# Représentations parcimonieuses adaptées à la compression d'images

Aurélie Martin

N° d'ordre : 4117 ANNÉE 2010

**THÈSE / UNIVERSITÉ DE RENNES 1**
*sous le sceau de l'Université Européenne de Bretagne*

pour le grade de

**DOCTEUR DE L'UNIVERSITÉ DE RENNES 1**

*Mention : Traitement du signal*

**Ecole doctorale Matisse**

présentée par


# Aurélie Martin

préparée à l'unité de recherche 6074 : IRISA
Institut de recherche en informatique et systèmes aléatoires
STM


---

# Représentations parcimonieuses adaptées à la compression d'images

**Thèse soutenue à l'IRISA**
**le 2 Avril 2010**

devant le jury composé de :

**M. Olivier DEFORGES**
Professeur à l'Institut National de Sciences Appliquées de Rennes / Président

**Mme Béatrice PESQUET-POPESCU**
Professeur à Télécom ParisTech / Rapporteur

**M. Pascal FROSSARD**
Professeur à l'Ecole Polytechnique Fédérale de Lausanne / Rapporteur

**Mme Christine GUILLEMOT**
Directeur de recherche à l'INRIA / directrice de thèse

**M. Jean-Jacques FUCHS**
Professeur à l'université de Rennes 1 / Co-directeur de thèse

**M. Dominique THOREAU**
Ingénieur de recherche à Thomson / Co-directeur de thèse

*Ein unnütz Leben ist ein früher Tod.*
Goethe

# Remerciements

J’adresse tous mes remerciements à l’ensemble des personnes qui ont contribué à la réalisation et l’amélioration de mes travaux de thèse ainsi qu’à la rédaction de mon manuscrit.

Je remercie tout particulièrement ma directrice et mes co-directeurs de thèse qui m’ont suivie et soutenue au cours de ces trois années. La richesse de leur encadrement fut pour moi leur écoute, leur souci de rigueur et d’exigence allié à la confiance qu’ils m’ont portée au cours de ces années.

Je remercie ainsi ma directrice de thèse Madame Christine Guillemot, Directrice de recherche à l’INRIA et responsable du projet TEMICS, pour m’avoir offert l’opportunité de réaliser mes travaux de thèse et de m’avoir judicieusement orientée au cours de ces trois années.

Je remercie également mon co-directeur de thèse Monsieur Jean-Jacques Fuchs, Professeur de l’Université de Rennes 1 et membre de l’équipe TEMICS, pour sa grande disponibilité et pour m’avoir guidée vers de nouvelles pistes d’étude sources d’enrichissement.

Je remercie mon co-directeur de thèse Monsieur Dominique Thoreau, ingénieur de recherche à Technicolor, pour m’avoir suivie depuis mon stage de fin d’études et encouragée dans la poursuite de travaux de thèse. Je le remercie de sa présence, de son écoute attentive et des nombreux échanges fructueux que nous avons eus au cours de mes travaux.

Je remercie Monsieur Olivier Déforges, Professeur à l’Institut National de Sciences Appliquées de Rennes, qui m’a fait l’honneur de présider ce jury. Je suis reconnaissante à Madame Béatrice Pesquet-Popescu, Professeur à Télécom ParisTech, et à Monsieur Pascal Frossard, Professeur à l’Ecole Polytechnique Fédérale de Lausanne qui ont accepté d’être rapporteurs de mon manuscrit de thèse.

Je souhaite également remercier les deux équipes qui m’ont accueillie et dans lesquelles j’ai eu plaisir à travailler. L’équipe TEMICS à l’INRIA dirigée par Madame Guillemot et le laboratoire Compression de Technicolor, dirigé par Monsieur Philippe Guillotel. Les ambiances chaleureuses et décontractées m’ont permis de réaliser mes travaux dans de très bonnes conditions, soutien essentiel pour achever mes travaux de thèse.

