda}}(1) & \text{(2.38a)} \\ d_y^{k+1} = (\nabla_y u^{k+1} + b_y^k) s_{\frac{1}{\Lambda}}(1) & \text{(2.38b)} \end{cases}$$

avec $\Lambda = \lambda\sqrt{(\nabla_x u^{k+1} + b_x^k)^2 + (\nabla_y u^{k+1} + b_y^k)^2}$.

Explicitons maintenant la solution de l'équation (2.31a) dont nous avions dit qu'elle était un simple calcul des moindres carrés. En effet, l'expression est convexe, coercive et différentiable. Il suffit donc de calculer le lieu où la différentielle s'annule. En différenciant, nous obtenons donc,

$$\mu\langle Au - f, A(\cdot)\rangle + \lambda\langle d_x^k - \nabla_x(u) - b_x^k, -\nabla_x(\cdot)\rangle + \lambda\langle d_y^k - \nabla_y(u) - b_y^k, -\nabla_y(\cdot)\rangle = 0.$$

En identifiant $\mathbb{R}^n$ et son dual par le produit scalaire canonique, cette dernière expression peut s'écrire,

$$\mu A^T(Au - f) + \lambda\nabla_x^T(\nabla_x u + b_x^k - d_x^k) + \lambda\nabla_y^T(\nabla_y u + b_y^k - d_y^k) = 0.$$

En isolant u, nous obtenons finalement,

$$(\mu A^T A + \lambda\nabla_x^T\nabla_x + \lambda\nabla_y^T\nabla_y)u = \mu A^T f + \lambda\nabla_x^T(d_x^k - b_x^k) + \lambda\nabla_y^T(d_y^k - b_y^k).$$

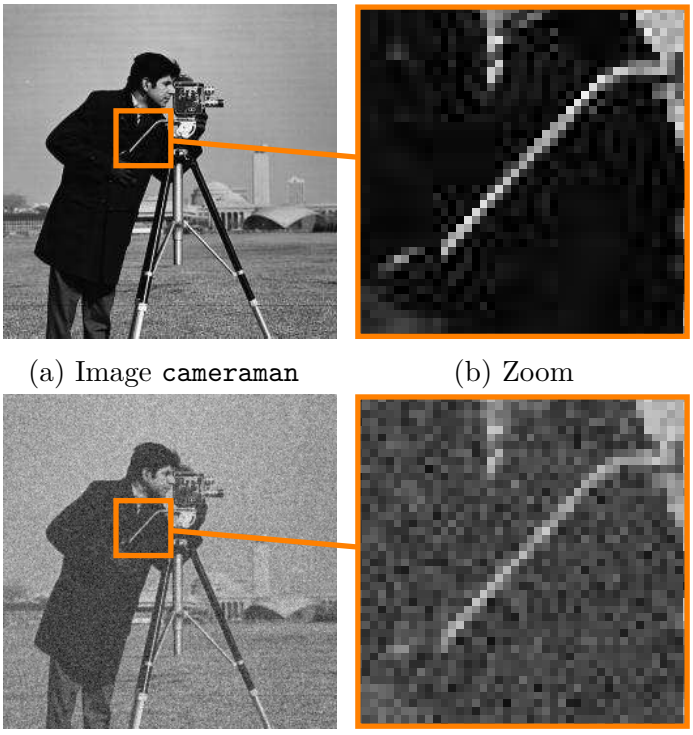
En posant $D = \mu A^T A + \lambda\nabla_x^T\nabla_x + \lambda\nabla_y^T\nabla_y$ et en supposant que D est inversible, nous trouvons,

$$u^{k+1} = D^{-1}(\mu A^T f + \lambda\nabla_x^T(d_x^k - b_x^k) + \lambda\nabla_y^T(d_y^k - b_y^k)).$$

Nous sommes à présent en mesure de décrire l'algorithme de minimisation TV isotrope à l'algorithme 2.

Data : μ, A, f, λ, err
Result : u^{k+1}
$b_x^0 = b_y^0 = d_x^0 = d_y^0 = 0$;
$D_{\text{inv}} = (\mu A^T A + \lambda \nabla_x^T \nabla_x + \lambda \nabla_y^T \nabla_y)^{-1}$;
$c = \mu A^T f$;
$k = 0$;
$u_0 = 0$;
$u_1 = err$;
while $\|u^{k+1} - u^k\|_2 > err$ **do**
  $u^{k+1} = D_{\text{inv}}(c + \lambda \nabla_x^T (d_x^k - b_x^k) + \lambda \nabla_y^T (d_y^k - b_y^k))$;
  $\Lambda = \lambda \sqrt{(\nabla_x u^{k+1} + b_x^k)^2 + (\nabla_y u^{k+1} + b_y^k)^2}$;
  $d_x^{k+1} = (\nabla_x u^{k+1} + b_x^k) s_{\frac{1}{\Lambda}}(1)$;
  $d_y^{k+1} = (\nabla_y u^{k+1} + b_y^k) s_{\frac{1}{\Lambda}}(1)$;
  $b_x^{k+1} = b_x^k + \nabla_x u^{k+1} - d_x^{k+1}$;
  $b_y^{k+1} = b_y^k + \nabla_y u^{k+1} - d_y^{k+1}$;
  $k = k + 1$;
end

Algorithme 2 : Algorithme de minimisation TV isotrope



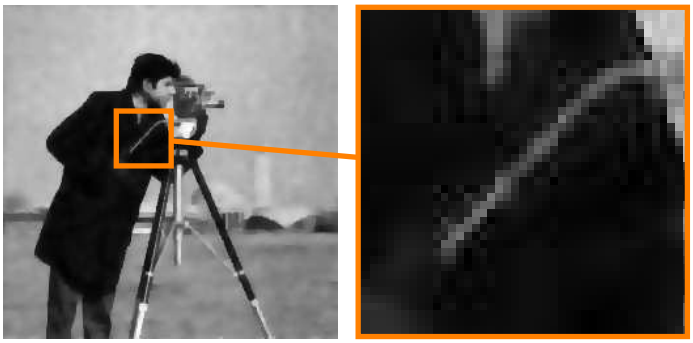
(a) Image `cameraman` (b) Zoom

(c) Image `cameraman` bruit gaussien, 10% (d) Zoom

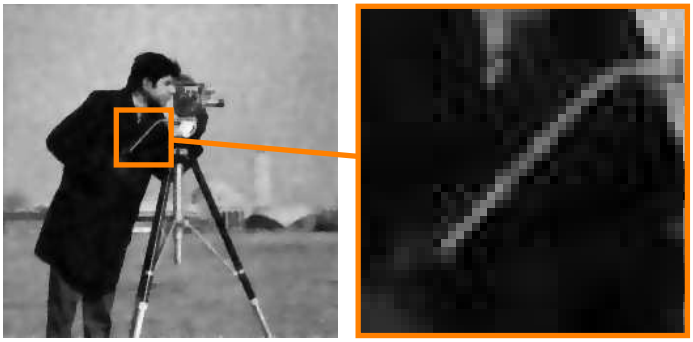
Figure 2.4 – Image `cameraman` bruitée

La figure 2.5 montre le résultat des algorithmes 1 et 2 de débruitage anisotrope et isotrope sur l'image bruitée `cameraman` présentée à la figure 2.4. Nous avons pris, $\lambda = \frac{1}{2}, \mu = 8$ et la condition d'arrêt est une erreur de $5 \cdot 10^{-2}$.

Le débruitage anisotrope fait une approximation de la norme TV. Il surévalue le gradient lorsque les composantes sur selon l'axe des x et des y sont voisines en valeur absolue. Cependant, sur l'image `cameraman`, comme le montrent les agrandissements de l'image, cette approximation est acceptable. C'est pourquoi, lorsque nous utiliserons une minimisation TV, nous prendrons la minimisation anisotrope.



(a) Débruitage TV/L_2 anisotrope (b) Zoom



(c) Débruitage TV/L_2 isotrope (d) Zoom

Figure 2.5 – Débruitage de l'image `cameraman`

2.4.3 Application à la reconstruction TV d'une image

Dans les deux applications précédentes, nous disposons de l'image bruitée f à laquelle nous appliquons un algorithme de débruitage. Dans cette section, nous faisons l'hypothèse que nous ne disposons plus que de la donnée Au de l'image u. Nous souhaitons reconstruire u en connaissant seulement Au. Ce problème est

considéré dans [GO09] où il est question de reconstruire une image à partir de données partielles sur la transformée de Fourier de cette image. Cette supposition simplifie les calculs et permet de traiter des images sans recourir à un découpage préalable.

Ici, dans un premier temps, nous ne particularisons pas la matrice A. Nous savons juste qu'elle est de rang maximal et qu'elle a beaucoup plus de colonnes que de lignes. L'algorithme utilisé est le Bregman scindé et le problème est le suivant,

$$\arg\min_u J(u) \quad \text{tel que} \quad \|Au-b\|_2^2 < \sigma^2. \tag{2.39}$$

Ce type de problème avec contrainte n'entre pas directement dans le cadre du Bregman scindé qui se propose de résoudre efficacement des problèmes sans contrainte. En revanche, nous pouvons utiliser la transformation de Bregman classique qui transforme un problème avec contrainte en une succession de problèmes sans contrainte. Nous avons donc la succession de problèmes suivante,

$$\begin{cases} u^{k+1} = \arg\min_u J(u) + \dfrac{\mu}{2}\|Au-b^k\|_2^2 & \text{(2.40a)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.40b)} \end{cases}$$

En fait, la contrainte dans (2.39) est une contrainte d'inégalité alors que la contrainte que prend en charge la transformation de Bregman doit être une égalité. L'algorithme étant itératif, nous traitons cette inégalité comme une égalité c'est à dire que nous supposons que nous voulons $Au = b$ mais en réalité, nous arrêterons l'itération lorsque la condition $\|Au-b\|_2^2 < \sigma^2$ sera remplie.

Il reste donc à résoudre les minimisations (2.40a) qui rentrent cette fois dans le cadre du Bregman scindé. Nous avons alors l'algorithme suivant,

$$\begin{cases} \begin{cases} u^{k+1} = \arg\min_u \frac{\mu}{2}\|Au-b^k\|_2^2 + \frac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 \\ d^{k+1} = \arg\min_d \|d\|_1 + \frac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} \end{cases} & \text{(2.41a)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.41b)} \end{cases}$$

Pour simplifier, nous choisissons une unique itération pour la boucle intérieure. Ainsi,

$$\begin{cases} u^{k+1} = \arg\min_u \dfrac{\mu}{2}\|Au-b^k\|_2^2 + \dfrac{\lambda}{2}\|d^k - \phi(u) - f^k\|_2^2 & \text{(2.42a)} \\ d^{k+1} = \arg\min_d \|d\|_1 + \dfrac{\lambda}{2}\|d - \phi(u^{k+1}) - f^k\|_2^2 & \text{(2.42b)} \\ f^{k+1} = f^k + \phi(u^{k+1}) - d^{k+1} & \text{(2.42c)} \\ b^{k+1} = b^k + b - Au^{k+1} & \text{(2.42d)} \end{cases}$$

Nous pouvons ensuite dériver l'algorithme de reconstruction TV anisotrope

Data : $\phi, \mu, A, b, \lambda, err$
Result :
$b^0 = b$;
$f^0 = 0$;
$d^0 = 0$;
$D_{\text{inv}} = (\mu A^T A + \lambda \phi^T \phi)^{-1}$;
while $\|Au^{k+1} - b\|_2 > err$ **do**
  $u^k = u^{k+1}$;
  $u^{k+1} = D_{\text{inv}}(\mu A^T b^k + \lambda \phi^T (d^k - f^k))$;
  $d^{k+1} = s_{\frac{1}{\lambda}}(\phi u^{k+1} + f^k)$;
  $f^{k+1} = f^k + \phi u^{k+1} - d^{k+1}$;
  $b^{k+1} = b^k + b - Au^{k+1}$;
end

Algorithme 3 : Algorithme de reconstruction TV anisotrope

Nous choisissons dans cet algorithme d'inverser la matrice D plutôt que de recourir à des algorithmes itératifs qui calculeraient la solution. Cela limite la quantité de donnée que l'algorithme peut traiter. Nous choisissons ici de limiter les images à des tailles de 32×32 et lorsque nous traiterons des images plus grandes, nous la découpons en imagettes qui sont traitées une par une. Afin de limiter les effets de blocs, les imagettes découpées se chevauchent. Lors de la reconstruction, une moyenne est effectuée là où une ou plusieurs imagettes se chevauchent. Nous proposons une reconstruction L_2 qui minimise l'énergie de l'image tout en respectant la contrainte $Au = b$ et une reconstruction avec une minimisation TV contrainte et non contrainte. Pour les algorithmes de Bregman, nous fixons $\mu = 1, \lambda = 1000$ et la condition d'arrêt est une erreur de 0.1.

Taux de mesures (%)	Reconstruction L_2	Reconstruction TV/L_2 contrainte	Reconstruction TV/L_2 non contrainte
5	5.55	25.06	25.23
10	6.61	27.1	27.31
20	8.85	29.65	29.79
30	11.68	31.57	31.65

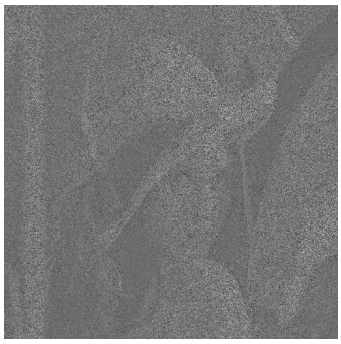
Table 2.1 – Calcul du PSNR des différentes reconstructions

Nous observons que la reconstruction L_2 à la figure 2.6b est très mauvaise même avec 30% des données. Les autres reconstructions, qui prennent en compte la géométrie de l'image à travers la norme TV donnent de meilleurs résultats. À partir de 5% des données, l'effet de blocs apparaît.

En revanche, la reconstruction TV se passe plutôt bien. L'image est correctement reconstruite même avec seulement 10% des données.



(a) Image *lena* 512×512



(b) Reconstruction L_2, 30% des données



(c) Reconstruction Bregman contraint, 5% des données



(d) Reconstruction Bregman contraint, 10% des données



(e) Reconstruction Bregman contraint, 20% des données



(f) Reconstruction Bregman contraint, 30% des données

Figure 2.6 – Reconstruction par blocs 32×32 avec un chevauchement de 2 pixels

2.5 Conclusion

Dans ce chapitre, nous avons présenté les algorithmes de type Bregman en commençant par la distance de Bregman. Nous avons ensuite introduit l'algorithme de Bregman le plus simple, l'itération de Bregman qui résout des problèmes contraints en les transformant en une suite de problèmes non contraints. L'itération de Bregman se simplifie assez notablement lorsque la contrainte est quadratique ce qui sera le cas de toutes les minimisations auxquelles nous serons confrontés. Nous avons ensuite étudié une technique de scindement qui nous a permis de résoudre des problèmes non contraints aux fonctionnelles plus compliquées. La technique de scindement introduit une contrainte et ce nouveau problème contraint est ensuite résolu de manière classique avec l'itération de Bregman.

Nous détaillons ensuite pas à pas la mise en œuvre de deux minimisations TV avec deux modélisations différentes de la norme TV, isotrope et anisotrope. Nous donnons ensuite deux algorithmes en pseudo-code qui réalisent ces deux minimisations. Nous illustrons ces algorithmes en réalisant le débruitage d'une image suivant le modèle ROF et la reconstruction d'une image dans le cadre d'une acquisition compressée.

À travers ces deux applications, les algorithmes de Bregman se montrent très souples et montrent qu'ils peuvent être mis à profit dans des problèmes posés par l'acquisition compressée. Dans le chapitre suivant, nous verrons que nous pouvons ajouter sans difficulté d'autres contraintes aux minimisations rencontrées pour réaliser la reconstruction des lieux d'une signature dans une image multispectrale dans le cas d'une acquisition compressée.

Chapitre 3

Détection compressée de signatures

Sommaire

3.1 Détection de signature simple 48
3.2 Détection compressée de signature 51
3.2.1 Modèle de mesures 52
3.2.2 Détection compressée 53
3.2.3 Formulation du problème 55
3.3 Mise en œuvre et illustrations 58
3.4 Conclusion 64

Lorsqu'on dispose de données multispectrales, une des premières applications intéressantes est la détection de signatures spectrales. D'abord parce qu'il s'agit d'une opération en apparence simple de recherche et aussi parce qu'une signature spectrale identifie souvent un ou plusieurs objets précis.

Dans le cadre de cette thèse qui s'intéresse à la détection de points d'intérêt dans une image multispectrale, nous proposons dans ce chapitre une détection de signature dans une image multispectrale par acquisition compressée. Notre point de départ est une minimisation introduite par Guo et Osher dans [GWO09]. Cet algorithme de détection de signature est original à plusieurs titres. D'abord parce qu'il utilise une minimisation L_1 qui est plutôt employée pour des problèmes de parcimonie. Et ensuite car il utilise une propriété des algorithmes de minimisation L_1 qui est un désavantage pour des problèmes de parcimonie mais qui s'avère ici cruciale pour le bon fonctionnement de l'algorithme. Nous résoudrons les minimisations obtenues avec les algorithmes introduits au chapitre précédent.

Après avoir présenté la minimisation de Guo et Osher, nous introduisons des éléments justifiant la pertinence de cette minimisation. Nous généralisons ensuite

cet algorithme de plusieurs manières. D'abord nous l'adaptons pour qu'il puisse être utilisé avec des données issues d'une acquisition compressée. D'autre part, pour améliorer les résultats, nous allons modifier la fonction objectif en ajoutant à la norme L_1 des normes qui introduisent une dimension géométrique. Nous terminerons enfin par des illustrations comparant les deux algorithmes sur des images synthétiques et réelles.

Notons que ce chapitre a fait l'objet d'une publication dans *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* [RHCBT13] et d'une revue acceptée en correction dans *Transactions on Geoscience and Remote Sensing (TGARS)*.

3.1 Détection de signature simple

Tout d'abord, nous allons détailler l'algorithme introduit dans [GO11]. Il s'agit de détecter dans une image multispectrale X les lieux d'une signature spectrale s. Pour cela, les auteurs proposent la minimisation suivante,

$$\arg\min_{u \geqslant 0} \|u\|_1 \quad \text{tel que} \quad X^T u = s. \tag{3.1}$$

L'image X est une image multispectrale sous sa forme matricielle. Elle est de taille $n_P \times n_B$ où n_P est le nombre de pixels et n_B le nombre de bandes. La signature spectrale à détecter est un vecteur de taille n_B. Le vecteur u inconnu est de taille n_P. Il est non nul si le pixel correspondant est de signature s et nul si ce n'est pas le cas. L'algorithme ne considère que les vecteurs u qui ont tous leurs coefficients positifs.

La minimisation précédente peut être généralisée à la détection simultanée de plusieurs signatures. Supposons que l'on veuille détecter k signatures $s_1, \ldots, s_k$. Nous formons alors k problèmes de minimisation,

$$\arg\min_{X^T u_k = s_k} \|u_k\|_1, \quad k = 1, \ldots, n,$$

qui sont équivalents à,

$$\arg\min_{X^T U = S} \|U\|_1,$$

avec $U = [u_1, \ldots, u_k]$ et $S = [s_1, \ldots, s_k]$. L'égalité $X^T U = S$ peut se récrire $\bar{X}^T u = s$ avec $\bar{X} = I \otimes X^T$, $u = \mathrm{vec}(U)$ et $s = \mathrm{vec}(S)$. Nous avons ainsi réduit la détection multiple à une détection simple sur un vecteur k fois plus grand.

Afin de comprendre pourquoi une telle minimisation fonctionne, nous allons donner une explication intuitive. Premièrement, intéressons-nous à la contrainte $X^T u = s$. Le terme $X^T u$ est une combinaison linéaire à coefficients dans u des

lignes de X. Les lignes de X étant les pixels de l'image multispectrale, l'égalité dit qu'une combinaison linéaire à coefficients dans u est égale à la signature spectrale s que l'on recherche. Si la signature s est présente dans X, une première solution serait de choisir u qui sélectionne cette signature. Si la signature est présente en n endroits, une seconde solution consiste à choisir u qui sélectionne ces n endroits avec un coefficient $\frac{1}{n}$, la norme L_1 étant alors la même. On peut aussi imaginer que, même si l'image multispectrale X ne contient pas la signature spectrale s, comme le nombre de pixels est important, il existe tout de même une solution combinant des pixels dont la signature est différente de s et donnant malgré tout s. Les trois solutions qui viennent d'être évoquées représentent trois grands groupes de solutions.

1. La première représente les solutions de faible norme L_0 qui ne sélectionne pas toutes les signatures s. On a donc une détection partielle des lieux d'intérêt.
2. La deuxième représente les solutions vers lesquelles on veut converger. Le coefficient de u est non nul si et seulement si le pixel correspondant est de signature s.
3. La troisième représente les solutions qui, au mieux, font de la fausse détection puisque qu'elle combine des pixels qui ne sont pas de signatures s.

C'est maintenant qu'intervient la fonction objectif dans la minimisation (3.1). La norme L_1 écarte les solutions de type 3 puisque pour générer la signature s à partir de pixels différents de s, u doit vraisemblablement être de norme L_1 plus grande que 1. Dès lors, dans la minimisation, les solutions de type 1 et 2 sont privilégiées car la norme L_1 de u vaut alors 1.

Ensuite, les solutions de type 1 sont écartées car l'algorithme va converger vers une solution qui a la plus grande norme L_0. Nous pouvons formaliser cette observation dans l'hypothèse suivante.