# Table des matières

# Introduction

Il suffit de regarder autour de nous, pour constater que le monde numérique envahit petit à petit notre environnement quotidien. Les fins sont diverses : pour faciliter les échanges dans notre vie professionnelle ou pour nos loisirs. Le cinéma, la photographie, la télévision, les téléphones... Nous sommes définitivement ancrés dans l'ère du numérique. Grâce à la numérisation et à internet, on peut partager toutes nos données avec le monde entier. Ce qui est indéniable, c'est que la rapidité des échanges permis par le numérique est un véritable progrès. Un autre avantage du numérique est sa robustesse. Des données peuvent être copiées un nombre incalculable de fois, sans perdre en qualité. La richesse des contenus évolue de jour en jour, on se tourne maintenant vers des modélisations de notre environnement en trois dimensions. Même si chaque année on observe un accroissement de la capacité de stockage de nos disques durs, une nouvelle miniaturisation de composants électroniques, ou encore des fibres optiques de plus grande bande passante, l'inventivité dépasse la capacité de nos supports matériels actuels. C'est là qu'intervient la compression numérique, pour réduire la taille des données pour le stockage, le traitement ou la transmission.

On distingue à ce jour deux types de compression. La première catégorie est la compression sans perte, conduisant à une reconstruction *parfaite*. Ce type de compression est notamment utilisé dans des domaines où l'on souhaite comprimer des informations, sans toutefois introduire une quelconque erreur sur des données hautement sensibles. Par exemple, des données médicales sont trop précieuses à l'établissement d'un diagnostique, pour autoriser un traitement qui serait susceptible de modifier les conclusions du corps médical. D'un point de vue plus général, la compression sans perte ne permet pas d'atteindre des taux de compression suffisamment élevés, au vu des nouveaux contenus de plus en plus riches. Ainsi les efforts actuels portent plus spécifiquement sur l'élaboration de nouvelles techniques en compression d'images *avec* pertes. La différence majeure, en comparaison avec la première catégorie, est que l'on ne cherche plus à obtenir une reconstruction identique à la source. On tolère ainsi une approximation qui n'induit pas de dégradations visuellement perceptibles. Bien évidemment, tout le savoir-faire réside dans la maîtrise de ces pertes.

L'enjeu de cette compression avec pertes est de rechercher le meilleur compromis entre la qualité de l'image reconstruite et le taux de compression atteint. L'ingéniosité des systèmes actuels réside notamment, dans la nature même des données qui sont effectivement encodées. Les modèles les plus performants se basent sur le codage prédictif. Une image, dite de prédiction, que l'on veut la plus semblable à la source, est générée, selon diverses méthodes, puis est soustraite à l'image originale, pour ainsi former une image résiduelle. Ce sont ces données qui sont ensuite encodées. Bien moins coûteuses qu'une image complète, ces informations suffisent à recréer l'image, après diverses opérations connues de décodage.

Parmi les techniques permettant de générer cette image de prédiction, il existe la prédiction spatiale, plus couramment appelée intra-image. Dans ce cas, le codeur génère l'image de prédiction

en se basant sur un voisinage connu de texture, au sein de cette image. Les performances de compression sont ainsi directement liées à la qualité de cette extrapolation. Plus les algorithmes de prédiction sont efficaces, moins il y aura de données contenues dans l'image résiduelle et meilleurs seront les taux de compression. C'est le cas par exemple, des modes de prédiction directionnelle utilisée dans H.264 / AVC. Bien qu'offrant de bonnes performances, ces méthodes ne permettent pas de prédire des textures complexes. Il apparait que la prédiction se heurte à de nombreuses problématiques, propres à la modélisation des textures. Il reste assez complexe de définir mathématiquement ce qu'est une texture. Il n'en demeure pas moins que ce type de signal s'apparente, dans de nombreux cas, à une superposition évidente de structures géométriques ou ondulatoires.

L'analyse des textures est un vaste domaine dont le but est de cerner la nature propre d'une texture, soit via des algorithmes de classification, soit via des algorithmes de synthèse visant la création d'une texture, visuellement *similaire* à la texture originale. Ces dernières années, des travaux du domaine se sont orientés vers l'analyse harmonique des signaux et l'extraction de leurs composantes principales. A l'origine, la décomposition harmonique était réservée à des signaux de natures différentes, et plus spécifiquement, des données audio. Appliquée à l'image, l'analyse harmonique offre de nouvelles perspectives.

Cette thèse se concentre sur l'étude des représentations parcimonieuses pour la prédiction d'images, outil prometteur pour l'approximation et la modélisation de textures. Elles sont en effet utilisées pour des applications d'*inpainting*, qui s'avère être un problème de synthèse de texture, tout comme la prédiction. Les représentations parcimonieuses ont connu un véritable essor ces dernières années. Elles permettent d'obtenir une bonne approximation du signal de texture, sous une forme compactée. L'objectif des représentations parcimonieuses est d'observer un signal original dans un autre domaine et de distinguer, pour ce domaine, quelles sont les composantes significatives du signal. On obtient ainsi une représentation de ce signal composée d'un faible nombre de coefficients, qui suffisent à le définir complètement, pour une erreur d'approximation tolérée.

Les principales contributions de cette thèse portent sur les thèmes suivants.

- Les contributions de cette thèse portent tout d'abord sur une méthode originale de prédiction spatiale de texture basée sur les représentations parcimonieuses. Nous étudions les performances de cette approche dans un schéma de compression de type H.264 / AVC. ;
- Nous étendons la méthode au cas des prédictions inter-couches dans un schéma de compression vidéo scalable, basé sur l'encodeur H.264 / SVC ;
- Nous montrons que l'approche permet également de débruiter et de raffiner le signal obtenu par prédiction inter-images et ainsi d'améliorer les performances de la-dite prédiction ;
- Nous établissons ensuite le parallèle théorique entre une méthode basée déconvolution spectrale et la méthode de prédiction basée représentations parcimonieuses, dans le cas mono-dimensionnel. Nous avons également comparé les performances de chacune des approches dans le cadre de la prédiction spatiale de texture ;
- Nous étudions enfin quelques pistes permettant d'améliorer ou d'enrichir le dictionnaire, par le biais notamment d'atomes spatiaux texturés ou encore via un recalage de la phase spatiale des atomes, permettant ainsi de mieux s'adapter au signal de texture source.

La suite du manuscrit est structurée comme suit.

**Chapitre 1** Le chapitre 1 présente les principaux concepts et principes inhérents à la compression vidéo avec pertes dont tout d'abord, les comportements sensoriels du système visuel humain qui ont guidé certains traitements effectués tout au long de la chaine de compression. On comprend tout naturellement que l'introduction des pertes, évoquées précédemment, sera privilégiée au niveau des caractéristiques du signal, auxquelles notre oeil lui-même est moins sensible, afin de minimiser l'impact de cette dégradation au niveau de la qualité subjective d'une image. Les notions théoriques de base, indispensables pour comprendre l'enchainement des traitements appliqués au signal en vue de sa compression, sont ensuite évoquées. Les domaines concernés sont la transformation, la quantification, la gestion du mouvement dans une vidéo et le codage entropique. La présentation de ces concepts ne serait pas complète sans la description d'une norme de compression actuelle. Nous présentons ainsi les spécificités liées à la norme H.264 / MPEG 4 - AVC, norme de référence utilisée dans les travaux de cette thèse.

**Chapitre 2** Ce chapitre est un court état de l'art sur les techniques actuelles en synthèse de texture. Les problématiques rencontrées en synthèse sont très voisines de celles soulevées par l'extrapolation de texture. Ce rapide tour d'horizon permet de mieux comprendre la démarche au coeur de cette thèse, qui a été l'utilisation des représentations parcimonieuses dans un contexte de prédiction d'images. Les principaux travaux en synthèse cherchent à recréer un signal texturé, à partir d'un échantillon connu. Encore une fois, la démarche consiste à générer une texture visuellement proche de l'originale et non une texture strictement identique, au sens mathématique du terme. Plusieurs méthodes existent : certaines recréent le motif, en disposant les pixels selon des lois de probabilités apprises sur le motif original ; d'autres recopient par blocs de pixels entier, le motif initial, tout en assurant par divers procédés une continuité entre ces blocs ; ou alors, certaines techniques s'attachent à reformer une texture similaire, pixel à pixel. Plus récemment, certaines approches, proposant un angle de vue légèrement différent, ont vu le jour. Ces travaux suggèrent de décomposer l'image dans un autre espace, où l'on recherche des correspondances avec des fonctions de base connues. L'avantage de telles méthodes est la flexibilité inhérente aux fonctions choisies pour former cet espace. Par exemple, cela permet de propager simultanément la géométrie et la texture d'une image. En choisissant habilement des fonctions de base représentatives de contours et d'autres à même de représenter une pure texture, on réussit à recréer toute la complexité du signal original. Les travaux de cette thèse ont pour vocation d'exploiter ces méthodes de décomposition au sein d'un codeur.