Les algorithmes utilisés pour résoudre des minimisations telle que (3.1) *convergent vers la solution qui a la norme L_0 la plus grande parmi les solutions qui ont la norme L_1 la plus petite.*

Cette hypothèse se justifie de la manière suivante. L'ensemble des solutions sans minimiser la norme, c'est à dire l'ensemble des éléments satisfaisant la contrainte de la minimisation forme un sous-espace affine. Parmi ces éléments, ceux qui minimisent la norme L_1 appartiennent à la sphère L_1 de plus petit rayon intersectant le sous-espace affine. Cette intersection est une face de la sphère L_1 qui est un polytope. Or, la sphère L_1 a la propriété que les points situés sur une face de dimension d sont $(d+1)$-parcimonieux. Finalement, si on choisit au hasard une solution dans cette face, elle sera de support maximum.

La figure 3.1 illustre cette propriété en dimension 2. La boule L_1 est l'ensemble représentée en bleu, le sous-espace est la droite tracée en noir et les solutions de la

minimisation sont représentées en rouge. On voit que l'ensemble des solutions est

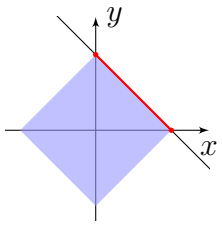


Figure 3.1 – Boule L_1 minimale

une face de la boule L_1 et que parmi ces solutions, seulement deux d'entre elles ont un support égal à un. Les autres ont un support égal à deux et sont en nombre infini. Avec une grande probabilité, l'algorithme converge vers une de ces solutions.

Cette hypothèse permet de monter que toutes les solutions de norme L_0 maximale parmi les solutions de norme L_1 minimale ont le même support.

Proposition 3.1. *Soient u_1 et u_2 deux solutions de norme L_0 maximale parmi les solutions de norme L_1 minimale, alors elles ont le même support.*

Démonstration. En effet, pour tout λ tel que $0 < \lambda < 1$, $u_\lambda = (1-\lambda)u_1 + \lambda u_2$ est solution. On a donc

$$\begin{aligned} \|u_1\|_1 &\leqslant \|u_\lambda\|_1 \\ &\leqslant \|(1-\lambda)u_1 + \lambda u_2\|_1 \\ &\leqslant (1-\lambda)\|u_1\|_1 + \lambda\|u_2\|_1 \\ &\leqslant \|u_1\|_1. \end{aligned}$$

D'après la condition d'égalité de l'inégalité triangulaire, on trouve que u_1 et u_2 ont même signe. Ainsi,

$$\|(1-\lambda)u_1 + \lambda u_2\|_0 = \mathrm{Card}(\mathrm{Supp}(u_1) \cup \mathrm{Supp}(u_2)).$$

Mais puisque u_λ est une solution de norme L_1 minimale, on a

$$\|u_\lambda\|_0 \leqslant \|u_1\|_0$$

d'où

$$\mathrm{Card}(\mathrm{Supp}(u_1)) \geqslant \mathrm{Card}(\mathrm{Supp}(u_1) \cup \mathrm{Supp}(u_2))$$

donc $\mathrm{Supp}(u_1) = \mathrm{Supp}(u_2)$. ∎

Ainsi, quelque soit la solution, nous détectons les mêmes lieux de la signature s.

3.2 Détection compressée de signature

La détection de signature réalisée précédemment suppose la connaissance de l'image X pour résoudre la minimisation. Étant donné l'énorme quantité de données que peut représenter l'image X, il serait intéressant d'acquérir l'image de manière compressée. Ensuite, on reconstruit l'image lorsqu'on en a besoin pour faire une recherche de signature. Ce paradigme est schématisé à la figure 3.2.

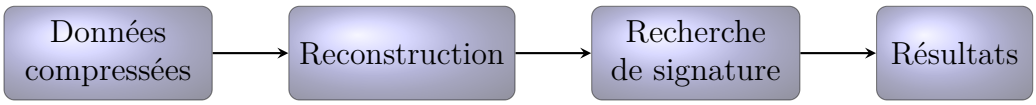


Figure 3.2 – Principe de la détection de signature par acquisition compressée avec reconstruction préalable

Cependant, il serait encore plus intéressant d'éviter la reconstruction et de se servir uniquement des données compressées pour réaliser directement le traitement voulu. Ce nouveau paradigme est schématisé à la figure 3.2 où le bloc de reconstruction, très coûteux, est contourné.

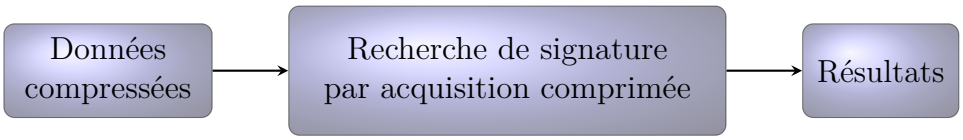


Figure 3.3 – Principe de la détection de signature par acquisition compressée sans reconstruction

Après avoir fixé le modèle de mesures utilisé, nous allons voir comment se passer de l'étape de reconstruction dans le cas où le traitement recherché est la détection de signature présenté à la section précédente.

3.2.1 Modèle de mesures

L'acquisition compressée suppose de faire des mesures linéaires sur un objet. Dans le cas qui nous intéresse, il faut faire des mesures sur une image multispectrale. Si l'image multispectrale X est sous forme vectorisée, l'expression générale d'une mesure linéaire sur X s'écrit,

$$a = f^T \operatorname{vec}(X),$$

où f est un vecteur de mesure et a est le résultat de la mesure. Outre le fait que cette expression contient une vectorisation de X qui rend difficile son exploitation mathématique, cette modélisation apparaît peu réaliste d'un point de vue physique

puisqu'elle suppose de pouvoir combiner indistinctement n'importe quel pixel avec n'importe quelle bande.

Une seconde modélisation est la suivante,

$$a = Xg.$$

Elle est plus restrictive mais plus réaliste. Pourtant, ce modèle se rapproche de l'acquisition classique puisqu'il ne fait que combiner différentes bandes d'une image hypothétique. Nous nous retrouvons donc, avec ces mesures, une image multispectrale avec des bandes qui résultent de la combinaison à différentes longueurs d'ondes pour différents g. Nous ne sommes plus dans les principes d'une acquisition compressée qui réalise un mélange spatial des pixels.

Ce qui nous amène à une dernière modélisation.

$$a^T = f^T X.$$

Cette fois, les pixels de chaque bande sont combinés selon f et le résultat est stocké bande par bande dans un vecteur a. Cela revient à faire une acquisition compressée classique sur chacune des bandes avec le même vecteur mesure. En rassemblant plusieurs mesures sous une même écriture, nous avons le modèle suivant,

$$M = FX. \tag{3.2}$$

L'image multispectrale X est de taille $n_P \times n_B$. Si nous effectuons m mesures différentes alors M est de taille $m \times n_B$ et F est de taille $m \times n_P$.

Supposons que l'on veuille acquérir une fraction p des données représentées par X. L'image multispectrale X contient $n_B \cdot n_P$ données et M en contient $m \cdot n_B$. Nous avons alors l'équation,

$$m \cdot n_B = p \cdot n_B \cdot n_P,$$

Le nombre m étant un entier, nous prenons donc,

$$m = \lfloor p \cdot n_P \rfloor,$$

où $\lfloor \cdot \rfloor$ est l'opérateur partie entière.

3.2.2 Détection compressée

Lorsque nous effectuons une acquisition suivant le modèle (3.2), nous disposons d'une matrice $M = FX$. La contrainte est toujours $X^T u = s$. Nous avons donc problème suivant,

$$\arg\min_{u \geqslant 0} \|u\|_1 \quad \text{tel que} \quad \begin{cases} \|X^T u - s\|_2 < \sigma \\ M = FX \end{cases} . \tag{3.3}$$

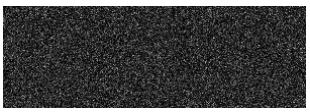
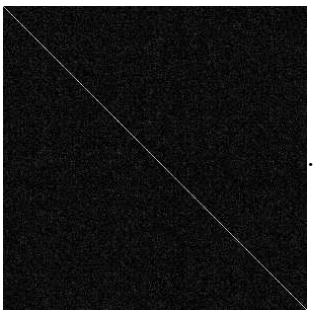
Il faut à présent éliminer X des deux équations $X^T u = s$ et $M = FX$. Une solution serait d'introduire une matrice entre X^T et u de la forme $F^T A$ qui soit égale à l'identité pour pouvoir remplacer $X^T F^T$ par M^T mais c'est impossible car $F^T A$ n'est jamais inversible et donc n'a aucune chance d'être égal à l'identité. En revanche, nous pouvons chercher une matrice A qui vérifie $F^T A \approx I_{n_P}$. Un premier candidat pour A vient du fait que si F est une matrice dont les entrées suivent des lois gaussiennes indépendantes (nous dirons alors simplement par la suite qu'elle est gaussienne), alors nous avons d'après [Rud99],

$$\frac{1}{m} F^T F \approx I_{n_P}.$$

Cela vient grossièrement du fait que si les variables aléatoires $Y_1, \ldots, Y_n$ et $Z_1, \ldots, Z_n$ sont indépendantes et gaussiennes, alors,

$$\mathbb{E}\left[\sum_{i=1}^{n} Y_i Z_i\right] = 0 \quad \text{et} \quad \mathbb{E}\left[\sum_{i=1}^{n} Y_i^2\right] \neq 0.$$

Les entrées diagonales du produit $F^T F$ ont donc une espérance non nulle alors que les autres entrées ont une espérance nulle. Qualitativement, en grande dimension, nous observons que si,

$F =$  alors $\frac{1}{m} F^T F =$ .

Un premier candidat pour A est donc,

$$A = \frac{1}{m} F. \tag{3.4}$$

Un autre candidat est obtenu en résolvant la minimisation suivante,

$$\arg\min_A \left\| F^T A - I \right\|_F, \tag{3.5}$$

où $\|\cdot\|_F$ est la norme de Frobenius qui est la norme euclidienne de la matrice vectorisée. C'est un problème de minimisation classique dont la solution s'écrit à l'aide de la pseudo-inverse. Nous avons en effet, $A = (F^T)^+$, où $+$ est l'opérateur de passage à la pseudo-inverse. Comme F est une matrice de mesures, nous supposons

qu'elle est de rang maximal, nous pouvons alors expliciter la pseudo-inverse. Nous obtenons,

$$A = (FF^T)^{-1}F.$$

Il s'agit en fait d'une correction orthogonale par rapport au cas précédent. En effet, il est équivalent de prendre F dont les lignes sont orthogonales dans l'expression F^TF que de prendre l'expression $F^T(FF^T)^{-1}F$.

Nous savons que quelque soit la matrice F de rang maximal, A minimise $\left\|F^TA - I\right\|_F$. Nous pourrions minimiser ensuite $\left\|F^T(FF^T)^{-1}F - I\right\|_F$ mais cette expression est en fait constante pour tout F de rang maximal. Plus précisément, on a,

$$\left\|F^T(FF^T)^{-1}F - I\right\|_F = \sqrt{n_P - m}.$$

Par exemple, si nous choisissons F tel que,

$$F = \begin{pmatrix} I_m & 0 \end{pmatrix},$$

alors,

$$F^T(FF^T)^{-1}F = \begin{pmatrix} I_m & 0 \\ 0 & 0 \end{pmatrix},$$

qui est un très mauvais candidat puisque la distance à l'identité n'est pas également répartie. Pour mesurer la qualité de la matrice A choisie, nous allons nous intéresser à la norme infinie et à la minimisation,

$$\arg\min_{\lambda} \|\lambda F^T(FF^T)^{-1}F - I_{n_P}\|_\infty. \tag{3.6}$$

Cette minimisation calcule le paramètre λ idéal pour que l'erreur maximale commise soit la plus petite possible. Dans le cas d'une matrice de mesures F gaussienne, nous savons déjà que,

$$\frac{1}{m}F^TF \approx I_{n_P}.$$

De la même manière, nous avons,

$$\frac{1}{n_P}FF^T \approx I_m.$$

Nous prenons alors l'approximation,

$$F^T(FF^T)^{-1}F \approx \frac{m}{n_P}I_{n_P}, \tag{3.7}$$

c'est à dire $\lambda = \frac{n_p}{m}$. La figure 3.4 vient confirmer cette égalité sur une image de 100 pixels avec $p = \frac{1}{3}$. Nous retrouvons bien $\lambda = 3$.

La figure 3.5 montre le calcul de cette norme en fonction du taux de mesures pour différentes matrices de mesures. Nous observons que la correction orthogonale diminue sensiblement l'erreur maximale.

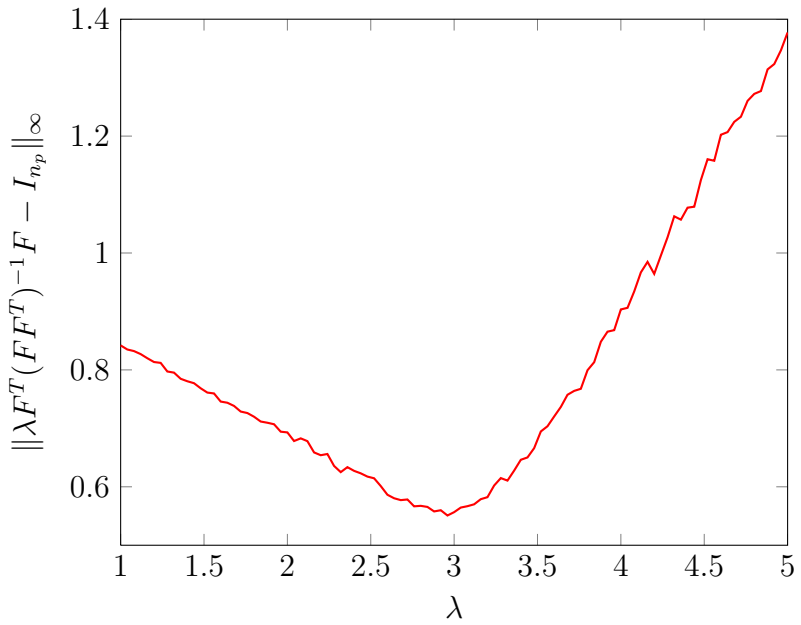


Figure 3.4 – Erreur maximale pour différentes valeur de λ

3.2.3 Formulation du problème

Nous sommes à présent en mesure de formuler le problème de détection compressée de signature. En injectant l'approximation (3.7) dans $X^T u = s$, nous trouvons,

$$X^T F^T (FF^T)^{-1} Fu \approx \frac{m}{n_p} s,$$

et donc,

$$M^T (FF^T)^{-1} Fu \approx \frac{m}{n_p} s,$$

car $M = FX$. Nous avons bien éliminé X de l'équation et il nous faut donc résoudre la minimisation suivante,

$$\arg\min_{u \geqslant 0} \|u\|_1 \quad \text{tel que} \quad \left\| M^T (FF^T)^{-1} Fu - \frac{m}{n_P} s \right\|_2 < \sigma. \tag{3.8}$$

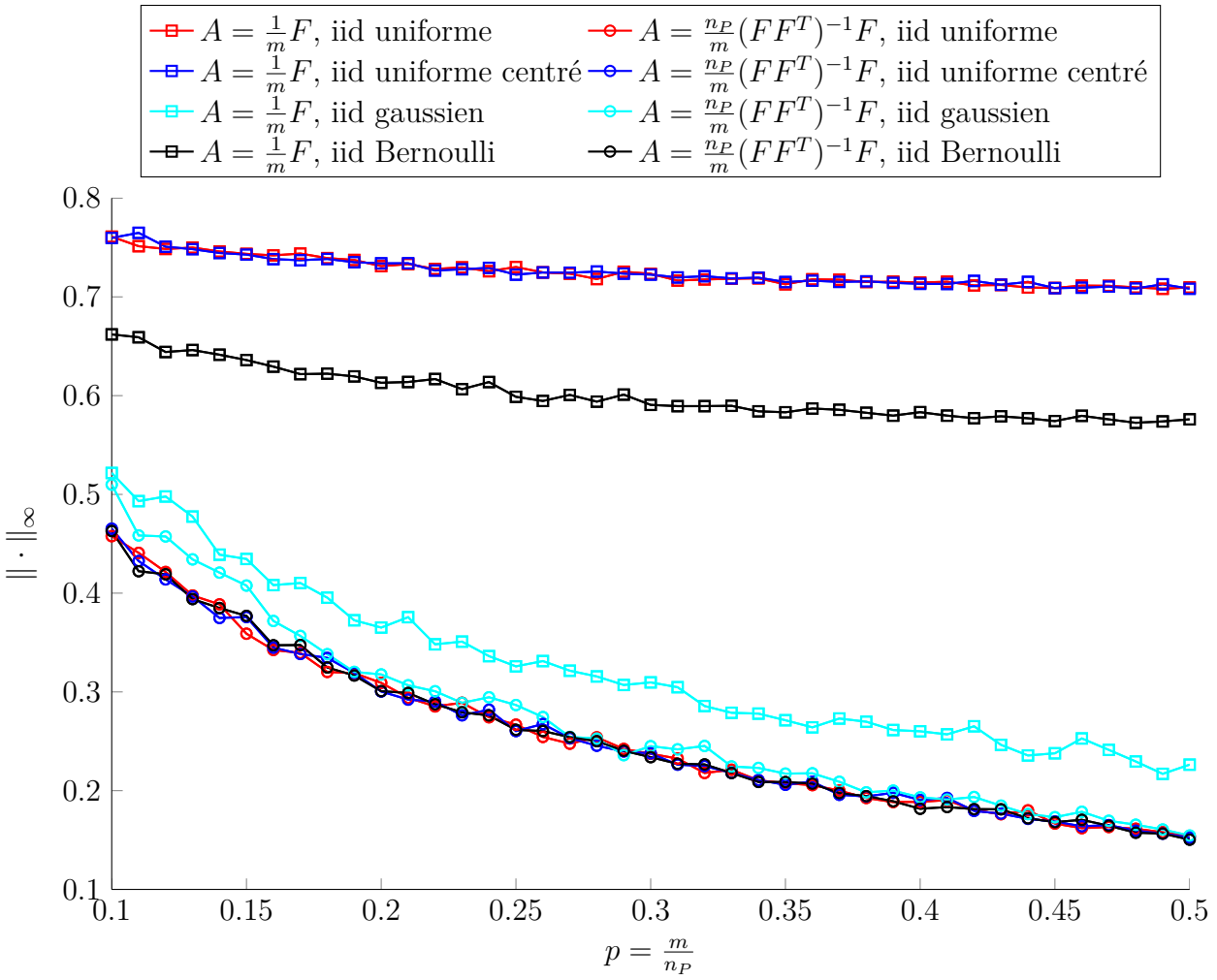


Figure 3.5 – Comparaison de différentes matrices de mesures

3.3 Mise en œuvre et illustrations

Nous illustrons l'algorithme de détection compressée de signature sur plusieurs images. Tout d'abord sur une image multispectrale réelle de taille 64×64, à 16 bandes issue d'une banque d'images d'objets communs [1] présenté dans [HFM04]. La signature spectrale s qui nous intéresse est celle du nez de Sylvester, représentée à la figure 3.6.

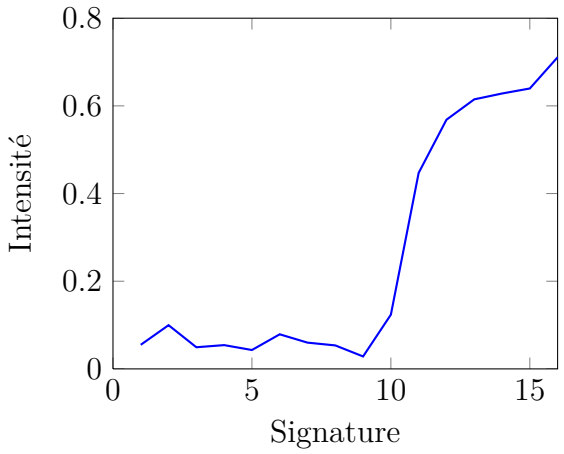


Figure 3.6 – Signature du nez de Sylvester

Les résultats sont montrés à la figure 3.7. La figure 3.7a montre l'image originale en fausses couleurs calculée par une décomposition en valeurs singulières. La figure 3.7c montre les résultats de l'algorithme classique qui détecte bien la structure du nez. Nous avons ensuite testé deux fonctionnelles différentes pour la reconstruction des lieux des signatures.

La première, à la figure 3.7d est une simple norme L_1. Nous remarquons que la reconstruction n'arrive pas à détecter le nez correctement. Cela est dû en partie au fait que la norme L_1 ne fait pas de distinction entre les pixels d'où l'aspect dispersé de la détection.

La seconde fonctionnelle est un mélange de norme L_1 et de norme TV. L'introduction de la norme TV permet d'ajouter une dimension géométrique à la reconstruction. La reconstruction est ainsi meilleure comme le montre la figure 3.7e.

La table 3.1 montre les erreurs de détection pour différents taux de mesures. La dernière ligne montre les erreurs de l'algorithme classique qui dispose de toute l'image. Les erreurs sont calculées de la manière suivante. Une vérité de terrain est

1. Disponible à l'adresse,
`http://www2.cmp.uea.ac.uk/Research/compvis/MultiSpectralDB.htm`

Taux de mesures (%)	Faux positifs (%)	Faux négatifs (%)	Erreurs (%)
10	0.26	0.98	1.25
20	0.24	0.72	0.96
30	0.27	0.9	1.16
40	0.13	0.86	0.99
100	0	1.25	1.25

Table 3.1 – Erreurs de détection sur l'image `sylvester` avec la régularisation TV/L_1

établie par rapport à la signature du nez en prenant toutes les signatures voisines au sens de la norme L_2. Ensuite, toutes les images des lieux des signatures reconstruits sont binarisées par un simple algorithme de groupement de Lloyd-Max et ensuite comparées à la vérité de terrain. Un faux positif représente une détection positive par l'algorithme qui n'est pas vérifiée sur le terrain. Un faux négatif représente une détection négative par l'algorithme qui est pourtant présente dans la vérité de terrain.