**Chapitre 3** Les fondements mathématiques liés aux représentations parcimonieuses sont exposés dans ce chapitre. Nous verrons par la suite que nous pouvons poser le problème de l'extrapolation de texture sous la forme d'une recherche de la meilleure combinaison linéaire pondérée. Le but des représentations parcimonieuses est de réduire au maximum le nombre de fonctions de base participant à la représentation. Cette problématique se formule sous la forme d'un problème d'optimisation où l'on cherche à minimiser le nombre de coefficients de pondération non nuls. Cette formulation n'ayant qu'un intérêt purement théorique, des propositions ont été faites pour relâcher les contraintes et permettre ainsi la résolution du problème. Ce chapitre présente également certaines des solutions algorithmiques qui ont été développées pour résoudre ce type de problème.

**Chapitre 4** Nous exposons dans ce chapitre une nouvelle méthode de prédiction spatiale de texture, basée sur les représentations parcimonieuses. Cette méthode a été intégrée pour être évaluée dans un schéma de compression. La motivation de ces travaux est d'améliorer la prédiction de zones texturées complexes. La méthode permet par nature de générer des signaux complexes bi-dimensionnels. Il faut néanmoins garder à l'esprit que le but est d'améliorer la compression.

La meilleure prédiction obtenue, quand bien même visuellement très proche du signal original, ne conduira pas nécessairement à un taux de compression plus faible. Les méthodes de prédiction intra-image de la norme H.264 / AVC forment des signaux uniquement mono-directionnels mais ces méthodes permettent cependant de mener à des taux de compression, qui actuellement sont les meilleurs dans le domaine. Il y a donc un réel défi, qui consiste à réussir à modéliser et étendre des signaux de texture plus complexes, tout en cherchant à minimiser le coût de codage. Nous exposons dans ce chapitre les résultats obtenus tenant compte de la qualité visuelle de la texture générée et également des performances en terme de débit et de distorsion. Afin d'étendre notre approche, nous évaluons aussi les résultats obtenus dans un cadre de débruitage ou de raffinement. Plus particulièrement, nous nous sommes intéressés à l'amélioration de la prédiction spatiale inter-couches de H.264 / SVC. Les contraintes sont différentes mais notre approche ouvre des perspectives intéressantes pour d'autres applications.

**Chapitre 5** Le chapitre 5 fait le parallèle théorique entre la méthode de prédiction basée sur les représentations parcimonieuses décrite au chapitre précédent, et une méthode de la littérature basée sur une déconvolution spectrale. Cette méthode s'avère être un cas particulier de la méthode basée représentations parcimonieuses que nous avons proposée, dans le cas des signaux mono-dimensionnels et lorsque les fonctions de base du dictionnaire sont issues des bases de Fourier complexe. Introduit au sein du processus de prédiction intra-image d'un encodeur H.264 / AVC, cette technique révèle avoir des performances comparable à la prédiction basée représentations parcimonieuses. L'avantage de la prédiction basée déconvolution spectrale réside dans sa rapidité d'exécution, ainsi que de la richesse des fonctions de base, dans le cas d'un dictionnaire issu de la base de Fourier complexe. Néanmoins la prédiction basée représentations parcimonieuses offre un avantage certain lié à la liberté de choix des fonctions de base du dictionnaire.

**Chapitre 6** Le choix des fonctions proposées pour modéliser le signal texturé est un des points majeurs de la technique d'extrapolation, basée sur les représentations parcimonieuses. Il existe dans la littérature des méthodes d'apprentissage de dictionnaire, à partir d'un ensemble d'images suffisamment représentatif. Il exite également des méthodes qui adaptent le dictionnaire au fil de l'eau, directement sur l'image traitée. Néanmoins ce n'est pas l'approche que nous avons privilégié. Dans ce chapitre, nous proposons des méthodes alternatives basées sur une adaptation locale des fonctions de base aux caractéristiques du signal. Nous avons ainsi exploré une technique visant à recaler la phase des atomes pour s'adapter au mieux à la phase spatiale de la texture étudiée. Nous avons également choisi de former un dictionnaire contenant des patchs de petites tailles de l'image source elle-même. L'objectif est d'améliorer la parcimonie de la représentation obtenue à l'aide de ces atomes spatiaux, ainsi que la prédiction intra-image. Ces premières pistes explorent des solutions pour adapter localement les fonctions de base au signal. Nous avons également étudié la méthode duale visant, cette fois ci, à adapter le signal de texture aux fonctions de base. Le principe a été d'orienter les pixels du voisinage texturé le long d'orientations privilégiées, afin d'avoir à utiliser un minimum de fonctions de base, relativement simples. Ce chapitre présente en outre l'apport obtenu par l'ajout d'atomes directionnels, facilitant la représentation des contours et des angles.

Le manuscrit conclut finalement par un résumé des principales contributions de la thèse et présente quelques perspectives.

# Chapitre 1

# Introduction à la compression vidéo

Les bases théoriques à l'origine de la notion de compression d'une source d'information furent établies par Claude Shannon dans les années cinquante. Fondateur de la théorie de l'information et précurseur dans le domaine des télécommunications, Shannon introduisit avec cette théorie un modèle mathématique visant notamment à évaluer le coût de l'information et les contraintes engendrées par sa transmission sur des canaux physiques. Par la suite, des outils de compression sans pertes découlèrent de cette théorie. Nous pouvons à ce titre citer le codage de Huffman, le codage arithmétique ou encore le codage de Lempel et Ziv. Malheureusement, ces techniques ne répondent qu'en partie aux besoins de stockage et de diffusion des médias actuels. Les taux de compression qu'offrent ces techniques demeurent toujours trop insuffisants. C'est en se tournant tout naturellement vers de nouveaux schémas, que la compression est devenue aujourd'hui un outil indispensable pour les télécommunications. Deux techniques fondamentales permirent d'atteindre des taux de compression largement supérieurs. Il s'agit de l'exploitation des corrélations au sein du signal et de l'introduction de *pertes* dans le processus de codage. Nous présenterons dans ce chapitre les blocs fonctionnels clés d'un encodeur vidéo numérique : les différentes étapes de décorrélation du signal d'une part au sein d'une image via des transformations et d'autre part entre les images via les étapes de prédiction temporelle par compensation de mouvement. Ce chapitre décrit ensuite l'étape de quantification qui permet de réduire le débit au prix de distorsions introduites sur le signal et enfin la phase de codage qui précède la transmission de l'information.

## 1.1 Représentation des couleurs

Pour représenter une couleur, une première approche consiste à estimer l'ensemble des composantes spectrales, en filtrant le signal de couleur par de nombreux filtres à bande étroite. L'intensité de chaque sortie filtrée donnerait une estimation des longueurs d'onde présentes dans le spectre. Seulement cela nécessite d'utiliser un trop grand nombre de capteurs. Une autre possibilité de représentation consiste à utiliser un espace de couleurs à $k$ dimensions. Des expériences psychovisuelles d'égalisation [KGK93] ont montré qu'en combinant trois stimuli de longueurs d'onde particulières, il est possible de synthétiser presque toutes les couleurs existantes.

L'espace de couleurs RGB (*Red Green Blue*) est fondé sur les trois couleurs monochromatiques suivantes : rouge $R$ (700 $nm$), vert $V$ (546 $nm$) et bleu $B$ (435.8 $nm$). On définit également pour chacune des couleurs, la luminance : elle permet d'éclaircir ou d'assombrir une couleur en ajustant la quantité de noir. Par exemple, en modulant du minimum au maximum la luminance, on passe

du noir au blanc en générant toutes les teintes de gris.