La deuxième image est extraite d'une image multispectrale AVIRIS [2]. Nous avons extrait 16 bandes parmi les 224 disponibles et extrait une image de taille 64×64. La signature spectrale que nous voulons détecter est prise sur un des bâtiments et est représentée à la figure 3.8.

La figure 3.9 présente les différents résultats. La régularisation L_1 apparaît ici clairement insuffisante. La régularisation TV/L_1 permet d'obtenir de meilleurs résultats. La table 3.2 compare l'algorithme classique avec une détection compressée pour différents taux de mesures.

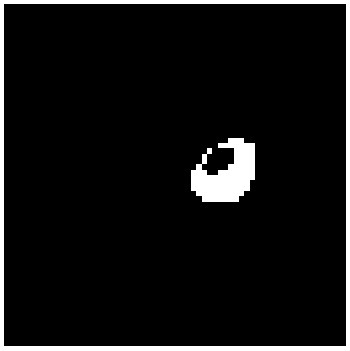
Taux de mesures (%)	Faux positifs (%)	Faux négatifs (%)
10	0.37	3.62
20	0.67	2.5
30	1.16	1.64
40	1.4	1.53
100	1.56	0

Table 3.2 – Erreurs de détection sur l'image `Moffett field` avec la régularisation TV/L_1

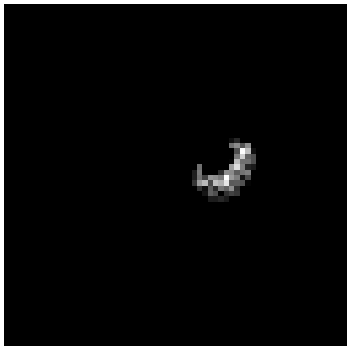
2. Disponible à l'adresse :
`http://aviris.jpl.nasa.gov/html/aviris.freedata.html`



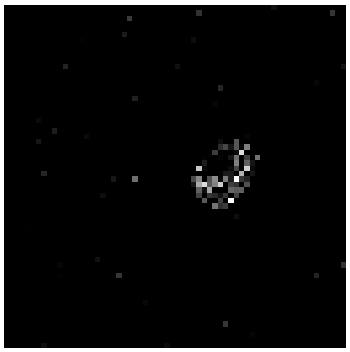
(a) Image `sylvester` en fausses couleurs



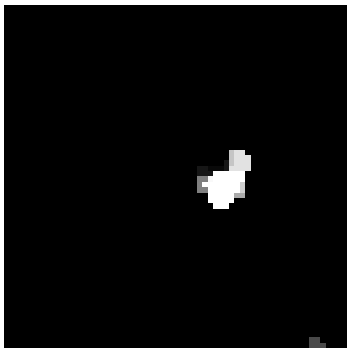
(b) Vérité de terrain



(c) Détection de signature



(d) Détection de signature compressée, 30% des données, fonctionnelle L_1



(e) Détection de signature compressée, 30% des données, fonctionnelle TV/L_1

Figure 3.7 – Détection de signature sur l'image `sylvester`

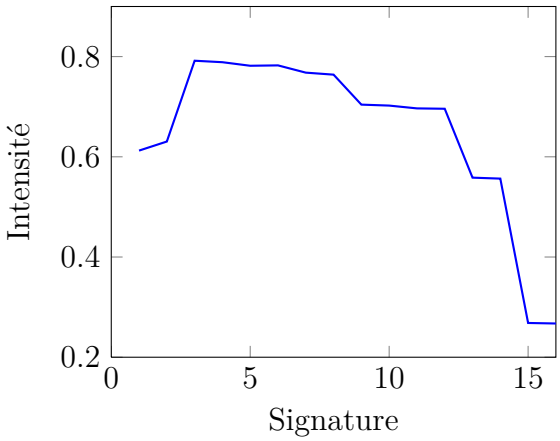
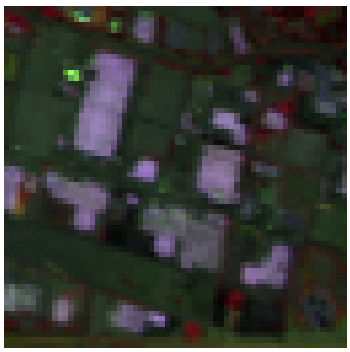
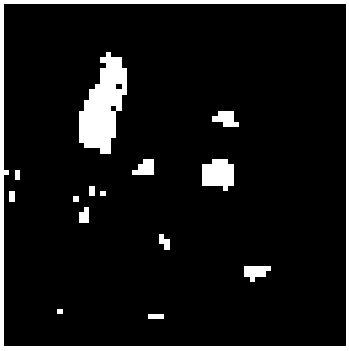


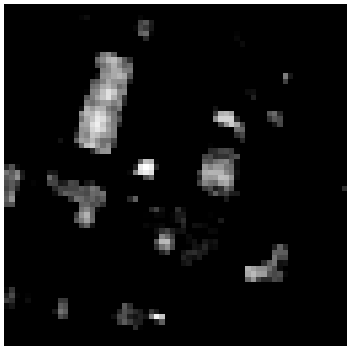
Figure 3.8 – Signature d'un bâtiment de l'image `Moffett Field`



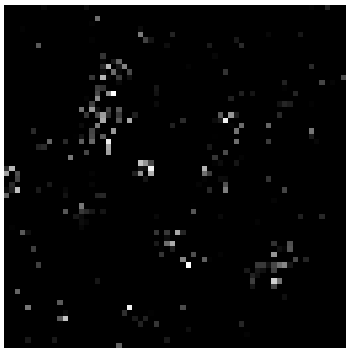
(a) Image `Moffett field` en fausses couleurs



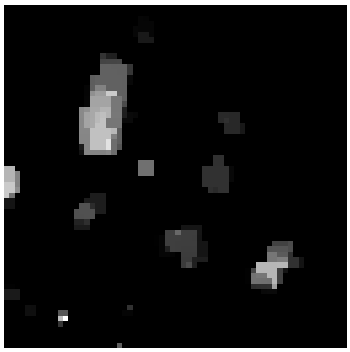
(b) Vérité de terrain



(c) Détection de signature sur a



(d) Détection compressée de signature, 30% des données, régularisation L_1



(e) Détection compressée de signature, 30% des données, régularisation TV/L_1

Figure 3.9 – Détection de signature sur l'image `Moffett field`

3.4 Conclusion

Dans ce chapitre, nous avons présenté un algorithme de détection de Guo et Osher qui repose sur une minimisation L_1. Il se distingue de tous les algorithmes que nous avons rencontrés jusqu'alors car il contient une contrainte de positivité sur la solution. Nous avons ensuite détaillé quelques raisons qui montrent pourquoi cet algorithme fonctionne. Nous proposons ensuite d'adapter cet algorithme pour des données compressées. Nous aboutissons à un système d'équations duquel nous arrivons à éliminer l'image en faisant une hypothèse sur la matrice de mesures. Les résultats de la minimisation ainsi établie suggèrent l'ajout d'une fonctionnelle géométrique pour améliorer la détection. Les algorithmes de Bregman nous permettent de tester une fonctionnelle mixte TV/L_1 qui améliore sensiblement les résultats.

Dans le chapitre qui suit, nous proposons un algorithme basé sur la détection compressée de signatures qui ajoute encore à la dimension géométrique en détectant des motifs.

Chapitre 4

Détection compressée de motifs

Sommaire

4.1 Spectralisation suivant un motif 66
4.1.1 Motif 66
4.1.2 Motif normalisé 67
4.1.3 Opérateur de décalage 67
4.1.4 Spectralisation d'une image en niveaux de gris 69
4.1.5 Spectralisation d'une image multispectrale 71
4.2 Mesures sur une image spectralisée 72
4.2.1 Reconstruction de mesures naïves 73
4.2.2 Reconstruction de mesures glissées 74
4.2.3 Motif de mesures glissées optimal 75
4.3 Mise en œuvre et illustrations 80
4.3.1 Mise en œuvre 80
4.3.2 Tests sur l'image `peppers` 81
4.3.3 Tests sur l'image réelle `Bâche` 85
4.4 Conclusion 87

Le chapitre précédent propose de s'appuyer sur une minimisation L_1 introduite par Guo et Osher [GO11] pour détecter les lieux d'une signature précise dans une image multispectrale. Nous étendons ensuite la détection dans le cas d'une acquisition compressée. Les algorithmes développés au chapitre 2 permettent de résoudre la minimisation résultante et d'améliorer la détection en ajoutant des fonctions de régularisation. Ces fonctions de régularisation introduisent une dimension géométrique à travers l'utilisation d'une norme TV.

Cependant, même si l'ajout de ces fonctions de régularisation permet de corréler les pixels entre eux, à aucun moment nous ne détectons de motifs précis dans l'image multispectrale. Nous nous intéressons à une unique signature. Il serait

intéressant de pouvoir détecter un motif multispectral consistant en un agencement spatial précis de signatures spectrales différentes.

Dans ce chapitre nous allons étendre la détection de signature simple par acquisition comprimée à la détection de motifs par acquisition comprimée dans une image multispectrale. De la même manière que l'introduction de l'acquisition comprimée permet de faire l'économie d'une étape de compression souvent coûteuse, la détection de motifs par acquisition comprimée va faire l'économie de l'étape de reconstruction de l'image multispectrale précédant l'algorithme de détection proprement dit. Ce principe est illustré par la figure 4.1.

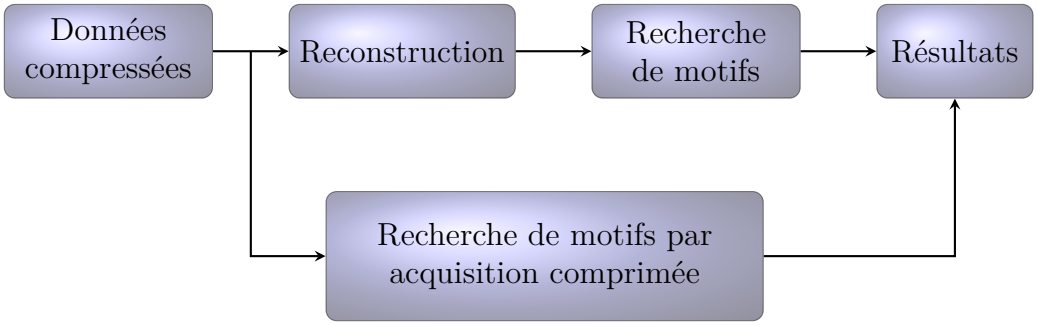


Figure 4.1 – Principe de la détection de motifs par acquisition comprimée

Cette extension repose sur l'idée suivante. Un motif dans une image est une signature dans une image plus complexe. Ainsi, la détection d'un motif dans une image se réduit à la détection d'une signature dans une image plus complexe, ce que nous savons faire. L'opération qui permet de passer d'une image simple à l'image plus complexe qui porte dans ses différentes bandes l'information correspondant à un motif sera appelée la *spectralisation*. Elle dépend uniquement de l'image et de la forme du motif que nous voulons détecter. Nous allons donc commencer par quelques définitions avant d'introduire la spectralisation d'une image puis de la généraliser aux images multispectrales. Ensuite, nous nous intéresserons à la reconstruction de mesures sur l'image spectralisée à partir de mesures sur l'image originale.

Notons que ce chapitre a fait l'objet d'une publication dans *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* [RHCBT13] et d'une revue acceptée en correction dans *Transactions on Geoscience and Remote Sensing (TGARS)*.

4.1 Spectralisation suivant un motif

4.1.1 Motif

Nous nous proposons de détecter une forme dans une image. Cette forme est modélisée par une matrice où sont mis à 0 les pixels qui ne nous intéressent pas et à 1 les pixels décrivant la forme qui nous intéresse. Ainsi, le motif suivant,

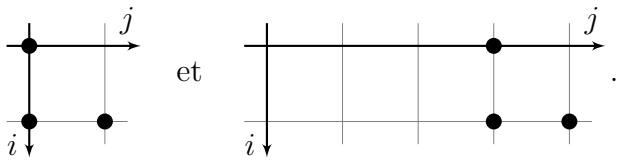
$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \tag{4.1}$$

modélise un motif en forme de crochet.

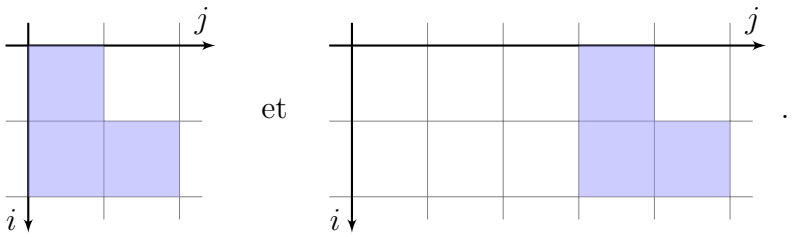
Nous pouvons également décrire ce motif à translation près comme un sous-ensemble de $P \subset \mathbb{Z}^2$. Par exemple, la matrice 4.1 peut être décrite en coordonnées matricielles par les deux sous-ensembles suivant,

$$P_1 = \{(0,0),(1,0),(1,1)\} \quad \text{ou} \quad P_2 = \{(0,3),(1,3),(1,4)\},$$

dont les représentations graphiques sont respectivement,



Dans la suite de ce chapitre nous privilégierons la représentation graphique suivante,



où chaque carré coloré symbolise la coordonnée supérieure gauche.

4.1.2 Motif normalisé

Pour définir la spectralisation, nous avons besoin d'une représentation $P \subset \mathbb{Z}^2$ du motif telle que $(0,0) \in P$. C'est toujours possible car, quitte à translater n'importe quelle représentation, nous pouvons supposer $(0,0) \in P$. D'autre part, il faut également fixer un ordre dans les éléments de P de telle sorte que $(0,0)$ soit le premier élément.

Ainsi, toujours pour le motif en crochet 4.1, la représentation P_1 donne lieu aux deux représentations normalisées suivantes,

$$((0,0),(1,0),(1,1)) \quad \text{et} \quad ((0,0),(1,1),(1,0)).$$

Dorénavant, lorsque nous parlerons de motifs, ils seront implicitement normalisés.

4.1.3 Opérateur de décalage

Comme nous le verrons dans la section suivante, un motif va coder des décalages successifs dans l'image. Nous commençons donc par introduire cet opérateur de décalage.

Étant donnée une matrice I de taille $N \times M$ et un vecteur $p = (i,j) \in \mathbb{Z}^2$, nous définissons l'opérateur S_p sur la matrice I de la manière suivante,

$$[S_p(I)]_{ab} = I_{(a+i-1 \mod N)+1,(b+j-1 \mod M)+1}. \tag{4.2}$$

L'opérateur S_p décale l'entrée $(1,1)$ de la matrice I vers l'entrée $(i+1, j+1)$. Les entrées qui sortent sont réinjectées de l'autre coté de la matrice.

Par exemple, si nous définissons I, p_1 et p_2 de la manière suivante,

$$I = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad p_1 = (-1,0) \quad \text{et} \quad p_2 = (-1,-1), \tag{4.3}$$

nous avons alors,

$$S_{p_1}(I) = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{pmatrix} \quad \text{et} \quad S_{p_2}(I) = \begin{pmatrix} 5 & 6 & 4 \\ 8 & 9 & 7 \\ 2 & 3 & 1 \end{pmatrix}.$$

Nous avons évidemment $S_{(0,0)} = \mathrm{id}$, $S_{p_1+p_2} = S_{p_1} \circ S_{p_2}$ pour tout $p_1, p_2 \in \mathbb{Z}^2$ et donc $S_{p_1}^{-1} = S_{-p_1}$. Nous pouvons écrire la transformation S_p en termes de multiplications de matrices. En effet, soit D_n la matrice de permutation suivante,

$$D_n = \begin{pmatrix} 0 & & & 1 \\ & \ddots & & \\ 1 & & \mathbf{(0)} & \\ & & & \\ & \mathbf{(0)} & & \\ & & \ddots & \\ & & 1 & 0 \end{pmatrix}.$$

Appliquée à gauche, elle décale la matrice vers le bas d'un cran. Appliquée à droite, elle décale la matrice vers la gauche d'un cran. Ainsi pour une matrice I de taille $N \times M$ et un vecteur de décalage $p = (i, j) \in \mathbb{Z}^2$, nous avons,

$$S_p(I) = D_N^i I D_M^{-j}. \tag{4.4}$$

L'opérateur S_p agit sur une matrice. Cependant, la plupart du temps, nous traitons des images mises en ligne c'est à dire que nous ne travaillons pas avec I directement mais avec $u = \text{vec}(I)$. Nous allons donc définir un opérateur qui réalise la même transformation que S_p mais sur des données linéarisées. En somme, nous voulons trouver l'opérateur Q_p qui rend le diagramme suivant commutatif.

$$\begin{array}{ccc} I & \xrightarrow{S_p} & \tilde{I} \\ {\scriptstyle \text{vec}}\downarrow & & \downarrow{\scriptstyle \text{vec}} \\ u & \xrightarrow[Q_p]{} & \tilde{u} \end{array}$$

Ce qui signifie,

$$\text{vec}(S_p(I)) = Q_p(\text{vec}(I)). \tag{4.5}$$

Pour trouver l'expression de Q_p, il suffit vectoriser l'expression (4.4),

$$\begin{aligned} \tilde{u} &= \text{vec}(S_p(I)) \\ &= \text{vec}(D_N^i I D_M^{-j}) \\ &= (D_M^j \otimes D_N^i)\,\text{vec}(I). \end{aligned}$$

Nous avons alors nécessairement,

$$Q_p = D_M^j \otimes D_N^i \quad \text{avec} \quad p = (i, j).$$

La matrice Q_p est orthogonale puisque D_M et D_N le sont. D'autre part, nous pouvons montrer facilement que $Q_{-e} = Q_e^{-1}$. Nous avons ainsi les propriétés suivantes,

$$Q_{-e} = Q_e^{-1}, \tag{4.6}$$
$$Q_e^{-1} = Q_e^T. \tag{4.7}$$

Les opérateurs de décalages ainsi que les différentes propriétés établies nous permettent à présent de définir la *spectralisation* d'une image monochrome.

4.1.4 Spectralisation d'une image en niveaux de gris

Nous définissons dans un premier temps, la *spectralisation* d'une image monochrome I selon un motif P.

Définition 4.1. Soit I une image en niveaux de gris et P une représentation normalisée d'un motif. Nous posons $P = (p_1, \ldots, p_{|P|})$. La spectralisation de l'image I selon P, notée $\mathrm{Spec}_P(I)$ est la matrice tridimensionnelle obtenue en empilant les matrices $S_{-p_1}(I), \ldots, S_{-p_{|P|}}(I)$,

$$\mathrm{Spec}_P(I) = S_{-p_1}(I) \wedge \cdots \wedge S_{-p_{|P|}}(I).$$

De la même manière, lorsque nous manipulons des données linéarisées, la spectralisation d'un vecteur u est définie par,

$$\mathrm{Spec}_P(u) = \left[Q_{-p_1}(u), \ldots, Q_{-p_{|P|}}(u)\right]. \tag{4.8}$$

Par exemple, en reprenant l'exemple (4.3), nous choisissons le motif (4.1) décrit par le motif $((0,0),(1,0),(1,1))$. Nous empilons donc les matrices,

$$S_{(0,0)}(I) = I = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad S_{-(1,0)}(I) = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{pmatrix}, \quad S_{-(1,1)}(I) = \begin{pmatrix} 5 & 6 & 4 \\ 8 & 9 & 7 \\ 2 & 3 & 1 \end{pmatrix}.$$

Nous obtenons un volume tridimensionnel représenté à la figure 4.2b.

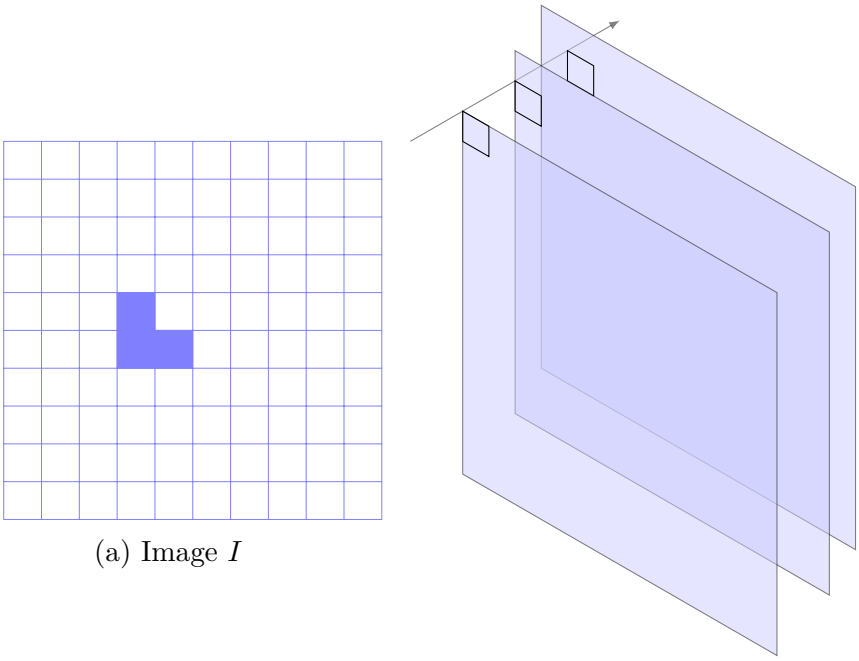
Nous représentons ensuite l'image *spectralisée* sous sa forme vectorisée : chacune des bandes est mise en ligne. Nous retrouvons la modélisation habituelle d'une image multispectrale.

$$\mathrm{Spec}_P(u) = \begin{pmatrix} 1 & 4 & 5 \\ 4 & 7 & 8 \\ 7 & 1 & 2 \\ \mathbf{2} & \mathbf{5} & \mathbf{6} \\ 5 & 8 & 9 \\ 8 & 2 & 3 \\ 3 & 6 & 4 \\ 6 & 9 & 7 \\ 9 & 3 & 1 \end{pmatrix}. \tag{4.9}$$

La figure 4.3 montre le résultat de la spectralisation d'un motif plus compliqué.