En plus du système RGB, il existe d'autres systèmes de couleurs tels les systèmes *YUV* ou *YCbCr*. Ces systèmes exploitent le fait que le cerveau traduit le signal trichromatique perçu par l'oeil, comme un signal composé de trois composantes, dont l'une est achromatique : la luminance. La compression d'images et de vidéo exploite ce principe. Ainsi la grande majorité des contenus en compression sont représentés dans la base *YUV* obtenue par la transformation (1.1) suivante :

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1.1}$$

## 1.2 Perception visuelle

La perception d'une image se rattache aux caractéristiques du système visuel humain. L'oeil est sensible à certaines fréquences du spectre électromagnétique. Ces fréquences représentent la lumière visible. La représentation d'une image consiste à transformer, par exemple, cette entité physique sous une forme numérique. Cette représentation est primordiale car elle conditionne la mise en place des traitements à effectuer pour compresser une image. L'oeil est un système complexe. La lumière incidente est réfractée par la cornée et dirigée vers la pupille. La pupille est l'ouverture de l'iris par laquelle la lumière pénètre dans l'oeil. La lumière est ensuite réfractée une seconde fois en direction du fond de l'oeil, sur la rétine. Cette dernière est composée d'une série de récepteurs photosensibles, reliés à des cellules qui transmettent des signaux au nerf optique. Les signaux sont ensuite acheminés vers le cerveau pour y être analysés.

La lumière correspond à une partie du spectre d'énergie électromagnétique, constitué de plusieurs longueurs d'onde $\lambda$ (ou fréquences), mesurées en nanomètres (*nm*). Des dispositifs dispersifs, un prisme par exemple, permettent de séparer ces fréquences pour former le spectre de la lumière visible que nous connaissons, représenté ici en figure 1.1.

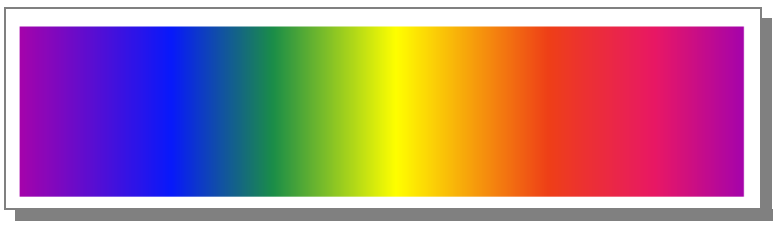


FIG. 1.1 – Spectre continu de la lumière visible

C'est le mélange des longueurs d'onde qui produit une sensation de couleur. La lumière est donc une distribution d'énergie émise à certaines fréquences ayant une certaine intensité. Pour caractériser une couleur monochromatique, il suffit de connaître sa longueur d'onde $\lambda$ et la **luminance** $L$, expression qualitative de la brillance énergétique. Dans le cas d'une source émettant plusieurs radiations de longueurs d'onde différentes, il se pose le problème de la résultante, pour le récepteur visuel, de l'addition de l'ensemble de ces radiations.

## 1.3 Effets visuels exploités en compression

### 1.3.1 Masquage entre composantes chromatiques

Dans ce nouvel espace *YUV*, les composantes chromatiques, $U$ et $V$, contiennent beaucoup moins d'information que la composante de luminance $Y$. La compression exploite en outre le

fait que le système visuel humain est moins sensible aux composantes chromatiques $U$ et $V$ qu'à la luminance, en sous-échantillonnant horizontalement et verticalement les composantes chromatiques avant leur compression.

Plusieurs formats ont ainsi été définis :

- Le format $4:4:4$ est le format de base où les composantes chromatiques n'on pas été sous-échantillonnées.
- Le format $4:2:2$ où les composantes $U$ et $V$ ont été échantillonnées d'un facteur deux verticalement.
- Le format $4:2:0$ où les composantes $U$ et $V$ ont été échantillonnées d'un facteur deux verticalement et horizontalement.
- Le format $4:0:0$ ne contient aucune information chromatique. Il s'agit donc d'une image en niveaux de gris ; seule la luminance $Y$ est conservée.

La figure 1.2 donne une illustration de ces différents formats.

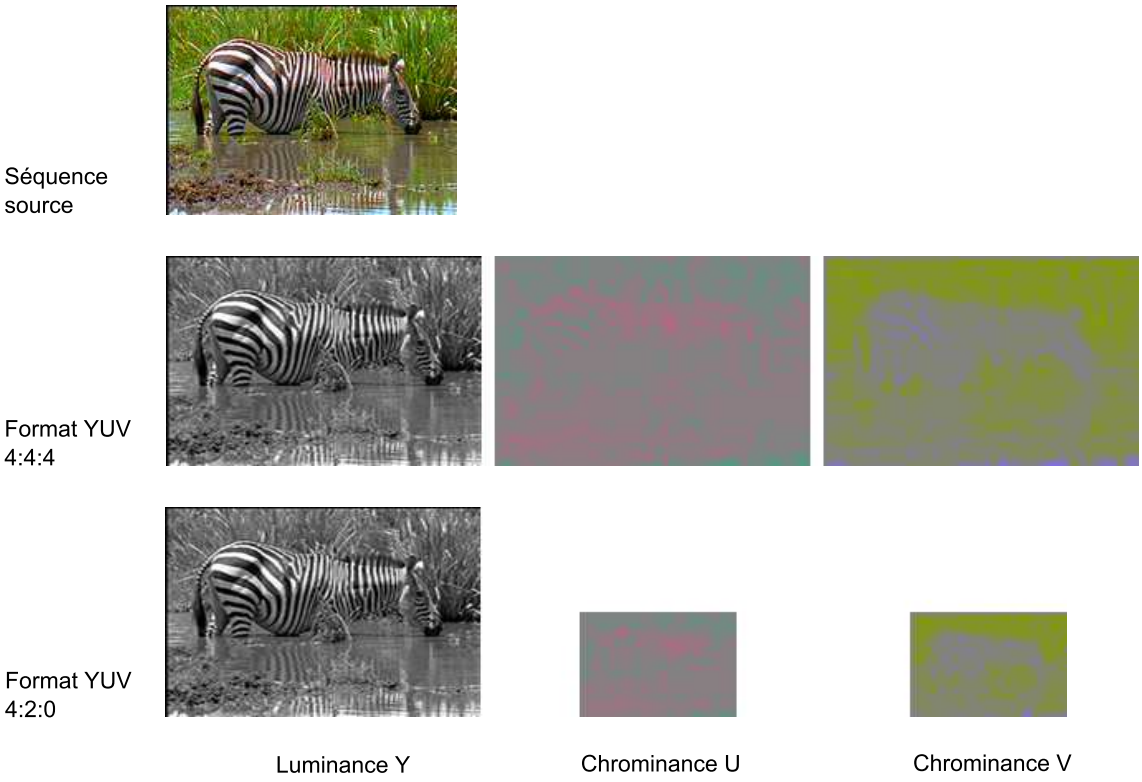



FIG. 1.2 – Formats d'images *YUV*

Un phénomène plus difficilement exploitable, car plus difficile à modéliser, concerne les interactions entre les composantes chromatiques et achromatique [Bar99]. Il s'avère en effet que le masquage spatial peut être accentué ou réduit sur une composante selon les autres composantes. Les interactions de la composante achromatique sur les composantes chromatiques sont les plus prononcées. Elles se traduisent notamment par le fait que des dégradations non perceptibles sur une image monochrome le deviennent sur une image couleur.

### 1.3.2 Notions de masquage et de facilitation

Le signal tri-dimensionnel (deux dimensions spatiales et une dimension temporelle) que constitue une séquence vidéo n'est pas perçu par le système visuel humain de façon décorrélée entre ses différentes composantes spatiales, fréquentielles et temporelles.

En effet, on a pu démontrer l'existence de multiples interactions entre ces différentes composantes. Il apparaît en particulier deux phénomènes essentiels : le phénomène de facilitation de perception et le phénomène de masquage correspondant à une diminution de la perception. Ces deux phénomènes sont relatifs à la perception d'un stimulus se juxtaposant à un signal dit masquant. S'il est nécessaire de renforcer l'amplitude de ce stimulus pour pouvoir le percevoir, on parlera d'effet de masquage. Au contraire, s'il s'avère possible de diminuer cette amplitude, on parlera d'effet de facilitation, ce qui signifie une augmentation des capacités de perception.