L'image *spectralisée* est donc l'image de départ à laquelle nous avons adjoint d'autres bandes. Le nombre de bandes de l'image *spectralisée* est calculé par la relation suivante.

$$n_B\left(\mathrm{Spec}_P(u)\right) = |P|. \tag{4.10}$$



(a) Image I

(b) Image spectralisée tridimensionnelle

Figure 4.2 – Image en niveaux de gris I et sa spectralisation selon le motif en crochet sous forme tridimensionnelle.

La propriété la plus intéressante de l'image spectralisée est qu'elle est construite de manière à ce que le spectre en un pixel donné soit la mise en ligne des pixels que décrit le motif sur l'image d'origine au pixel considéré.

En effet, en reprenant l'exemple précédent, si nous voulons détecter la forme,

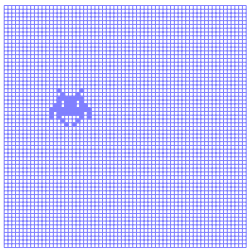
$$\begin{pmatrix} 2 & \star \\ 5 & 6 \end{pmatrix},$$

décrite par le motif P, il nous suffit de rechercher la signature $(2, 5, 6)$ dans l'image spectralisée (4.9).

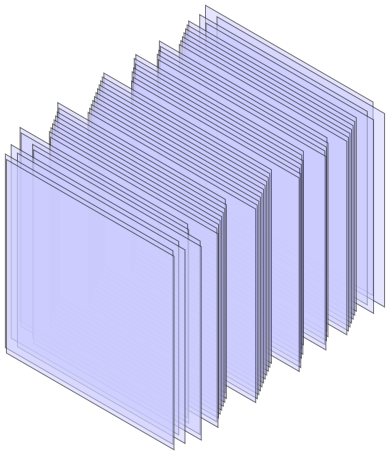
4.1.5 Spectralisation d'une image multispectrale

La définition (4.8) se généralise sans mal pour le cas d'une image multispectrale X. Nous supposons que l'image multispectrale X est donnée sous forme linéarisée. Nous définissons alors,

$$\mathrm{Spec}_P(X) = \left[Q_{-p_1}X, \ldots, Q_{-p_{|P|}}X\right]. \tag{4.11}$$



(a) Image I



(b) Image spectralisée tridimensionnelle

Figure 4.3 – Image en niveaux de gris I et sa spectralisation selon le motif « invader » sous forme tridimensionnelle.

La relation (4.10) devient,

$$n_B\left(\mathrm{Spec}_P(X)\right) = n_B(X) \cdot |P| \,.$$

4.2 Mesures sur une image spectralisée

Nous pouvons désormais détecter un motif dans une image multispectrale en détectant une signature dans une image plus complexe. Cependant, dans le cadre de l'acquisition comprimée nous ne disposons pas de l'image d'origine et donc encore moins de l'image spectralisée. Nous disposons seulement de mesures sur l'image d'origine. Pour appliquer un algorithme de recherche de signature, il nous faut d'abord disposer de mesures sur l'image spectralisée. En somme, nous avons une image multispectrale X et des mesures M_{eff} prises sur elle, à l'aide d'une matrice de mesures F_{eff}, c'est à dire,

$$M_{\text{eff}} = F_{\text{eff}} X$$

L'indice « eff » se rapporte au mot *effectif* qui rappelle que ce sont des mesures réelles, prises physiquement sur l'image réelle sans être reconstruites. À l'inverse, l'image $\mathrm{Spec}_P(X)$ est une image fictive et les mesures prises sur elles ne seront donc pas réelles mais reconstruites.

$$M_{\text{virt}} = F_{\text{virt}}\,\mathrm{Spec}_P(X)$$

Dans cette section, nous allons étudier s'il est possible de reconstruire les mesures M_{virt} sur l'image spectralisée à partir de mesures M_{eff} sur l'image d'origine.

Dans le cas où une telle reconstruction est possible, il faudra veiller à ce que la reconstruction de mesures virtuelles ne soit pas trop coûteuses en termes de mesures réelles. Nous allons donc nous intéresser au rapport suivant qui mesure l'efficacité de la reconstruction de mesures.

$$\alpha(P) = \frac{\text{mesures effectives}}{\text{mesures virtuelles}}. \tag{4.12}$$

Les mesures effectives désignent les mesures qui seront prises sur l'image d'origine. Les mesures virtuelles sont celles que nous allons pouvoir reconstruire à partir des mesures effectives et qui sont des mesures sur l'image spectralisée. Une mesure virtuelle nécessite au moins une mesure effective, nous avons donc toujours,

$$\alpha(P) \geqslant 1.$$

Le but est d'obtenir une valeur de $\alpha(P)$ qui est la plus proche de 1 possible. En effet, plus $\alpha(P)$ est proche de 1, moins nous avons besoin de mesures effectives supplémentaires pour reconstruire un nombre donné de mesures virtuelles. Nous allons d'abord nous placer dans le cas où nous prenons des mesures quelconques pour fournir une première borne supérieure. Par la suite, nous allons voir comment nous pouvons choisir les mesures afin de diminuer sensiblement le coefficient α.

4.2.1 Reconstruction de mesures naïves

Nous nous plaçons dans le cas d'une image en niveaux de gris. La matrice X est donc un vecteur colonne. Si nous souhaitons reconstruire une mesure virtuelle faite sur $\mathrm{Spec}_P(X)$ avec le vecteur mesure f, d'après la définition (4.8), nous avons,

$$\begin{aligned} f^T \mathrm{Spec}_P(X) &= f^T \left[Q_{-p_1} X, \ldots, Q_{-p_{|P|}} X \right] \\ &= \left[f^T X, f^T Q_{-p_2} X, \ldots, f^T Q_{-p_{|P|}} X \right]. \end{aligned}$$

Il nous faut donc au plus $|P|$ mesures différentes sur X, de vecteur mesure,

$$\begin{aligned} f_1 &= Q_{p_1} f, \\ f_2 &= Q_{p_2} f, \\ &\vdots \\ f_{|P|} &= Q_{p_{|P|}} f, \end{aligned} \tag{4.13}$$

pour reconstruire une seule mesure virtuelle. Nous avons ainsi,

$$\alpha(P) \leqslant |P|.$$

Cette borne supérieure est cependant très mauvaise. En effet, en prenant l'exemple de la détection du motif (4.1), nous avons $|P| = 3$. Il faudrait alors faire trois

fois plus de mesures pour reconstruire les mesures virtuelles. Dans le cas où nous voudrions un taux de mesures de 30% pour l'image spectralisée, il faudrait alors mesurer à hauteur de 90% sur l'image d'origine sur qui contredit le principe de l'acquisition comprimée.

4.2.2 Reconstruction de mesures glissées

La méthode précédente n'est pas satisfaisante car chaque mesure virtuelle nécessite $|P|$ mesures effectives distinctes et toutes ces mesures effectives ne sont pas redondantes. Il faut choisir des mesures virtuelles qui partagent un nombre important de mesures effectives pour être reconstruite.

Nous pouvons utiliser le fait que les bandes de l'image spectralisée sont très redondantes puisqu'elles sont déductibles les unes des autres par simple translation. En translatant de la même manière les mesures, il est possible d'introduire de la redondance dans les mesures virtuelles : une mesure effective servant à la reconstruction d'une mesure virtuelle sert également à la reconstruction de nombreuses autres mesures virtuelles. Nous pouvons ainsi faire chuter la borne $|P|$ du coefficient α.

Nous allons donc supposer que toutes les mesures virtuelles faites sur l'image spectralisée $\mathrm{Spec}_P(X)$ sont les mêmes à translation près. En fixant l'une de ces mesures virtuelles f, nous pouvons décrire l'ensemble des mesures par un motif $E = (e_1, \ldots, e_{|E|})$ qui décrit les glissements à effectuer pour avoir les autres mesures virtuelles. Les mesures virtuelles sont alors,

$$\begin{aligned} f_1 &= Q_{e_1} f, \\ f_2 &= Q_{e_2} f, \\ &\vdots \\ f_{|E|} &= Q_{e_{|E|}} f. \end{aligned} \tag{4.14}$$

Et le résultat de ces mesures vaut,

$$\begin{aligned} m_1 &= (Q_{e_1} f)^T \,\mathrm{Spec}_P\,(X)\,, \\ m_2 &= (Q_{e_2} f)^T \,\mathrm{Spec}_P\,(X)\,, \\ &\vdots \\ m_{|E|} &= (Q_{e_{|E|}} f)^T \,\mathrm{Spec}_P\,(X)\,. \end{aligned} \tag{4.15}$$

En notant,

$$M_{\text{virt}} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{|E|} \end{pmatrix} \quad \text{et} \quad F_{\text{virt}} = \begin{pmatrix} f_1^T \\ f_2^T \\ \vdots \\ f_{|E|}^T \end{pmatrix}, \tag{4.16}$$

nous pouvons reformuler les différentes mesures (4.15) en notation matricielle,

$$M_{\text{virt}} = F_{\text{virt}} \operatorname{Spec}_P(X). \tag{4.17}$$

Pour les besoins de l'algorithme de détection, il nous faut reconstruire la matrice M_{virt} car elle est le résultat de mesures prises sur l'image spectralisée $\operatorname{Spec}_P(X)$. Cependant, nous ne disposons que de mesures sur l'image d'origine X. En détaillant un peu plus l'expression (4.17) à l'aide de (4.11) et (4.16), nous obtenons,

$$M_{\text{virt}} = \begin{pmatrix} f_1^T \\ f_2^T \\ \vdots \\ f_{|E|}^T \end{pmatrix} \times \left[Q_{-p_1} X, \ldots, Q_{-p_{|P|}} X \right], \tag{4.18}$$

puis, en notation condensée et à l'aide de (4.14), (4.6) et (4.7), nous trouvons,

$$M_{\text{virt}} = \left(f^T Q_{-e_i} Q_{-p_j} X \right)_{\substack{1 \leqslant i \leqslant |E| \\ 1 \leqslant j \leqslant |P|}} \tag{4.19}$$

$$= \left((Q_{e_i + p_j} f)^T X \right)_{\substack{1 \leqslant i \leqslant |E| \\ 1 \leqslant j \leqslant |P|}}. \tag{4.20}$$

La relation (4.20) nous dit que pour calculer toutes les mesures virtuelles, il suffit de faire des mesures glissées sur l'image d'origine X. Ces mesures glissées sont représentées par le motif de mesure $E + P$. Nous avons donc besoin de $|E + P|$ mesures effectives. Le coefficient de reconstruction α défini par la relation (4.12) vaut alors,

$$\alpha(P) = \frac{|E + P|}{|E|}. \tag{4.21}$$

Nous avons donc une expression du coefficient de reconstruction en fonction de deux motifs ce qui va nous permettre de l'optimiser dans le cas de mesures glissées.

4.2.3 Motif de mesures glissées optimal

Afin d'optimiser le coefficient de reconstruction (4.21) il faut trouver les mesures virtuelles telles que pour un motif fixé P, $|E + P|$ soit minimum. Nous avons d'abord la proposition simple suivante qui montre que plus nous prenons de mesures, mieux c'est.

Proposition 4.1. *Pour un motif fixé P, nous avons,*

$$\inf_E \frac{|E + P|}{|E|} = 1.$$

Démonstration. Nous savons déjà que $|E+P| \geqslant |E|$ et donc $\frac{|E+P|}{|E|} \geqslant 1$. Soit R un motif rectangulaire ayant a lignes et b colonnes contenant le motif P. Pour $n \geqslant 1$, soit E un motif rectangulaire ayant na lignes et nb colonnes. Il est facile de voir que $E+R$ est un motif rectangulaire ayant $(n+1)a-1$ lignes et $(n+1)b-1$ colonnes. Nous avons alors,

$$\frac{|E+R|}{|E|} = \frac{((n+1)a-1)\,((n+1)b-1))}{n^2ab} \underset{n\to+\infty}{\longrightarrow} 1.$$

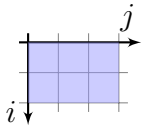
■

Pour que le coefficient de reconstruction tende vers 1, il faut donc prendre un grand nombre de mesures. Cependant, le nombre de mesures est justement limité dans le cadre de l'acquisition compressée. Nous allons donc nous intéresser au problème suivant,

$$\inf_{|E|=N} \frac{|E+P|}{|E|}.$$

C'est un problème de combinatoire que nous ne pouvons pas résoudre de manière générale.

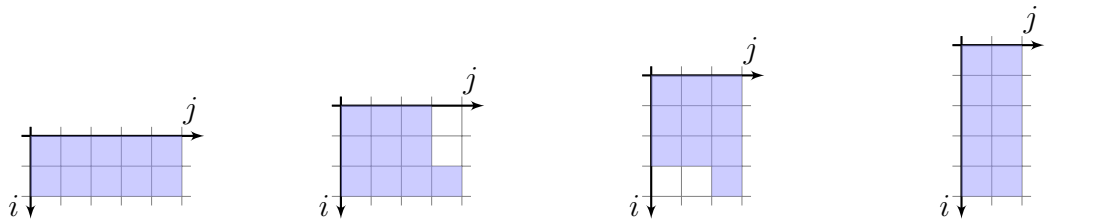
Pour simplifier le problème, nous faisons l'hypothèse que le motif P est rectangulaire. Nous allons détailler le résultat de la minimisation du coefficient de reconstruction pour l'exemple suivant. Supposons que P est un motif rectangulaire à deux lignes et trois colonnes,



et que N vaut 10. Différents motifs de mesures E représentés à la figure 4.4 satisfont tous $|E| = 10$ et semblent minimiser $|E+P|$. D'après la figure 4.5, qui représente les motifs $E+P$ correspondants, la disposition 4.4a semble être optimale. Cependant, si nous rajoutons une mesure passant ainsi à $N = 11$, c'est le motif 4.4b en ajoutant la case $(1,3)$ qui devient optimal.

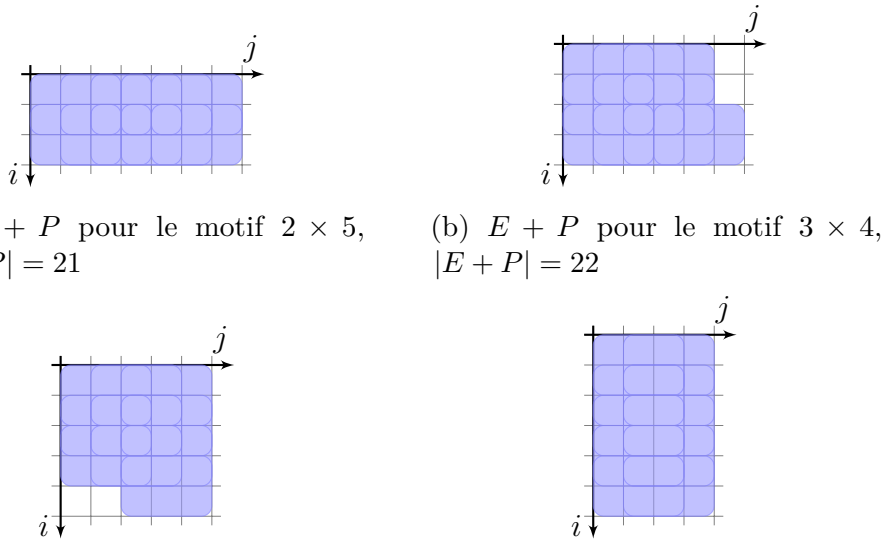
Nous pouvons reformuler ce problème d'une manière plus visuelle. Supposons que nous disposons d'une feuille quadrillée et d'un tampon rectangulaire de taille 2×3 par rapport au quadrillage de cette feuille. Supposons que nous devons appliquer 10 coups de tampon distincts. Comment disposer ces coups de tampons de manière à ce que la tâche résultante ait la surface la plus réduite possible ?

Hormis le fait qu'il faille évidemment concentrer autant que possible le motif de mesure E pour minimiser $|E+P|$, il semble que la disposition est optimale lorsque le motif de mesures E à la même forme que le motif P. Comme le motif E ne peut pas toujours être rectangulaire, nous définissons une forme *quasi-rectangulaire* de



(a) Motifs 2×5 (b) Motifs 3×4 (c) Motif 4×3 (d) Motif 5×2

Figure 4.4 – Quelques motifs de mesures E de cardinal 10



(a) $E + P$ pour le motif 2×5, $|E + P| = 21$

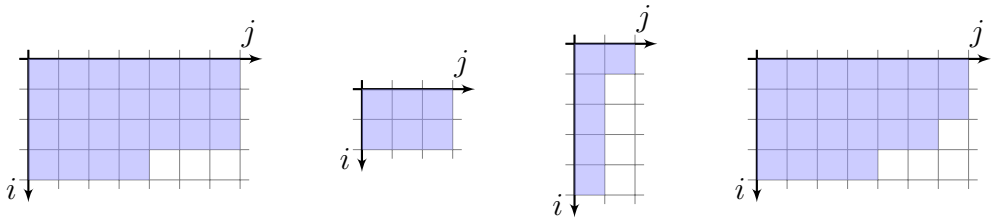
(b) $E + P$ pour le motif 3×4, $|E + P| = 22$

(c) $E + P$ pour le motif 4×3, $|E + P| = 23$

(d) $E + P$ pour le motif 5×2, $|E + P| = 24$

Figure 4.5 – Représentation de $E + P$ pour les motifs de mesures de la figure 4.4

la manière suivante. Un motif quasi-rectangulaire est un motif rectangulaire auquel nous adjoignons une ligne ou une colonne qui est incomplète d'un côté seulement. Les trois premiers motifs suivants sont quasi-rectangulaire, le dernier ne l'est pas,



Nous pouvons alors formuler la proposition suivante qui contrôle le motif de mesures que nous prendrons lorsque nous aurons à détecter un motif rectangulaire.

Proposition 4.2. *Soit P un motif rectangulaire. Alors, un motif de mesure E tel que $|E| = A$ minimisant $\frac{|E+P|}{|E|}$ est obtenu lorsque E est quasi-rectangulaire et que sa hauteur s minimise la fonctionnelle suivante,*

$$(a-1)\left\lceil \frac{A}{s} \right\rceil + (b-1)s,$$

où a et b sont respectivement la hauteur et la largeur de P.

Démonstration. Sans perte de généralité, nous pouvons supposer que,

$$P = \{(i,j),\, 0 \leqslant i < a,\, 0 \leqslant j < b\}.$$

Soit E un motif de mesures tel que $|E| = A$ et soient n_i et m_i respectivement le nombre d'éléments de $E + P$ et E d'ordonnée i. Nous avons alors,

$$\sum_{i\in\mathbb{Z}} n_i = |E + P|\,, \tag{4.22}$$

et

$$\sum_{i\in\mathbb{Z}} m_i = |E|\,. \tag{4.23}$$

Nous remarquons également que pour tout $i \in \mathbb{Z}$ et $j \in \{i, i-1, \ldots, i-a+1\}$ nous avons,

$$\begin{cases} m_j = 0 & \text{si } n_i = 0, \\ n_i \geqslant m_j + b - 1 & \text{si } n_i \neq 0. \end{cases}$$

Si nous définissons,

$$(\phi_a(u))_i := \max_{k\in\{i,i-1,\ldots,i-a+1\}} u_k, \tag{4.24}$$

nous pouvons alors écrire,

$$\begin{cases} (\phi_a(m))_i = 0 & \text{if } n_i = 0, \\ n_i \geqslant (\phi_a(m))_i + b - 1 & \text{if } n_i \neq 0. \end{cases}$$

En sommant tous les n_i pour $i \in \mathbb{Z}$, nous avons,

$$|E + P| \geqslant \sum_{i\in\mathbb{Z}} (\phi_a(m))_i + \sum_{\substack{i\in\mathbb{Z} \\ n_i\neq 0}} b - 1 \tag{4.25}$$

$$\geqslant \sum_{i\in\mathbb{Z}} (\phi_a(m))_i + (|m| + a - 1)(b - 1). \tag{4.26}$$

En prenant la borne inférieure des membres de droite pour tous les motifs de mesures E tel que $|E| = A$, nous avons,

$$|E + P| \geqslant \inf_{s\geqslant 1} \left[\inf_{|m|=s} \sum_{i\in\mathbb{Z}} (\phi_a(m))_i + (s + a - 1)(b - 1) \right]. \tag{4.27}$$

Nous utilisons ensuite le lemme 4.3 qui dit que,

$$\inf_{|m|=s} \sum_{i\in\mathbb{Z}} (\phi_a(m))_i = A + (a - 1) \left\lceil \frac{A}{s} \right\rceil, \tag{4.28}$$

puis, en remplaçant dans (4.27), nous trouvons,

$$|E + P| \geqslant A + (a - 1)(b - 1) + \inf_{s\geqslant 1} (a - 1) \left\lceil \frac{A}{s} \right\rceil + (b - 1)s. \tag{4.29}$$

■

Lemme 4.3. *Soit a un entier strictement positif, u une suite d'entiers positifs indexée par $\mathbb{Z}$ telle que $\sum_{i\in\mathbb{Z}} u_i = A \geqslant s$ et ϕ définie par l'égalité* (4.24)*. Nous avons,*

$$\inf_{|u|=s} \sum_{i\in\mathbb{Z}} (\phi_a(u))_i = A + (a - 1) \left\lceil \frac{A}{s} \right\rceil, \tag{4.30}$$

et une suite minimisante est,

$$v_i = \begin{cases} q + 1 & si\ 0 \leqslant i < r, \\ q & si\ r \leqslant i < s, \\ 0 & sinon, \end{cases}$$

où q et r sont respectivement le quotient et le reste de la division euclidienne de A par s.

Démonstration. Nous pouvons montrer que v vérifie,

$$\sum_{i\in\mathbb{Z}} v_i = A \quad \text{et} \quad \sum_{i\in\mathbb{Z}} (\phi_a(v))_i = A + (a-1)\left\lceil \frac{A}{s} \right\rceil.$$

Il suffit alors de montrer que,

$$\sum_{i\in\mathbb{Z}} (\phi_a(u))_i \geqslant \sum_{i\in\mathbb{Z}} (\phi_a(v))_i,$$

pour tout u tel que $|u| = m$ et $\sum_{i\in\mathbb{Z}} u_i = A \geqslant s$. Sans perte de généralité, nous pouvons supposer que le support de u est un sous-ensemble contiguë de $\mathbb{Z}$ et que ce sous-ensemble est $[\![0, s-1]\!]$. Le problème se divise ensuite en deux cas :

– Si pour tout $i, j \in [\![0, s-1]\!], |u_i - u_j| \leqslant 1$, nous avons,

$$\sum_{i\in\mathbb{Z}} (\phi_a(u))_i \geqslant \sum_{i\in\mathbb{Z}} (\phi_a(v))_i. \tag{4.31}$$

– Sinon, soit $i_0 \in [\![0, s-1]\!]$ un indice tel que $u_{i_0} \geqslant 2 + u_i$ pour tout $i \neq i_0$. Nous choisissons un autre indice j_0 distinct de i_0 et nous définissons une autre suite u' correspondant au motif de mesures E' comme suit,

$$u'_i = \begin{cases} u_i - 1 & \text{if } i = i_0, \\ u_i + 1 & \text{if } i = j_0, \\ u_i & \text{sinon.} \end{cases} \tag{4.32}$$

Nous remarquons alors que,

$$\sum_{i\in\mathbb{Z}} (\phi_a(u))_i \geqslant \sum_{i\in\mathbb{Z}} (\phi_a(u'))_i.$$

Par récurrence sur u_{i_0}, nous sommes réduit au cas précédent.
Ainsi, pour tout u tel que $|u| = s$, nous avons

$$\sum_{i\in\mathbb{Z}} (\phi_a(u))_i \geqslant A + (a-1)\left\lceil \frac{A}{s} \right\rceil.$$

■

Nous avons à présent des mesures efficaces, les mesures glissées, qui permettent de réduire considérablement le coefficient de reconstruction α et, dans ce cadre, un motif de mesures optimal pour des motifs à détecter rectangulaire. Pour des motifs non rectangulaires, nous prendrons le motif de mesures obtenu pour le plus petit rectangle contenant le motif non rectangulaire qui donne des résultats satisfaisant lorsque le motif est concentré.

4.3 Mise en œuvre et illustrations

4.3.1 Mise en œuvre

La détection de motifs dans l'image multispectrale X repose donc sur une détection de signatures dans l'image spectralisée $\mathrm{Spec}_P(X)$. Nous avons ainsi la minimisation suivante,

$$\arg\min_{u\geqslant 0} \|u\|_1 \quad \text{tel que} \quad \begin{cases} \mathrm{Spec}_P(X)^T u = s \\ M_{\text{virt}} = F_{\text{virt}}\,\mathrm{Spec}_P(X). \end{cases} \tag{4.33}$$

Nous éliminons ensuite $\mathrm{Spec}_P(X)$ des deux équations, en supposant,

$$F_{\text{virt}}^T (F_{\text{virt}} F_{\text{virt}}^T)^{-1} F_{\text{virt}} \approx \frac{m}{n_P} I_{n_P}. \tag{4.34}$$

La condition se résume alors à,

$$M_{\text{virt}}^T (F_{\text{virt}} F_{\text{virt}}^T)^{-1} F_{\text{virt}} u \approx \frac{m}{n_P} s,$$

et la minimisation (4.33) devient ainsi,

$$\arg\min_{u\geqslant 0} \|u\|_1 \quad \text{tel que} \quad M_{\text{virt}}^T (F_{\text{virt}} F_{\text{virt}}^T)^{-1} F_{\text{virt}} u = \frac{m}{n_P} s. \tag{4.35}$$

Dans cette contrainte, nous connaissons F_{virt} qui est obtenue à partir de la matrice de mesures effectives F_{eff} donnée par le motif $E + P$. De même, nous avons choisi les mesures effectives pour être capable de reconstruire M_{virt}. Cependant, par souci de simplicité, nous allons dans un premier temps plutôt la calculer directement par la relation,

$$M_{\text{virt}} = F_{\text{virt}}\,\mathrm{Spec}_P(X),$$

alors que nous n'avons théoriquement pas accès à $\mathrm{Spec}_P(X)$ ni à X. Nous disposons néanmoins du coefficient de reconstruction qui nous indique le nombre de mesures supplémentaires à effectuer. Ce choix va limiter la taille du problème car la matrice $\mathrm{Spec}_P(X)$ devient très grande.

Nous choisissons également des mesures normalement distribuées qui sont de « bonnes » matrices de mesures et donnent généralement de bons résultats. Cela limite également la taille des problèmes envisagés.

Pour ces deux raisons, nous nous limitons à des tailles d'images de l'ordre de 100×100 et d'une dizaine de bandes. D'autre part, comme nous n'avons à résoudre que des minimisations faisant intervenir une norme L_1, nous utilisons la *toolbox* YALL1 [Zha09, YZ11] qui va nous permettre d'augmenter un peu les dimensions

du problème. Nous utiliserons le modèle de minimisation L_1 non contraint de la *toolbox* YALL1.

$$\arg\min_{u\geqslant 0} \|u\|_1 + \frac{1}{2\rho}\left\|M_{\text{virt}}^T(F_{\text{virt}}F_{\text{virt}}^T)^{-1}F_{\text{virt}}u - \frac{m}{n_P}s\right\|_2^2. \tag{4.36}$$

La tolérance est fixée à $5\cdot 10^{-4}$ et le coefficient ρ est fixé à 10^{-3}.

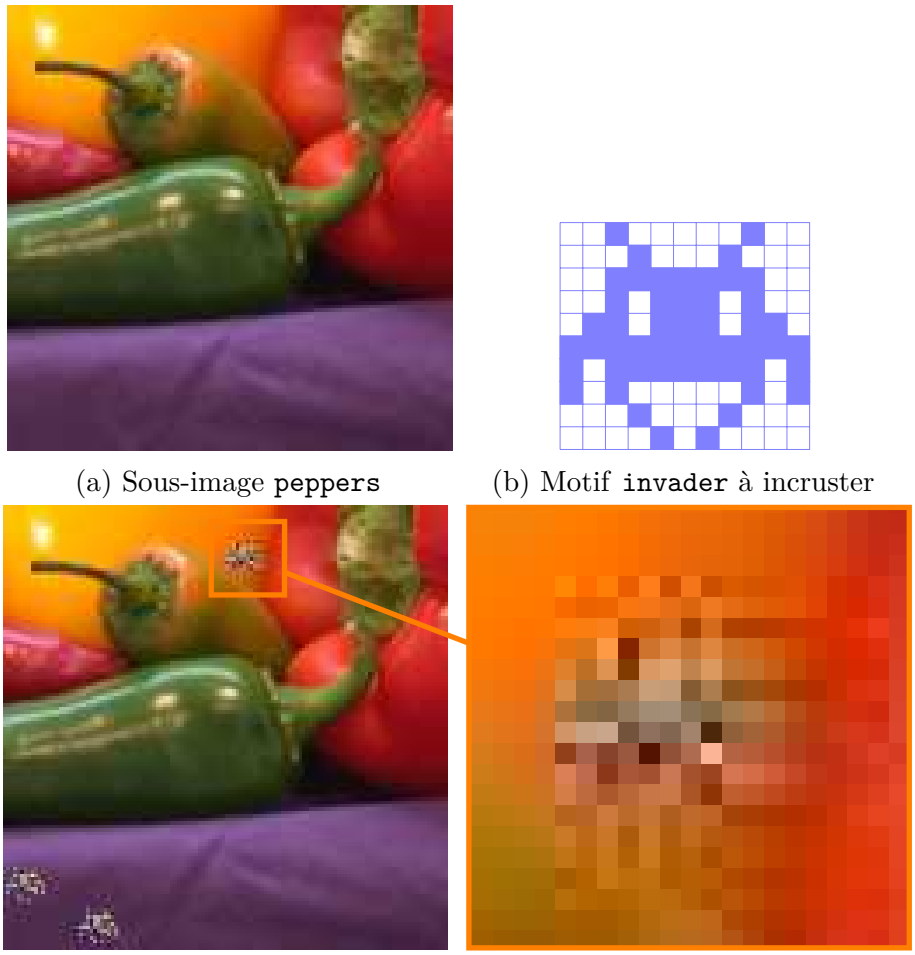
Nous proposons maintenant d'appliquer notre méthode sur différents types d'images.

4.3.2 Tests sur l'image `peppers`

Dans un premier temps, nous menons des tests sur une sous-image de taille 128×128 de l'image couleur classique `peppers` dans laquelle nous avons incrusté trois motifs à détecter. L'image d'origine, la structure du motif d'incrustation, l'image incrustée et un zoom sur un des motifs incrustés sont donnés à la figure 4.6. La structure du motif d'incrustation compte 51 éléments. Nous cherchons donc une signature de longueur 153 dans l'image spectralisée qui est donc de taille 128×128 et compte 153 bandes. Le motif d'incrustation est généré aléatoirement.

Les expérimentations sont menées pour différents taux de mesures allant de 1% à 30%. Le motif à détecter étant de taille 10×11, nous supposons que c'est un motif quasi-rectangulaire et nous choisissons donc un motif de mesures selon la proposition 4.2. Le tableau 4.1 résume l'efficacité de la reconstruction mesurée par le coefficient α, le temps de calcul et les fausses détections en fonction du taux de mesures virtuelles choisi. Pour ces deux dernières colonnes, une moyenne est effectuée pour 10 matrices de mesures différentes. Nous remarquons que plus le taux de mesures est élevé plus le taux de reconstruction baisse ce qui est naturel d'après la propriété 4.1. Le temps d'exécution augmente lorsque qu'il y a moins de mesures. Le temps de calcul accru est la contrepartie d'une compression plus grande. À noter l'augmentation du temps de calcul pour un taux de mesures virtuelles de 30% qui est le fait d'une unique minimisation parmi les dix.

Même à un taux de mesures effectives d'environ 6%, la détection est parfaite comme le montre le résultat de la reconstruction de motifs à la figure 4.7.



(a) Sous-image `peppers` (b) Motif `invader` à incruster

(c) Sous-image de `peppers` incrustée (d) Zoom sur le motif

Figure 4.6 – Image couleur issue de `peppers` avec trois motifs incrustés

Mesures virtuelles (%)	Mesures effectives (%)	Efficacité de la reconstruction α	Temps d'exécution (s)	Erreurs
1	2.83	2.83	15.14	2
2	4.46	2.23	12.67	0.2
3	5.93	1.98	11.45	0
4	7.33	1.83	12.72	0
5	8.68	1.74	11.04	0
10	15.06	1.51	10.87	0
15	21.11	1.41	10.67	0
20	26.97	1.35	9.46	0
25	32.78	1.31	9.77	0
30	38.47	1.28	14.57	0

Table 4.1 – Statistiques de l'algorithme de détection compressée de motifs pour différents taux de mesures moyennées sur 10 instances du même problème



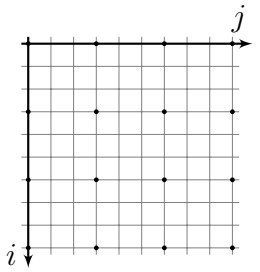
(a) Détection de motifs, 3% de données

(b) Détection de motifs, 30% de données

Figure 4.7 – Détection de motifs pour différents taux de mesures

4.3.3 Tests sur l'image réelle `Bâche`

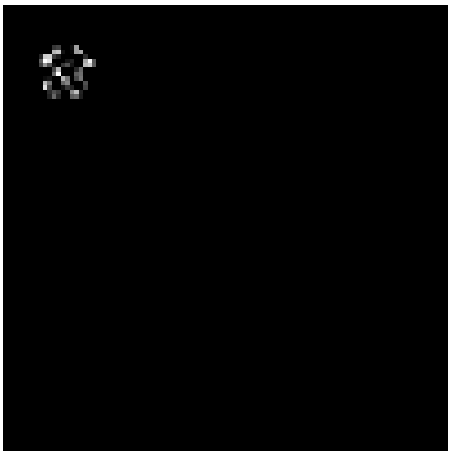
Nous testons ensuite l'algorithme de détection compressée de motifs sur une image multispectrale réelle issue du projet HYPLITT d'ACTIMAR [SSL, JGBT12]. Nous extrayons une image de taille 100×100 avec 12 bandes. Nous nous proposons de détecter la mire présente au milieu de l'image avec sa structure en damier. Le motif que nous recherchons a la structure suivante,



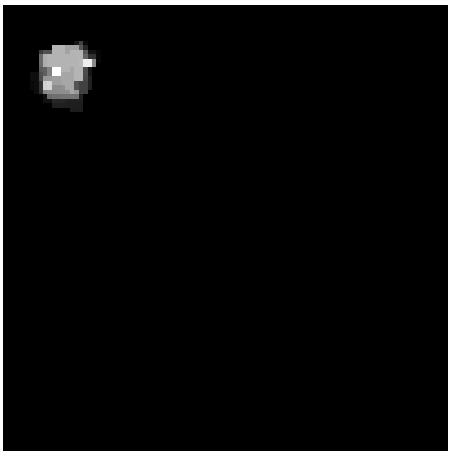
Le motif est carré mais éclaté. Nous choisissons quand même de déterminer le motif de mesures par la proposition 4.2. Pour un meilleur coefficient de reconstruction, il faudrait de la même façon éclater le motif de mesures. La figure 4.8 montre le résultat des différentes méthodes envisagées jusqu'alors. Comme le motif en damier à détecter comporte la même signature spectrale en chaque point du motif, nous pouvons utiliser directement les algorithmes de détection de signatures. La figure 4.8b montre la détection par la minimisation de Guo et Osher qui arrive correctement à détecter la structure en damier. La figure 4.8c montre la détection compressée de signature avec 30% des données. La structure en damier n'est plus détectée. L'utilisation d'une régularisation TV/L_1 s'accorde mal avec la détection d'une zone texturée. Nous pouvons seulement affirmer que la zone détectée contient des pixels ayant la signature recherchée. En revanche, la détection compressée de motifs ne souffre pas de cet inconvénient. Le motif est contenu dans la signature de l'image spectralisée. Donc toute détection de signature est en fait la détection d'un motif. Nous observons donc à la figure 4.8d que la reconstruction comporte un pixel à l'endroit du motif.



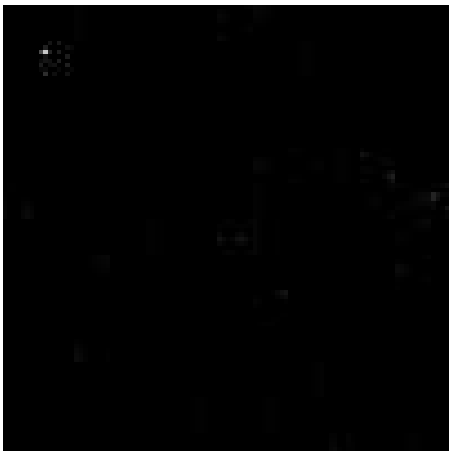
(a) Image originale en fausses couleurs



(b) Détection de signature classique



(c) Détection compressée de signature, 30% des données, régularisation TV/L_1



(d) Détection compressée de motif, 30% des données

Figure 4.8 – Détection compressée de motifs

4.4 Conclusion

Dans ce chapitre, nous avons proposé une extension de la minimisation établie au chapitre précédent pour réaliser une détection compressée de motifs dans une image multispectrale. Cette extension est rendue possible par une opération fictive de *spectralisation* qui place les informations du motif dans les bandes d'une image qui en comporte beaucoup plus. Ce nouveau point de vue pose le problème de la reconstruction de mesures sur l'image spectralisée. Nous proposons l'utilisation de mesures glissées sur l'image d'origine qui permettent une reconstruction efficace. Dans le cadre des mesures glissées, lorsque le motif à détecter est rectangulaire, nous démontrons quels motifs de mesures utiliser pour une reconstruction optimale. Enfin, nous proposons une application de la minimisation établie au travers d'exemples sur des images couleur et multispectrale. La mise en œuvre pose cependant plusieurs problèmes.

Le choix de calculer les mesures virtuelles directement comme si nous disposions de l'image $\mathrm{Spec}_P(X)$ limite les dimensions de l'image testée. Par exemple, si nous voulons réaliser une détection d'un motif P tel que $|P| = 10$ sur une simple image de taille 100×100 qui compte 20 bandes, alors $\mathrm{Spec}_P(X)$ est de taille 10000×200 et F_{virt} est de taille 3000×10000. La dimension de ces deux matrices empêche de tester l'algorithme sur une image beaucoup plus grande. Pour augmenter en dimension, il faut se passer de la matrice $\mathrm{Spec}_P(X)$. D'autre part, un problème plus fondamental se pose lorsque nous sommes amenés à calculer la matrice $(F_{\mathrm{virt}}F_{\mathrm{virt}}^T)^{-1}$. En effet, en reprenant l'exemple précédent, nous trouvons que la matrice $F_{\mathrm{virt}}F_{\mathrm{virt}}^T$ est de taille 3000×3000 et qu'il faut l'inverser. Pour monter en dimension, il faut trouver un moyen de calculer rapidement l'inverse de la matrice $F_{\mathrm{virt}}F_{\mathrm{virt}}^T$. C'est l'objet du chapitre suivant qui traite des matrices partiellement orthogonales.

Chapitre 5

Matrices circulantes partiellement orthogonales

Sommaire

5.1 Quelques définitions 90
5.1.1 Matrices circulantes 90
5.1.2 Matrices de Fourier 91
5.1.3 Opérateurs • et † 92
5.2 Matrices circulantes orthogonales 93
5.3 Matrices circulantes partiellement orthogonales 94
5.3.1 Analyse et synthèse 94
5.3.2 Exemple pratique 97
5.3.3 Exemple de « somme nulle » 99
5.4 Matrices circulantes par blocs, à blocs circulants partiellement orthogonales 100
5.4.1 Analyse et synthèse 100
5.4.2 Exemple pratique 105
5.5 Mise en œuvre 105
5.6 Applications 106
5.6.1 Sur l'image `peppers` 107
5.6.2 Sur l'image `bâche` 108
5.7 Conclusion 112

DANS le chapitre précédent, nous sommes limités en termes de dimension par le calcul de l'expression $(F_{\text{virt}}F_{\text{virt}}^T)^{-1}$. Même s'il n'est besoin de le faire qu'une seule fois par détection, en augmentant la taille des images, l'inversion peut s'avérer impossible. L'idéal serait d'obtenir une méthode de génération telle que $F_{\text{virt}}F_{\text{virt}}^T$ soit

la matrice identité. Nous savons déjà que la matrice F_{virt} a une structure particulière car pour optimiser la reconstruction des mesures virtuelles, nous prenons des mesures glissées de l'image multispectrale. En fait, la matrice F_{virt} peut être vue comme un sous-ensemble de lignes d'une matrice circulante par blocs, à blocs circulants. Ces matrices ont de nombreuses propriétés. En effet, d'une part, elles sont définies entièrement par leur première ligne d'où un gain de mémoire. D'autre part, de nombreuses opérations se traduisent par une opération sur leur première ligne d'où un gain en temps de calcul. L'acquisition avec des matrices de mesures issues de matrices circulantes par blocs, à blocs circulants est également beaucoup plus facile d'un point de vue physique. Cela revient à effectuer une convolution de l'image avec un masque fixe.

Les matrices circulantes sont également très intéressantes dans le cas de l'acquisition compressée puisqu'elles ont un nombre de degrés de liberté faible au regard des matrices gaussiennes mais elles ont des propriétés similaires aux matrices gaussiennes pour ce qui est de la reconstruction [Rom09].

Ce chapitre est organisé comme suit. Nous commençons par rappeler la manière dont sont générées les matrices circulantes orthogonales et nous montrons les limites de cette construction. Ensuite, nous étudions les matrices circulantes partiellement orthogonales qui ont les mêmes avantages mais dont la génération est moins restrictive. Nous présentons quelques exemples pratiques de génération de telles matrices. Ensuite, nous effectuons le même travail en dimension supérieure avec les matrices circulantes par blocs, à blocs circulants. Enfin, dans un cadre applicatif, nous utilisons les différents résultats obtenus pour pallier au problème de dimension rencontré au chapitre précédent.

5.1 Quelques définitions

Nous commençons par rappeler quelques définitions et résultats sur les matrices circulantes et les matrices de Fourier puis nous introduisons quelques notations commodes.

5.1.1 Matrices circulantes

Une matrice circulante est une matrice dont les lignes se déduisent l'une de l'autre par translation. Plus précisément, si $c = (c_0, c_1, c_2, \ldots, c_{n-1})$ est la première

ligne d'une matrice circulante C alors,

$$C = \begin{pmatrix} c_0 & c_1 & c_2 & \dots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & & c_{n-3} \\ \vdots & & & \ddots & \vdots \\ c_1 & c_2 & c_3 & \dots & c_0 \end{pmatrix},$$

ou, en notation abrégée,

$$C_{ij} = c_{|i-j|}.$$

Nous remarquons alors que la diagonale et les sous-diagonales sont constantes et qu'une matrice circulante ne dépend que de sa première ligne. Nous avons aussi les propriétés suivantes.

Proposition 5.1. *Soit C et D des matrices circulantes. Alors nous avons,*

1. *C et D commutent : $CD = DC$,*
2. *CD est aussi une matrice circulante.*

5.1.2 Matrices de Fourier

Les matrices circulantes sont très liées à la transformée de Fourier. Nous rappelons que la matrice de Fourier F de taille $n \times n$ est définie par,

$$F = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \dots & w^{(n-1)^2} \end{pmatrix} \quad \text{avec} \quad w = e^{-\frac{2i\pi}{n}},$$

ou, en notation abrégée,

$$F_{ij} = w^{(i-1)(j-1)}.$$

Nous avons alors les propriétés suivantes,

i) F est symétrique, $F^T = F$,

ii) $FF^* = F\overline{F} = nI_n$.

Les matrices circulantes sont diagonalisables dans une base de Fourier. Nous résumons cette propriété fondamentale dans le théorème suivant.

Théorème 5.2. *Soit C une matrice circulante dont la première ligne est $(c_0, \dots, c_{n-1})$. Soit c cette première ligne mise en colonne. Alors C est diagonalisable en matrice de Fourier,*

$$nC = FDF^*,$$

et la matrice diagonale D vérifie,

$$\mathrm{diag}(D) = Fc.$$

En particulier, ce résultat montre que pour calculer les valeurs propres d'une matrice circulante, il suffit de faire la transformée de Fourier de sa première ligne.

Il est également intéressant d'un point de vue algorithmique puisqu'il montre que pour calculer le produit matriciel Cu, il suffit de faire une transformée de Fourier inverse de u suivi d'une multiplication coordonnées par coordonnées puisque D est diagonale, suivi d'une transformée de Fourier. L'intérêt est que nous n'avons pas eu besoin de calculer C pour calculer Cu. Il nous faut juste calculer la diagonale D qui vaut Fc où c est la première ligne de C. En somme, le calcul de Cu se résume à des transformées de Fourier.

5.1.3 Opérateurs • et †

Nous introduisons l'opérateur • qui réalise la multiplication case par case.

Définition 5.1. Soit A, B deux matrices de même taille. Nous définissons le produit case par case entre A et B que nous notons $A \bullet B$ par la matrice suivante,

$$(a_{ij}b_{ij})_{ij}.$$

Proposition 5.3. *Soient H et K des matrices diagonales de taille $n \times n$ et u vecteur de longueur n, alors nous avons les propriétés suivantes,*

- $\mathrm{diag}(HK) = \mathrm{diag}(H) \bullet \mathrm{diag}(K)$,
- $\mathrm{diag}(H * u) = \mathrm{diag}(H) \bullet u$

Nous introduisons ensuite un opérateur qui nous permet de caractériser certains vecteurs par rapport à leur transformée de Fourier ce qui nous sera utile par la suite.

Définition 5.2. Soit $a = (a_1, \ldots, a_n) \in \mathbb{C}^n$, nous notons $a^\dagger$ le vecteur,

$$(\overline{a_1}, \overline{a_n}, \overline{a_{n-1}}, \ldots, \overline{a_2}),$$

obtenu en remplaçant la première coordonnée de a par son conjugué et les autres coordonnées par leur symétrique conjugué.

L'opérateur † est lié à la transformée de Fourier par la relation suivante,

$$a^\dagger = \frac{1}{n} F^2 \overline{a}.$$

L'opérateur † permet également de caractériser les vecteurs dont la transformée de Fourier est réelle comme le montre la propriété suivante.

Proposition 5.4. *Soit $a \in \mathbb{C}^n$, les propositions suivantes sont équivalentes,*

i) Fa *est réel,*

ii) F^*a *est réel,*

iii) Il existe b *réel tel que* $Fa = b$,

iv) Il existe b *réel tel que* $F^*a = b$,

v) $a^\dagger = a$.

Nous présentons ensuite la méthode classique de génération de matrices circulantes orthogonales.

5.2 Matrices circulantes orthogonales

Une solution pour générer des lignes qui se déduisent les unes des autres par translation et qui forment une famille orthogonale est de sélectionner des lignes d'une matrice circulante orthogonale comme, par exemple, dans [Rom09]. Pour générer une telle matrice C, en notant c la première ligne de C mise en colonne, nous avons, d'après le théorème 5.2,

$$nC = FDF^*.$$

Pour que cette matrice circulante C soit orthogonale, il faut et il suffit que $DD^* = I_n$. En effet, nous avons successivement,

$$\begin{aligned} CC^* &= \frac{1}{n} FDF^* \frac{1}{n} (FDF^*)^* \\ &= \frac{1}{n^2} FDF^* FD^* F^* \\ &= \frac{1}{n} FDD^* F^*, \end{aligned}$$

donc $CC^* = I_n$ si et seulement si $DD^* = I$.

Ainsi, les éléments diagonaux de la matrice diagonale $D = \mathrm{diag}(d_1, \ldots, d_n)$ sont nécessairement de norme 1. Une condition supplémentaire doit être rajoutée si nous voulons que la matrice C soit réelle. En effet, d'après la proposition 5.4, puisque nous avons $Fc = d$, il faut $d = d^\dagger$ ce qui se traduit par,

$$d_1 = \pm 1, \tag{5.1}$$

$$(d_2, d_3, \ldots, d_n) = (\bar{d}_n, \bar{d}_{n-1}, \ldots, \bar{d}_2). \tag{5.2}$$

Pour générer une matrice réelle, circulante et orthogonale il suffit alors de générer n nombres complexes $d_1, \ldots, d_n$ de norme 1 qui vérifient les conditions (5.1) et (5.2).

Nous calculons ensuite la transformée de Fourier de ce vecteur ce qui nous donne la première ligne de la matrice réelle, circulante et orthogonale recherchée.

Cette méthode est très simple mais soulève plusieurs questions. Premièrement, nous n'avons pas besoin que toutes les lignes de la matrice C soient orthogonales puisque nous allons de toute façon n'en prendre qu'une petite partie. Les vecteurs restants sont donc contraints à être orthogonaux sans que cela nous soit utile.

Deuxièmement, du fait de ces contraintes inutiles, il est difficile d'en appliquer d'autres sur les lignes extraites. Par exemple, supposons que nous voulions que la somme des coefficients de chaque ligne extraite soit nulle. Chaque ligne est donc orthogonale au vecteur $(1, \dots, 1)$. Nous avons alors une famille orthogonale de $n+1$ vecteurs de $\mathbb{R}^n$ ce qui est impossible. Il est donc impossible de générer des lignes de somme nulle par cette méthode.

Si nous raisonnons en termes de degrés de liberté lors de la génération d'une telle matrice, la méthode classique en a $\lceil \frac{n-1}{2} \rceil$. En effet, nous devons fixer d qui vérifie $d^\dagger = d$. Le premier élément d_1 ne compte pas comme un degré de liberté puisque $d_1 = \pm 1$. Il reste donc $n-1$ entrées contraintes par $d^\dagger = d$ ce qui nous donne $\lceil \frac{n-1}{2} \rceil$ degrés de liberté.

Dans l'étude qui suit, nous proposons une construction moins restrictive de telles matrices qui cible précisément les lignes dont nous voulons qu'elles forment une famille orthogonale. Nous montrerons qu'il est alors possible de rajouter des contraintes comme, par exemple, forcer la somme d'une ligne à être nulle. Enfin, nous montrerons que le nombre de degrés de liberté est plus élevé qu'avec la construction classique, ce qui montre que la matrice ainsi générée se rapproche plus du comportement d'une matrice gaussienne.

5.3 Matrices circulantes partiellement orthogonales

5.3.1 Analyse et synthèse

Nous allons reformuler la contrainte d'orthogonalité $CC^* = I_n$ de manière à ce qu'elle ne force pas toutes les lignes à former une famille orthonormale. Pour cela, nous définissons tout d'abord les lignes de C dont nous voulons qu'elles forment un système orthogonal par l'ensemble $\Omega \subset [\![1, n]\!]$. Nous introduisons ensuite un opérateur P_Ω qui va sélectionner les lignes spécifiées par l'ensemble Ω. Ainsi, la matrice $P_\Omega C$ est de taille $|\Omega| \times n$ et contient uniquement les lignes de C spécifiées par Ω. Nous formons ensuite les différents produits scalaires de ces lignes entre elles par le produit $P_\Omega C (P_\Omega C)^T$. Nous voulons que ces produits scalaires soient tous nuls pour deux vecteurs différents. Il faut donc que ce produit forme une matrice diagonale. Nous remarquons également que comme la matrice C est circulante,

la matrice CC^T l'est également et donc sa diagonale est constante et celle de $P_\Omega C(P_\Omega C)^T$ aussi. Finalement, le problème se formule de la manière suivante.

Problème. *Soit $\Omega \subset [\![1, n]\!]$. Trouver une matrice circulante réelle non nulle $C \in \mathbb{R}^{n\times n}$ telle que,*

$$P_\Omega C C^T P_\Omega^T \text{ soit diagonale.} \tag{5.3}$$

Si $\Omega = [\![1, n]\!]$, nous avons $P_\Omega = I_n$ et nous retrouvons la condition CC^T diagonale ce qui force toutes les lignes de C à former une famille orthogonale.

Nous voulons donc trouver une matrice C circulante qui vérifie la condition (5.3). Nous allons raisonner successivement par condition nécessaire. La condition (5.3) équivaut à,

$$(CC^T)_{ij} = 0 \quad \text{pour tout} \quad i, j \in \Omega, i \neq j. \tag{5.4}$$

Or, la matrice CC^T est circulante car elle est produit de deux matrices circulantes. Un élément à la ligne i et à la colonne j de CC^T dépend donc uniquement de $|i - j|$. Nous posons donc,

$$\overline{\Omega} = \{|i - j| + 1, i \neq j, i, j \in \Omega\}, \tag{5.5}$$

et la condition (5.4) se réécrit,

$$(CC^T)_{1j} = 0 \quad \text{pour tout} \quad j \in \overline{\Omega}. \tag{5.6}$$

La contrainte (5.6) s'exprime sur la première ligne de la matrice CC^T. Il nous reste à lier cette première ligne avec celle de C.

Pour cela, notons s la première ligne de CC^T mise en colonne et c celle de C mise en colonne. Comme C est circulante nous appliquons le théorème 5.2, ce qui nous donne,

$$nC = FDF^*.$$

Nous avons donc,

$$nCC^T = nCC^* = FDD^*F^*. \tag{5.7}$$

L'égalité (5.7) est la diagonalisation de la matrice circulante CC^T par la matrice de Fourier puisque que DD^* est diagonale. Nous avons donc, toujours d'après le théorème 5.2,

$$Fs = \operatorname{diag}(DD^*).$$

Puis, en explicitant $\operatorname{diag}(DD^*)$ avec l'opérateur $\bullet$ qui réalise une multiplication case par case,

$$\begin{aligned} \operatorname{diag}(DD^*) &= \operatorname{diag}(D) \bullet \operatorname{diag}(D^*) \\ &= \operatorname{diag}(D) \bullet \overline{\operatorname{diag}(D)} \\ &= Fc \bullet \overline{Fc}. \end{aligned}$$

Finalement, le problème (5.3) se résume à trouver c réel tel que,

$$Fs = Fc \bullet \overline{Fc}, \tag{5.8}$$

sachant que s vérifie,

$$s_i = 0 \quad \text{pour tout} \quad i \in \overline{\Omega}.$$

Nous remarquons d'abord qu'il est nécessaire que Fs soit réel positif d'après (5.8). Il suffit alors que,

$$|Fc| = \sqrt{Fs},$$

pour que l'égalité (5.8) soit vérifiée. Aucune contrainte n'est imposée sur l'argument des coefficients de Fc. Nous devons juste faire en sorte que Fc vérifie (5.1) et (5.2) pour que c soit réel. Soit e un vecteur à coefficients complexes de module 1 qui vérifie les conditions (5.1) et (5.2). Nous pouvons écrire,

$$Fc = e \bullet \sqrt{Fs},$$

et finalement,

$$c = \frac{1}{n} F^* \left(e \bullet \sqrt{Fs} \right). \tag{5.9}$$

En conclusion, une matrice circulante réelle C vérifiant (5.3) doit nécessairement être de la forme,

$$c = \frac{1}{n} F^* \left(e \bullet \sqrt{Fs} \right), \tag{5.10}$$

où s est un vecteur qui vérifie les conditions suivantes :

i) s est réel et $s^\dagger = s$,

ii) Fs est réel positif,

iii) $s_i = 0$ pour tout $i \in \overline{\Omega}$.

et e est un vecteur d'éléments quelconques de module 1 vérifiant,

$$e^\dagger = e.$$

Nous vérifions ensuite qu'une telle matrice répond au problème (5.3) pour établir l'équivalence.

Le résultat obtenu se synthétise dans le théorème suivant. La dernière partie du théorème est établie dans les exemples qui suivent.

Théorème 5.5. *Soit $\Omega \subset [\![1, n]\!]$. Soit C une matrice réelle circulante dont les lignes repérées par l'ensemble Ω forment une famille orthogonale et soit c la première ligne de cette matrice C, alors,*

$$c = \frac{1}{n} F^* \left(e \bullet \sqrt{Fs} \right),$$

où s est un vecteur réel tel que,

i) $s^\dagger = s$,

ii) Fs *est réel positif,*

iii) $s_i = 0$ *pour tout* $i \in \overline{\Omega}$,

et e *est un vecteur d'éléments quelconques de module 1 vérifiant,*

$$e^\dagger = e.$$

De plus, cette famille est orthonormale si nous avons $s_1 = 1$ *et* c *est de somme nulle si,*

$$\sum_{i=1}^{n} s_i = 0.$$

En fait, la condition i du théorème 5.5 est superflue puisqu'elle est impliquée par la condition ii. Il s'agit juste de montrer une piste pour la construction d'une matrice vérifiant la condition i.

Le nombre de degrés de liberté de cette nouvelle construction est donc le suivant. Le choix de e donne toujours $\lceil \frac{n-1}{2} \rceil$ degrés de liberté auxquels se rajoutent ceux du choix de s. Le nombre total de degré de liberté est donc,

$$\left\lceil \frac{n-1}{2} \right\rceil + \left\lceil \frac{n-1-\left|\overline{\Omega}\right|}{2} \right\rceil.$$

Le nombre de degré de liberté est donc plus important d'autant plus que dans le cadre de l'acquisition compressée, $\left|\overline{\Omega}\right|$ est petit devant n.

Nous proposons maintenant plusieurs exemples de génération de telles matrices en petite dimension.

5.3.2 Exemple pratique

Le premier exemple détaille les étapes établies dans la précédente section et montre la génération pratique d'une matrice circulante partiellement orthogonale.

En effet, supposons que l'on ait besoin d'une matrice circulante dont la première et la troisième ligne forment une famille orthogonale. Nous avons donc,

$$\Omega = \{1, 3\},$$

puis, d'après la définition (5.5), nous trouvons,

$$\bar{\Omega} = \{3\},$$

c'est à dire que le vecteur s doit avoir sa troisième coordonnée nulle. Nous voulons également que les lignes de cette matrice soient de norme 1, nous fixons alors $s_1 = 1$.

Ensuite, il nous faut compléter s de manière à ce que $s^\dagger = s$ ce qui donne juste une contrainte de symétrie sur s. Il faut aussi avoir Fs positif. Si tel n'est pas le cas, c'est à dire $\alpha = \min(Fs)_i < 0$ alors il existe $k > 0$ tel que,

$$F \begin{pmatrix} 1 \\ ks_2 \\ ks_3 \\ \vdots \\ ks_n \end{pmatrix} \text{ positif.}$$

Nous appliquons ensuite la formule (5.9). Si nous choisissons,

$$s = \begin{pmatrix} 1 & 0.2114 & 0 & 0.2114 \end{pmatrix}^T,$$

alors,

$$Fs = \begin{pmatrix} 1.4227 & 1 & 0.5773 & 1 \end{pmatrix}^T.$$

En prenant,

$$e = \begin{pmatrix} -1 & -0.41 + 0.9121i & -1 & -0.41 - 0.9121i \end{pmatrix}^T,$$

nous avons,

$$c = \begin{pmatrix} -0.6932 & 0.3478 & -0.2831 & -0.5643 \end{pmatrix}^T,$$

et donc,

$$C = \begin{pmatrix} -0.6932 & 0.3478 & -0.2831 & -0.5643 \\ -0.5643 & -0.6932 & 0.3478 & -0.2831 \\ -0.2831 & -0.5643 & -0.6932 & 0.3478 \\ 0.3478 & -0.2831 & -0.5643 & -0.6932 \end{pmatrix},$$

et le produit CC^* qui regroupe tous les produits scalaires des lignes de C,

$$CC^T = \begin{pmatrix} 1 & 0.2114 & -0 & 0.2114 \\ 0.2114 & 1 & 0.2114 & -0 \\ -0 & 0.2114 & 1 & 0.2114 \\ 0.2114 & -0 & 0.2114 & 1 \end{pmatrix}.$$

La matrice C est donc une matrice circulante dont les lignes 1 et 3 (et donc 2 et 4) forment une famille orthonormale mais qui n'est pas une matrice orthogonale.

5.3.3 Exemple de « somme nulle »

Un autre exemple illustre un des intérêts de cette technique. Il s'agit de construire une matrice circulante dont un sous-ensemble des lignes forme une famille orthonormale et dont la somme d'une ligne (et donc de toutes les lignes) vaut 0. Nous avons déjà montré dans l'introduction que c'était impossible en utilisant la technique classique. Regardons comment se traduit la contrainte de somme nulle sur les lignes d'une matrice C. La première ligne C étant notée c, nous avons,

$$\sum_{i=0}^{n-1} c_i = 0,$$

ce qui est équivalent à,

$$(Fc)_1 = 0.$$

Ainsi, il faut et il suffit que $(Fs)_1 = 0$ d'après (5.8), c'est à dire,

$$\sum_{i=0}^{n-1} s_i = 0.$$

En prenant,

$$s = \begin{pmatrix} 1 & -0.5 & 0 & -0.5 \end{pmatrix},$$

qui est de somme nulle et,

$$e = \begin{pmatrix} -1 & 0.7651 - 0.644i & 1 & 0.7651 + 0.644i \end{pmatrix},$$

nous avons,

$$C = \begin{pmatrix} 0.7361 & -0.6755 & -0.029 & -0.0316 \\ -0.0316 & 0.7361 & -0.6755 & -0.029 \\ -0.029 & -0.0316 & 0.7361 & -0.6755 \\ -0.6755 & -0.029 & -0.0316 & 0.7361 \end{pmatrix},$$

qui est une matrice circulante telle que toutes les lignes sont de somme nulle et telle que la ligne 1 et la ligne 3 forment une famille orthonormale comme le montre la matrice des produits scalaires des lignes,

$$CC^T = \begin{pmatrix} 1 & -0.5 & 0 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{pmatrix}.$$

La méthode de génération précédente de matrices partiellement orthogonales est adapté pour la prise de mesures sur un signal unidimensionnel. Nous allons généraliser cette méthode au cas de la dimension 2.

5.4 Matrices circulantes par blocs, à blocs circulants partiellement orthogonales

Dans la section précédente, notre problème était de trouver des vecteurs qui étaient orthogonaux entre eux et décalés les uns par rapport aux autres. Le même problème se pose en deux dimensions. Nous cherchons des matrices se déduisant les unes des autres par décalage et dont le produit scalaire des linéarisés de ces matrices est nul.

5.4.1 Analyse et synthèse

Soit M une matrice carrée. Pour formuler le problème posé, il nous faut linéariser cette matrice et toutes les matrices décalées et ensuite trouver une équation qui impose que leur produit scalaire soit nul. Nous allons donc définir la matrice B_M qui contient toutes les linéarisées des matrices décalées. Nous montrerons ensuite que si les lignes sont ordonnées correctement, cette matrice est circulante par blocs et que chacun des blocs qui la compose est aussi circulant. Cette propriété forte va nous permettre d'utiliser des propriétés similaires à celles utilisées dans la section précédente pour les matrices simplement circulantes.

Définition 5.3. Soit M une matrice de taille $n \times n$. Soit B_M une matrice de taille $n^2 \times n^2$ définie de la manière suivante. La première ligne de B_M est la matrice M mise en ligne, ligne par ligne. Le reste de la matrice se déduit en imposant que la matrice B_M soit constituée de n^2 blocs circulants de taille $n \times n$ et soit circulante par blocs.

Exemple 5.1. Prenons $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Alors, la première ligne de B_M est,

$$(a, b, c, d).$$

Puis en complétant, car B_M est circulante par blocs, à blocs circulants, nous obtenons,

$$B_M = \begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix}.$$

Nous pouvons alors montrer que la matrice M décalée f fois vers la droite et e fois vers le bas et ensuite linéarisée ligne par ligne se trouve à la ligne $ne + f + 1$ de la matrice B_M. En effet, en reprenant l'exemple précédent, si M est décalée une

fois vers la droite puis vers le bas, c'est à dire $e = f = 1$, nous avons,

$$S_{11}(M) = \begin{pmatrix} d & c \\ b & a \end{pmatrix}.$$

La matrice linéarisée ligne par ligne de $S_{11}(M)$, se trouve bien à la ligne $4 = 2 \times 1 + 1 + 1$ de la matrice B_M.

Nous sommes à présent en mesure de formuler le problème qui impose à certaines lignes de B_M d'être orthogonales entres elles. Ces lignes choisies peuvent être codées par un ensemble $\Lambda \subset [\![0, n-1]\!]^2$ qui représente les décalages correspondants dans la matrice M. Nous définissons l'opérateur P_Λ qui sélectionne les lignes correspondant aux décalages codés par Λ. Le problème en deux dimensions s'énonce comme suit.

Problème. *Soit $\Lambda \subset [\![0, n-1]\!]^2$. Trouver une matrice réelle non nulle M de taille $n \times n$ telle que,*

$$P_\Lambda B_M B_M^T P_\Lambda^T \text{ soit diagonale.} \tag{5.11}$$

Réécrivons la condition (5.11) en définissant l'ensemble suivant,

$$L = \{ne + f + 1, (e, f) \in \Lambda\}.$$

Nous avons alors,

$$(B_M B_M^T)_{ij} = 0 \quad \text{pour tout} \quad i, j \in L, i \neq j. \tag{5.12}$$

Cette dernière condition, qui force des entrées de la matrice $B_M B_M^T$ à être nulles se résume à des conditions de nullité sur la première ligne seulement de cette matrice puisque la matrice $B_M B_M^T$ est définie par sa première ligne.

En effet, si $(B_M B_M^T)_{ij} = 0$ avec $i = ne + f + 1$ et $j = ne' + f' + 1$ alors,

$$(B_M B_M^T)_{f+1, n|e-e'|+f'+1} = 0,$$

car la matrice est circulante par blocs. Puis, comme chaque bloc est circulant, nous obtenons,

$$(B_M B_M^T)_{1, n|e-e'|+|f-f'|+1} = 0.$$

Finalement, la condition (5.11) est vérifiée si et seulement si il existe une matrice N telle que,

$$N_{ij} = 0 \quad \text{pour tout} \quad (i, j) \in \overline{\Lambda}, \tag{5.13}$$

avec,

$$\overline{\Lambda} = \{(|e - e'|, |f - f'|), (e, f), (e', f') \in \Lambda, (e, f) \neq (e', f')\}, \tag{5.14}$$

et telle que,

$$B_M B_M^* = B_N. \tag{5.15}$$

Comme dans le cas des matrices circulantes partiellement orthogonales, nous allons maintenant lier la première ligne d'une matrice circulante par blocs, à blocs circulants à ses valeurs propres pour pouvoir expliciter la relation (5.15). Pour cela, nous nous servons du théorème suivant qui est l'analogue du théorème 5.2 en dimension 2.

Théorème 5.6. *Soit $C = B_M$ une matrice circulante par blocs, à blocs circulants, alors C est diagonalisable par la matrice $G = F \otimes F$,*

$$n^2 C = GDG^*,$$

et la matrice diagonalisée D vérifie,

$$D = \operatorname{diag} \operatorname{vec}(FM^T F).$$

Nous introduisons également l'opérateur † en dimension 2.

Définition 5.4. Soit M une matrice de taille $n \times n$. Nous notons $M^\dagger$ la matrice,

$$\frac{1}{n^2} F^2 \overline{M} F^2.$$

En fait, il s'agit simplement de réordonner les termes de la matrice $\overline{M}$. En effet, nous avons,

$$M^\dagger = \begin{pmatrix} \overline{m_{11}} & \overline{m_{1n}} & \dots & \overline{m_{12}} \\ \overline{m_{n1}} & \overline{m_{nn}} & \dots & \overline{m_{n2}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{m_{21}} & m_{2n} & \dots & \overline{m_{22}} \end{pmatrix}.$$

La sous-matrice constituée des $n-1$ dernières lignes et des $n-1$ dernières colonnes est obtenue par rotation de 180° suivie d'une conjugaison. Les autres blocs subissent la même transformation.

L'opérateur † est utile puisqu'il permet de caractériser les matrices dont la transformée de Fourier est réelle, comme l'indiquent les propriétés suivantes.

Proposition 5.7. *Soit M une matrice de taille $n \times n$, les propriétés suivantes sont équivalentes,*

i) FMF est réelle,

ii) F^*MF^* est réelle,

iii) Il existe A réelle telle que $FAF = M$,

iv) Il existe A réelle telle que $F^*AF^* = M$,

v) $M^\dagger = M$.

Le théorème 5.6 nous permet de démontrer sans difficulté les propriétés suivantes.

Proposition 5.8. *Soit M et N des matrices de taille $n \times n$, alors nous avons les propriétés suivantes,*

i) $B_N = B_M$ *si et seulement si* $N = M$,

ii) $B_M^* = B_{M^\dagger}$,

iii) $B_M B_N = B_{M \star N}$,

où $\star$ désigne le produit de convolution.

Revenons à présent à la condition (5.15) précédemment établie. D'après les propriétés de la proposition 5.8, nous avons,

$$N = M \star M^\dagger. \tag{5.16}$$

En détaillant le produit de convolution, nous obtenons,

$$N = \frac{1}{n^2} F^* (FMF \bullet FM^\dagger F) F^*,$$

donc,

$$FNF = FMF \bullet FM^\dagger F. \tag{5.17}$$

Nous remarquons ensuite que $FM^\dagger F = \overline{FMF}$, puisque nous avons successivement,

$$\begin{aligned} \overline{FM^\dagger F} &= \frac{1}{n^2} \overline{FF^2 \overline{M} F^2 F} \\ &= \frac{1}{n^2} \overline{F^3 \overline{M} F^3} \\ &= \overline{F^* \overline{M} F^*} \\ &= FMF. \end{aligned}$$

donc finalement,

$$FNF = FMF \bullet \overline{FMF}. \tag{5.18}$$

Nous remarquons que la matrice N vérifie nécessairement plusieurs propriétés. Elle est d'abord réelle d'après (5.15) puisque M est réelle. De plus, la matrice FNF doit être positive d'après (5.18). Nous pouvons ensuite isoler M, car nous avons successivement,

$$FNF = |FMF|^2,$$

puis,

$$|FMF| = \sqrt{FNF},$$

où $\sqrt{\cdot}$ désigne l'opérateur qui prend la racine carrée de chaque entrée de la matrice. Comme dans l'expression précédente, seul le module de FMF est fixé, nous pouvons à présent écrire,

$$FMF = E \bullet \sqrt{FNF},$$

où E est une matrice de module 1. Mais comme les matrices M et N sont réelles, les matrices FMF et FNF sont †-invariantes et donc E aussi. Nous avons donc,

$$E = E^{\dagger}.$$

Finalement, les solutions du problème (5.11) sont nécessairement de la forme,

$$M = \frac{1}{n^2} F^* \left(E \bullet \sqrt{FNF} \right) F^*, \tag{5.19}$$

où N est une matrice réelle telle que FNF est positive et telle que,

$$N_{ij} = 0 \quad \text{pour tout} \quad (i,j) \in \overline{\Lambda}, \tag{5.20}$$

et E est une matrice quelconque d'éléments de module 1 vérifiant,

$$E^{\dagger} = E.$$

De la même manière, nous vérifions que de telles matrices répondent bien au problème (5.11) pour avoir l'équivalence. Nous résumons les solutions au problème 5.4.1 dans le théorème suivant.

Théorème 5.9. *Soit $\Lambda \subset [\![0, n-1]\!]^2$. Soit M une matrice réelle telle que les décalages repérés par Λ donnent une famille de matrice dont les vectorisées forment une famille orthogonale. Alors M est de la forme,*

$$M = \frac{1}{n^2} F^* \left(E \bullet \sqrt{FNF} \right) F^*, \tag{5.21}$$

où N est une matrice réelle telle que

i) $N^{\dagger} = N$,

ii) FNF est réelle positive,

iii) $N_{ij} = 0$ pour tout $(i,j) \in \overline{\Lambda}$,

et E est une matrice quelconque d'éléments de module 1 vérifiant,

$$E^{\dagger} = E.$$

De plus, cette famille est orthonormale si nous avons $N_{11} = 1$ et c est de somme nulle si,

$$\sum_{i,j=1}^{n} N_{ij} = 0.$$

En fait, la condition i du théorème 5.9 est superflue puisqu'elle est impliquée par la condition ii. Il s'agit juste de montrer une piste pour la construction d'une matrice vérifiant la condition ii.

5.4.2 Exemple pratique

Par exemple, supposons que l'on cherche une matrice M dont les décalages suivant le motif Λ forment une famille de matrices dont les vectorisées forment une famille orthonormale. Nous avons ainsi le motif,

$$\Lambda = \{(0,0),(0,2),(2,0),(2,2)\},$$

donc,

$$\overline{\Lambda} = \{(0,2),(2,0),(2,2)\}.$$

La matrice N comporte donc trois zéros aux endroits spécifiés par $\overline{\Lambda}$. Les autres entrées sont générées aléatoirement de manière à ce que $N^\dagger = N$. Nous prenons,

$$N = \begin{pmatrix} 1 & -0.0933 & 0 & -0.0933 \\ 0.022 & -0.3546 & -0.2009 & 0.1454 \\ 0 & 0.3184 & 0 & 0.3184 \\ 0.022 & 0.1454 & -0.2009 & -0.3546 \end{pmatrix}.$$

De la même manière, si la matrice FNF n'est pas positive, il suffit de redimensionner toutes les entrées autres que N_{11}.

Nous appliquons ensuite la formule (5.19) avec,

$$E = \begin{pmatrix} 1 & -0.9927 + 0.121i & -1 & -0.9927 - 0.121i \\ 0.916 - 0.4011i & -0.9519 + 0.3064i & 0.2904 - 0.9569i & -0.9843 - 0.1768i \\ -1 & -0.8596 - 0.511i & 1 & -0.8596 + 0.511i \\ 0.916 + 0.4011i & -0.9843 + 0.1768i & 0.2904 + 0.9569i & -0.9519 - 0.3064i \end{pmatrix},$$

et nous trouvons,

$$M = \begin{pmatrix} -0.3474 & -0.0625 & 0.4475 & -0.0129 \\ 0.1112 & 0.1992 & 0.358 & -0.0059 \\ -0.2051 & 0.0476 & -0.0833 & -0.1195 \\ -0.1456 & 0.1436 & -0.1155 & 0.6116 \end{pmatrix}. \tag{5.22}$$

La matrice M et ses décalées suivant le motif Λ forment une famille de matrices dont les vectorisées forment une famille orthogonale. En plaçant les vectorisées de ces matrices en ligne dans la matrice F_{virt}, nous avons alors $F_{\text{virt}} F_{\text{virt}}^T = I$.

5.5 Mise en œuvre

Reprenons le problème établi au chapitre précédent.

$$\arg\min_{u \geqslant 0} \|u\|_1 \quad \text{tel que} \quad M_{\text{virt}}^T (F_{\text{virt}} F_{\text{virt}}^T)^{-1} F_{\text{virt}} u = \frac{m}{n_P} s. \tag{5.23}$$

L'idée est donc de choisir des mesures tel que nous ayons,

$$F_{\text{virt}} F_{\text{virt}}^T = I.$$

Les mesures virtuelles consistent en des décalages d'une mesure de référence, ces décalages étant contrôlés par un motif de mesures E. Pour que $F_{\text{virt}} F_{\text{virt}}^T = I$, il faut que tous ces décalages donnent des mesures qui une fois vectorisées forment une famille orthonormale. En suivant la résolution du problème 5.4.1, nous devons calculer $\overline{E}$. Le motif E est calculé à partir du motif à détecter P et du taux de mesures utilisé. Nous supposerons que le motif à détecter est rectangulaire. Dans ce cas-là, E est quasi-rectangulaire et la proposition suivante nous donne $\overline{E}$.

Proposition 5.10. *Soit E un motif quasi-rectangulaire et R le motif rectangulaire le plus petit contenant E. Alors, nous avons,*

$$\overline{E} = R \setminus \{(0,0)\}. \tag{5.24}$$

Nous pouvons à présent générer une mesure de référence d'après le théorème 5.9 telle que les mesures glissées codées par le motif de mesures E forment une famille orthogonale, une fois linéarisée. Nous avons donc $F_{\text{virt}} F_{\text{virt}}^T = I$.

Il reste le problème de la multiplication par $M_{\text{virt}}^T F_{\text{virt}}$ et par sa transposée qui sont nécessaires pour résoudre l'algorithme. Du fait de la structure circulante des matrices M_{virt} et F_{virt}, nous pouvons calculer ces produits efficacement à l'aide de trois transformées de Fourier.

Nous pouvons à présent résoudre la minimisation (5.23) sans problème d'inversion de matrice ou de multiplication trop coûteuse.

5.6 Applications

Nous comparons tout d'abord les temps de calculs des deux algorithmes, le premier reposant sur le calcul explicite de $M_{\text{virt}}^T F_{\text{virt}}$ comme au précédent chapitre et le nouvel algorithme qui calcule $M_{\text{virt}}^T F_{\text{virt}} u$ et $(M_{\text{virt}}^T F_{\text{virt}})^T v$ avec des transformées de Fourier. La table 5.1 résume les temps de calcul constatés pour différents taux de mesures. Ces temps de calculs n'incluent pas les calculs préliminaires comme l'inversion de matrice requise par le premier algorithme. Même en petite dimension, l'algorithme reposant sur des transformées de Fourier pour effectuer des multiplications matricielles est plus rapide.

5.6.1 Sur l'image `peppers`

Nous testons ensuite l'algorithme de détection compressée de motifs sur l'image couleur classique `peppers` de taille 384×512. La figure 5.1a montre l'image

Mesures virtuelles (%)	Temps d'exécution matriciel (s)	Temps d'exécution Fourier (s)
5	11.27	2.86
10	12.94	3.9
15	8.56	8.39
20	11.36	5.39
25	8.34	5.71
30	9.49	6.72

Table 5.1 – Comparaison des temps d'exécution des deux implémentations de la minimisation (5.23) sur l'image extraite de `peppers` pour différents taux de mesures

originale `peppers`. Nous avons ensuite incrusté dans cette image 10 motifs `invader` représentée à la figure 5.1 et nous nous proposons de détecter ces motifs.

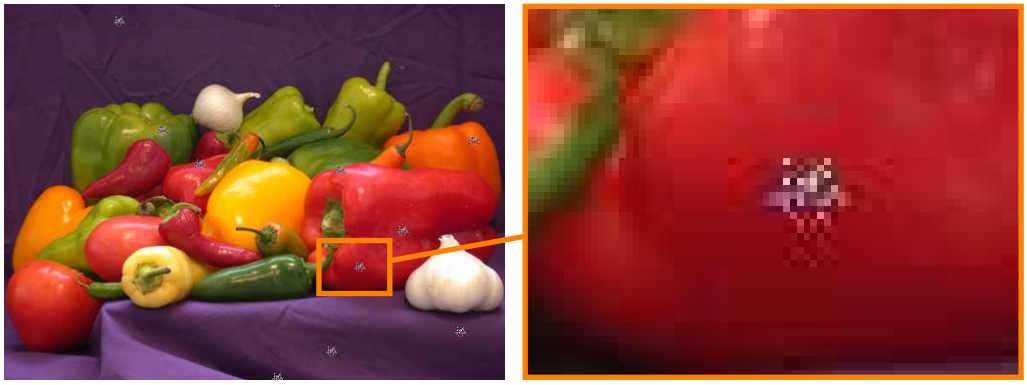
Mesures virtuelles (%)	Mesures effectives (%)	Efficacité de la reconstruction α	Temps d'exécution (s)	Erreurs
5	5.99	1.2	69.76	0
10	11.38	1.14	60.09	0
15	16.69	1.11	60.53	0
20	21.94	1.1	162.3	0
25	27.17	1.09	136.07	0
30	32.37	1.08	123.26	0

Table 5.2 – Mesures supplémentaires pour la détection de motifs sur l'image `peppers` pour différents taux de mesures

La table 5.2 montre quelques caractéristiques de l'algorithme de détection compressée de motifs pour différents taux de mesures. Du fait de la taille beaucoup plus grande de l'image, le coefficient α se rapproche de 1, ce qui veut dire que la détection compressée de motifs impose très peu de mesures supplémentaires par rapport à la détection compressée de signature. Tous les motifs sont correctement détectés même pour un taux de mesures assez faible de 5%.



(a) Image originale `peppers`



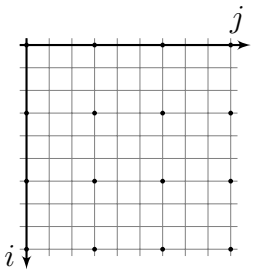
(b) Image `peppers` avec incrustations de 10 motifs `invader`

(c) Zoom sur le motif

Figure 5.1 – Image originale `peppers` et incrustations de motifs

5.6.2 Sur l'image `bâche`

Nous testons ensuite l'algorithme de détection compressée de motifs sur l'image du chapitre précédent qui est une image multispectrale réelle issue du projet HYPLITT d'ACTIMAR [SSL, JGBT12]. Nous nous proposons de détecter la mire présente au milieu de l'image avec sa structure en damier. L'image est de taille 316×312 et compte 160 bandes. Le motif que nous recherchons a la structure suivante,



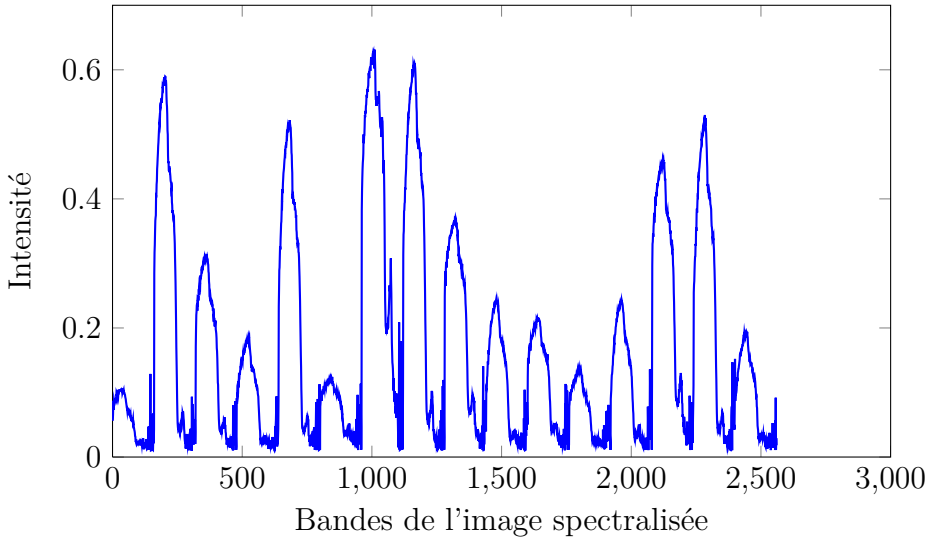


Figure 5.2 – Signature à détecter dans l'image *spectralisée* pour la détection du motif en damier

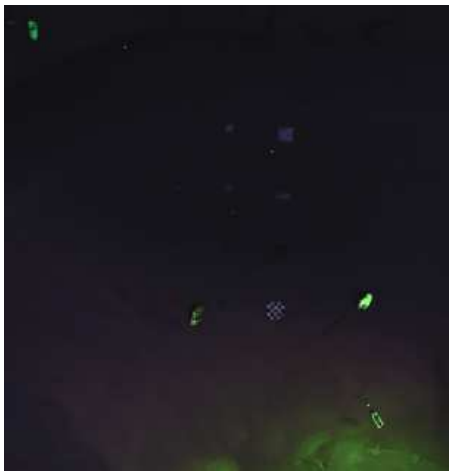
La figure 5.3 illustre les résultats obtenus. L'image `bâche` est représentée en fausse couleur à la figure 5.3a. La figure 5.3b montre le résultat de la détection pour un taux de mesures de 5%. La structure en damier est correctement détectée. La figure 5.3c montre le résultat de la détection avec 30% des données. Un agrandissement de la zone du motif à la figure 5.3d permet de voir que le motif est détecté là où le pixel est le plus lumineux. L'apparition de pixels secondaires est due à la structure répétitive du motif.

Le graphe de la figure 5.2 montre la signature résultante s qu'il faut détecter dans la minimisation (5.23). Elle résulte de la concaténation des 16 signatures prises sur l'image du motif en damier, elle est donc de longueur $16 \times 160 = 2560$.

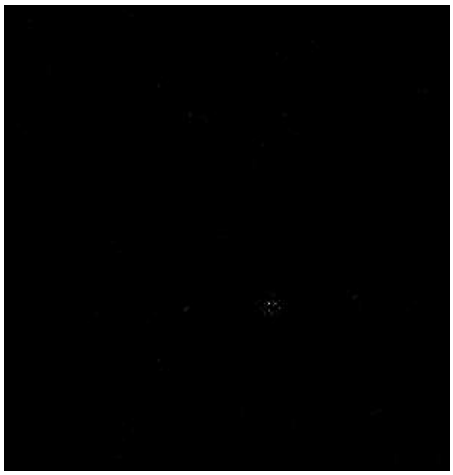
La table 5.3 résume les caractéristiques de l'algorithme pour différents taux de mesures. Comme dans le cas précédent, le surplus de mesures à effectuer n'est pas très grand. Qui plus est, le motif de mesures choisi n'est pas optimal car le motif n'est pas rectangulaire. Nous pourrions abaisser encore α si nous prenions un motif de mesures éclaté à l'image de la structure que nous voulons détecter. Les temps de calculs sont également surprenants au regard de ceux de l'exemple précédent qui ne compte que 3 bandes alors que celui-ci en compte 160. Ce résultat est sûrement dû à la faible dynamique de l'image `bâche` par rapport à l'image `peppers`.

Mesures virtuelles (%)	Mesures effectives (%)	Efficacité de la reconstruction α	Temps d'exécution (s)
5	6.4	1.28	15.45
10	11.93	1.19	18.77
15	17.35	1.16	25.36
20	22.7	1.13	19.67
25	28	1.12	23.48
30	33.28	1.11	28.26

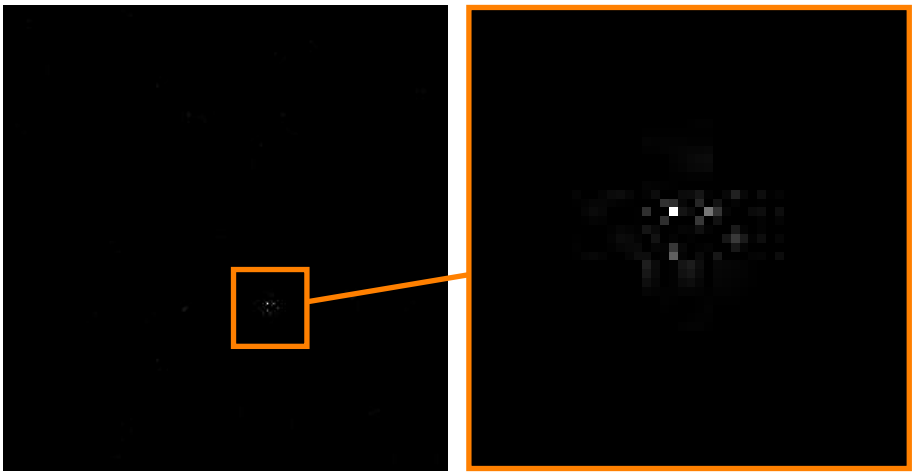
Table 5.3 – Mesures supplémentaires pour la détection de motifs sur l'image `bâche` pour différents taux de mesures



(a) Image originale `bâche` en fausses couleurs



(b) Détection compressée de motifs, 5% des données



(c) Détection compressée de motifs, 30% des données

(d) Zoom sur le motif

Figure 5.3 – Image originale `bâche` et détection de motifs

5.7 Conclusion

Dans ce chapitre, après avoir rappelé les limitations rencontrées dans les précédents chapitres lorsqu'il s'agissait d'appliquer une détection compressée de motifs, nous proposons tout d'abord d'utiliser des mesures glissées orthogonales qui permettent de simplifier l'inversion de la matrice $F_{\text{virt}}F_{\text{virt}}^T$. Nous montrons que les mesures glissées sont en fait liées aux matrices circulantes par blocs, à blocs circulants. Il nous faut donc étudier les matrices orthogonales circulantes par blocs, à blocs circulants. Pour ce faire, nous rappelons la méthode classique de génération de matrices circulantes orthogonales qui repose principalement sur une transformée de Fourier. Nous mettons ensuite en évidence les limites de cette méthode en montrant qu'une contrainte simple rend impossible la génération classique. Nous introduisons alors les matrices circulantes partiellement orthogonales et proposons une méthode simple de génération de telles matrices reposant sur deux transformées de Fourier. Nous généralisons enfin ces matrices au cas de la dimension 2 qui nous intéresse ici. Nous sommes alors capables de générer une matrice de mesures F_{virt} qui vérifie $F_{\text{virt}}F_{\text{virt}}^T = I$ et qui est adaptée aux décalages fixés par le motif de mesures. Nous validons ensuite notre approche en appliquant l'algorithme de détection compressée de motifs sur des images plus larges que celles du chapitre précédent. Même en petite dimension, la nouvelle implémentation est plus rapide. Elle prend en charge sans difficulté des images de quelques centaines de pixels de coté avec une centaine de bandes.

Conclusion

L'objet de cette thèse est la détection de points d'intérêt dans une image multispectrale. Nous avons exploré une première piste à travers l'utilisation de concepts issus de la géométrie riemannienne. En effet, le tenseur de Di Zenzo, qui est la généralisation du gradient d'une image monochrome à des images multispectrales peut être interprété comme la définition d'une métrique euclidienne sur l'image. Dès lors, en faisant varier cette métrique, nous retrouvons les méthodes classiques parmi lesquelles le traitement marginal et le traitement par luminance. En fait, le choix d'une métrique et l'application de l'algorithme est équivalent à une projection préalable de l'image suivie d'une détection classique avec une métrique euclidienne c'est à dire le tenseur de Di Zenzo. L'intérêt de ce nouveau point de vue est donc d'unifier différentes méthodes mais il ne débouche pas sur de nouveaux algorithmes. Par la suite, conscient de la quantité de calculs nécessaires pour traiter les données multispectrales, nous nous sommes intéressés à une discipline récente du traitement du signal et des images qui propose un autre modèle d'acquisition du signal. L'acquisition compressée permet de fusionner les étapes d'échantillonnage et de compression habituelles. La décompression est remplacée par une étape de reconstruction qui nécessite la résolution de problèmes de minimisation complexes reposant sur une norme L_1.

Nous consacrons ainsi un chapitre entier à l'étude des algorithmes de type Bregman qui permettent la résolution de telles minimisations. Notre choix s'est porté sur les algorithmes de type Bregman en raison de leur flexibilité car ils sont capables de résoudre aussi bien des problèmes contraints que non contraints grâce à une technique de scindement. Ils sont également très efficaces puisqu'ils ne reposent que sur des multiplications de matrices et des opérations de seuillage. Nous illustrons cette souplesse en réalisant un algorithme de débruitage suivant le modèle ROF. Nous illustrons également la reconstruction d'image par acquisition compressée.

Nous présentons ensuite l'algorithme de détection de signature dans une image multispectrale de Guo et Osher qui est assez original puisqu'il repose également sur une minimisation L_1. Nous étudions ensuite s'il est possible de faire fonctionner cet algorithme avec des données issues d'une acquisition compressée. En faisant une

hypothèse sur la matrice de mesure de l'image, nous aboutissons à une nouvelle minimisation qui permet de réaliser une détection de signature avec des données compressées sans reconstruire l'image multispectrale. La souplesse des algorithmes de Bregman nous permet de modifier les fonctions régularisantes pour obtenir une meilleure reconstruction. Nous obtenons ainsi des résultats comparables à la détection de signature classique de Guo et Osher avec 30% des données seulement.

Le chapitre suivant rend ce nouvel algorithme encore plus intéressant puisque nous montrons comment le généraliser pour qu'il réalise de la détection de motifs toujours dans un contexte d'acquisition compressée. Cette généralisation repose sur une transformation formelle d'une image appelée *spectralisation* qui construit une nouvelle image multispectrale contenant l'information des motifs dans ses bandes. Nous mettons en évidence les problèmes posés par l'application du nouvel algorithme sur une image spectralisée. Le premier est la reconstruction de mesures sur l'image *spectralisée* à partir de mesures sur l'image réelle. Nous introduisons un coefficient qui mesure l'efficacité de la reconstruction de mesures. Le deuxième problème est de choisir des mesures qui donnent un coefficient de mesures acceptable. Nous introduisons donc les mesures glissées et donnons une formule calculant le coefficient de reconstruction pour de telles mesures. Ces mesures dépendant d'un motif de mesures, nous démontrons une formule d'optimisation de ce motif de mesures lorsque le motif à détecter est simple. Les expérimentations montrent que le surplus de mesures nécessaires pour réaliser une détection compressée de motifs est raisonnable. En revanche, l'utilisation de mesures glissées sans propriété supplémentaire limite les dimensions de l'image traitée.

Dans le dernier chapitre, nous levons cette limitation en proposant d'utiliser des matrices de mesures extraites de matrices circulantes et orthogonales. Cela nous permet de résoudre les minimisations pour des images de grande taille. Néanmoins, nous n'utilisons pas la méthode de génération classique de matrices circulantes et orthogonales. Nous proposons en effet une nouvelle méthode qui génère une matrice circulante dont seulement les lignes requises forment un système orthogonal. Cela permet théoriquement de générer de meilleures matrices de mesures car leur construction nécessite un nombre accru de données aléatoires. Nous nous limitons à la génération de lignes orthogonales pour simplifier nos calculs mais la méthode utilisée peut très bien être généralisée et générer des lignes avec des produits scalaires spécifiques. Nous illustrons l'utilisation de ces matrices de mesures sur des images de taille moyenne avec un plus grand nombre de motifs à détecter.

Les différentes notions introduites dans cette thèse ont soulevé plusieurs questions intéressantes. En effet, d'un point de vue général, cette thèse a pour point de départ la minimisation de Guo et Osher que nous adaptons à l'acquisition compressée. En fait, la question de la prolongation des traitements classiques dans un contexte compressé se pose. Un certain nombre de traitements ont d'ailleurs

déjà été étudiés. Nous pouvons citer la soustraction d'arrière plan [1] dans [CSD+08] ou le démélange d'image multispectrale [LSKZ11].

Plus particulièrement, nous avons illustré la détection compressée de signature avec des matrices de mesures gaussiennes alors que les matrices circulantes semblent avoir un comportement voisin en terme d'erreur maximale lors de l'approximation de l'identité et sont beaucoup plus réalistes d'un point de vue physique.

Concernant la détection compressée de motifs développée au chapitre 4 et 5, il y a deux pistes que nous aimerions poursuivre.

Premièrement, les mesures utilisées sont décalées de manière circulaire, les entrées dépassant d'un coté sont réinjectées de l'autre. Cela signifie qu'une mesure sous sa forme matricielle est de même taille spatiale que l'image. Dès lors, nous pouvons envisager le cas où la mesure est plus petite que l'image ce qui reviendrait à prendre une mesure périodique spatialement. La prise de mesure serait alors plus simple car elle commencerait par un découpage de l'image en sous-image de la taille des périodes. En revanche, l'efficacité de l'encodage de l'image par la prise de cette mesure sera moins efficace car la matrice de mesures aura beaucoup moins de degrés de liberté. À l'opposé, une mesure plus grande que l'image peut améliorer l'encodage.

Ensuite, les matrices de mesures issues de matrices circulantes partiellement orthogonales sont théoriquement plus souples que celles issues de matrices orthogonales, d'après leurs degrés de liberté. Il nous faudrait trouver une mesure statistique qui puisse quantifier la qualité de l'encodage d'une matrice de mesures pour pouvoir les comparer. Dans le cas où ces matrices se révéleraient plus performantes, il existe un moyen d'augmenter encore le nombre de degrés de liberté sans que le calcul d'inversion soit difficile en fixant certains produits scalaires à des valeurs non nulles.

1. *background subtraction* en anglais.

Appendices

Annexe A

Produit de Kronecker

A.1 Définition et propriétés immédiates

Définition A.1. Soient A une matrice de taille $m \times n$ et B une matrice de taille $p \times q$. Nous définissons le produit de Kronecker des matrices A et B et nous le notons $A \otimes B$ la matrice de taille $mp \times nq$ suivante,

$$\begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Proposition A.1. *Le produit de Kronecker vérifie un certain nombre de propriétés immédiates. Il est entre autres bilinéaire et associatif.*

- $A \otimes (B + \lambda \cdot C) = (A \otimes B) + \lambda(A \otimes C)$
- $(A + \lambda \cdot B) \otimes C = (A \otimes C) + \lambda(B \otimes C)$
- $A \otimes (B \otimes C) = (A \otimes B) \otimes C$

Une propriété intéressante du produit de Kronecker est qu'il relie une matrice et sa forme vectorisée.

Proposition A.2. *Soient A, B, C des matrices de taille respectives $m \times n$, $n \times p$ et $p \times q$. Alors nous avons les propriétés suivantes.*

$$\operatorname{vec}(ABC) = (C^T \otimes A)\operatorname{vec}(B), \tag{A.1}$$

et plus particulièrement,

$$\operatorname{vec}(AB) = (Id_p \otimes A)\operatorname{vec}(B), \tag{A.2}$$

$$\operatorname{vec}(AB) = (B^T \otimes Id_m)\operatorname{vec}(A). \tag{A.3}$$

A.2 Application au calcul de la norme TV

Pour calculer la norme TV d'une image, il suffit d'appliquer un filtre sur l'image puis de sommer en valeur absolue tous les coefficients de la matrice obtenue. En revanche, lorsque la matrice nous est fournie sous forme vectorisée, comment procéder sans redimensionner la matrice pour effectuer le calcul précédent ?

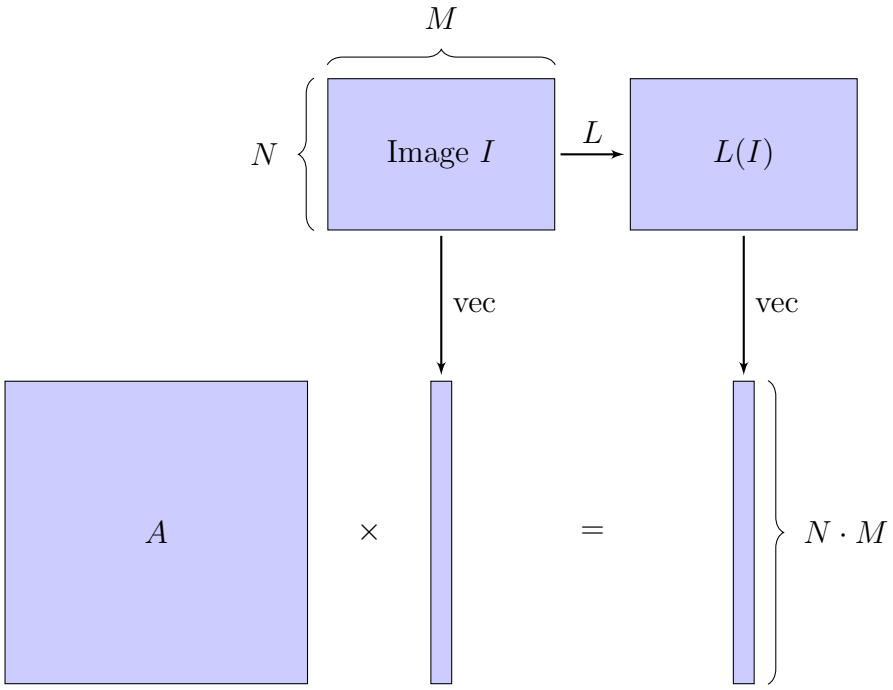


Figure A.1 – Vectorisation d'un opérateur

Le problème est de trouver l'opérateur A de la figure A.1 qui au lieu d'opérer sur une matrice opère sur les vectorisées. En revenant à la définition du gradient vertical, nous introduisons l'opérateur L_y qui calcule les gradients verticaux de l'image I. Nous définissons donc,

$$L_y(I) = I_y = TI \quad \text{où} \quad T = \begin{pmatrix} 1 & -1 & & \mathbf{(0)} & \\ & \ddots & \ddots & & \\ & & & 1 & -1 \\ \mathbf{(0)} & & & & 0 \end{pmatrix}. \tag{A.4}$$

Nous vectorisons ensuite l'égalité (A.4) en suivant la formule (A.2). Nous avons alors :

$$\mathrm{vec}(I_y) = (Id \otimes T)\,\mathrm{vec}(I).$$

L'opérateur linéaire recherché est donc $A = Id \otimes T$.

De la même manière, si nous voulons calculer les gradients horizontaux, nous calculons la matrice suivante :

$$I_x = I \begin{pmatrix} 1 & & & \\ -1 & \ddots & \mathbf{(0)} & \\ & \ddots & & \\ & & 1 & \\ \mathbf{(0)} & & -1 & 0 \end{pmatrix} \tag{A.5}$$

Ainsi nous avons,

$$\mathrm{vec}(I_x) = (T \otimes Id)\,\mathrm{vec}(I)$$

Bibliographie

[BD08] Thomas Blumensath and Mike E Davies. Gradient pursuits. *Signal Processing, IEEE Transactions on*, 56(6) :2370–2382, 2008.

[BMDG05] A. Banerjee, S. Merugu, I.S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *The Journal of Machine Learning Research*, 6 :1705–1749, 2005.

[Bre67] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming* 1. *USSR computational mathematics and mathematical physics*, 7(3) :200–217, 1967.

[CDS98] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20 :33–61, 1998.

[Çet89] A.E. Çetin. Reconstruction of signals from fourier transform samples. *Signal Processing*, 16(2) :129–148, 1989.

[Çet91] A.E. Çetin. An iterative algorithm for signal reconstruction from bispectrum. *IEEE Transactions on Signal Processing*, 39(12) :2621–2628, 1991.

[CL81] Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34(3) :321–353, 1981.

[COS09] J.F. Cai, S. Osher, and Z. Shen. Split bregman methods and frame based image restoration. *Multiscale Model. Simul*, 8(2) :337–369, 2009.

[CR05] Emmanuel Candes and Justin Romberg. l1-magic : Recovery of sparse signals via convex programming. *URL : www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 4, 2005.

[CRT06] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles : Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2) :489–509, 2006.

[CSD+08] Volkan Cevher, Aswin Sankaranarayanan, Marco F Duarte, Dikpal Reddy, Richard G Baraniuk, and Rama Chellappa. Compressive sensing for background subtraction. In *Computer Vision–ECCV 2008*, pages 155–168. Springer, 2008.

[DB09] M.F. Duarte and R.G. Baraniuk. Kronecker compressive sensing. *Submitted to IEEE Transactions on Image Processing*, 2009.

[DET04] DL Donoho, M Elad, and V Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. submitted to ieee trans. *Inform. Theory, February*, 2004.

[DMA97] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1) :57–98, 1997.

[Don95] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3) :613–627, 1995.

[Don06] D.L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4) :1289–1306, 2006.

[DTDS12] David L Donoho, Yaakov Tsaig, Iddo Drori, and J-L Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 58(2) :1094–1121, 2012.

[DZ86] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1) :116–125, 1986.

[GHL04] Sylvestre Gallot, Dominique Hulin, and Jacques Lafontaine. *Riemannian geometry*. Springer Verlag, 2004.

[GJB+07] ME Gehm, R John, DJ Brady, RM Willett, TJ Schulz, et al. Single-shot compressive spectral imaging with a dual-disperser architecture. *Opt. Express*, 15(21) :14013–14027, 2007.

[GO09] T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2 :323, 2009.

[GO11] Z. Guo and S. Osher. Template Matching via L1 Minimization and Its Application to Hyperspectral Data. *Inverse Problems and Imaging*, 5(1) :19–35, 2011.

[GWO09] Z. Guo, T. Wittman, and S. Osher. L1 unmixing and its application to hyperspectral image enhancement. *UCLA CAM report*, 2009.

[HFM04] S. Hordley, G. Finalyson, and P. Morovic. A multi-spectral image database and its application to image rendering across illumination. In *Third International Conference on Image and Graphics*, pages 394–397. IEEE, 2004.

[HYZ07] E.T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 2007.

[JGBT12] S. Jay, M. Guillaume, and J. Blanc-Talon. Underwater target detection with hyperspectral data : Solutions for both known and unknown water quality. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(4) :1213–1221, 2012.

[Kiw97] K.C. Kiwiel. Proximal minimization methods with generalized bregman functions. *SIAM Journal on Control and Optimization*, 35(4) :1142–1168, 1997.

[LSKZ11] C. Li, T. Sun, K. Kelly, and Y. Zhang. A compressive sensing and unmixing scheme for hyperspectral data processing. *Image Processing, IEEE Transactions on*, (99) :1–1, 2011.

[MZ93] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12) :3397–3415, 1993.

[NT08] D. Needell and J. A. Tropp. Cosamp : Iterative signal recovery from incomplete and inaccurate samples. Technical report, California Institute of Technology, Pasadena, 2008.

[NV09] Deanna Needell and Roman Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of computational mathematics*, 9(3) :317–334, 2009.

[OBG+06] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2) :460–489, 2006.

[PRK93] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[RHCBT10] Sylvain Rousseau, David Helbert, Philippe Carré, and Jacques Blanc-Talon. Metric tensor for multicomponent edge detection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1953–1956. IEEE, 2010.

[RHCBT13] S. Rousseau, D. Helbert, P. Carré, and J. Blanc-Talon. Compressive template matching on multispectral data. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013. IEEE.

[ROF92] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1-4) :259–268, 1992.

[Rom09] J. Romberg. Compressive sensing by random convolution. *SIAM Journal on Imaging Sciences*, 2(4) :1098–1128, 2009.

[Rud99] M. Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1) :60–72, 1999.

[SKM98] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. *IEEE Transactions on Image Processing*, 7(3) :310–318, 1998.

[SSL] S. Smand, G. Sicot, and M. Lennon. Evaluation des capacités de le télédétection hyperspectrale et développement de méthodes innovantes de traitement d'images pour des applications défense en zone littorale (hyplitt), contrat de recherche dga n°2010 34 0014, rapport d'acquisition des données, 28/03/2010.

[TAG07] Joel A. Tropp, Anna, and C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53 :4655–4666, 2007.

[WY07] Y. Wang and W. Yin. A fast algorithm for image deblurring with total variation regularization. *Image Rochester NY*, pages 1–19, 2007.

[YOGD08] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1) :143–168, 2008.

[YZ11] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1) :250–278, 2011.

[Zha09] Yin Zhang. User's guide for yall1 : Your algorithms for l1 optimization. *Technique report*, pages 09–17, 2009.