Le processus de codage va chercher à exploiter ces propriétés de facilitation et de masquage. Une des stratégies sera notamment de supprimer l'information masquée ou de générer de préférence les erreurs dues à la compression le plus possible dans les zones de masquage et le moins possible dans les zones de facilitation.

### 1.3.3 Sensibilité au contraste

Les phénomènes de facilitation et de masquage interviennent en premier lieu sur la composante de luminance. Des expériences ont montré que le système visuel humain présentait une sensibilité plus importante au contraste pour des plages de luminance moyenne. On définit le contraste $C$ de la manière suivante :

$$C = \frac{L_{max} - L_{min}}{L_{moy}}$$

où $L_{min}$, $L_{max}$ et $L_{moy}$ sont les valeurs de luminance minimum, maximum et moyenne du signal.

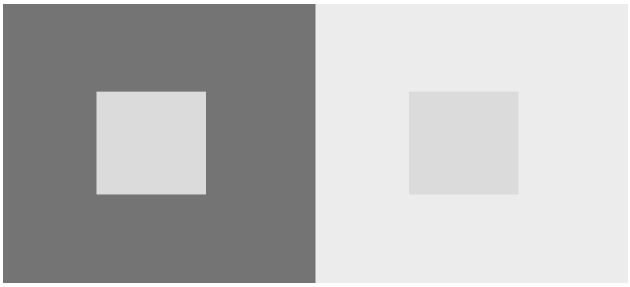


FIG. 1.3 – Effet de contraste simultané : les carrés centraux sont de même luminance mais sont perçus différemment à cause de l'intensité du fond

En d'autres termes, si l'on superpose un signal $A$ sur un signal masquant $B$, on percevra plus facilement $A$ lorsque $B$ est de luminance moyenne que lorsque $B$ est de luminance proche de 0 (noir) ou élevée (blanc) (cf. figure 1.3). Il y a donc un phénomène de facilitation pour des luminances moyennes et de masquage pour des plages de luminance extrêmes. On pourra exploiter ce phénomène en quantifiant plus finement les plages d'intensité moyenne.

### 1.3.4 Sensibilité statique à la fréquence spatiale

La notion de fréquence spatiale caractérise la rapidité de variation du signal à l'intérieur d'une image. Une zone uniforme correspond à une fréquence spatiale faible, alors qu'une zone très riche en texture (par exemple un motif se répétant très fréquemment) correspond à une fréquence

spatiale élevée. Le système visuel humain présente des sensibilités différentes au contraste selon la fréquence spatiale du signal auquel il est soumis. Cette sensibilité croît jusqu'à atteindre un pic, puis décroît ensuite rapidement avec la fréquence [CR68, MS74]. En termes pratiques, cela signifie que des zones très texturées sont moins bien perçues que des zones relativement uniformes. Dans le contexte de la compression, cette propriété peut être exploitée par une compression plus prononcée des composantes haute fréquence spatiale du signal vidéo.

### 1.3.5 Variation de la sensibilité en fonction du mouvement entre images

S'ajoute à cette dépendance de la sensibilité à la fréquence spatiale du signal, une dépendance au mouvement entre les images. Un modèle d'interaction mouvement-fréquence spatiale, obtenu à partir d'évaluations expérimentales, a été proposé dans [Kel79]. Ces travaux font apparaître que le pic de sensibilité se décale d'autant plus vers les fréquences spatiales basses que le mouvement est important. Pour des mouvements de très grande amplitude, on ne perçoit que les fréquences spatiales basses du signal vidéo. Cela signifie que l'on va pouvoir accentuer la compression des composantes haute fréquence spatiale des images sur les zones à fort mouvement.

## 1.4 Compression d'images fixes

Un système de compression vidéo est un ensemble d'outils permettant de transformer un signal numérique en un train binaire ou *bitstream*, moins volumineux, permettant ainsi d'être stocké ou transmis plus facilement. Nous présentons dans cette section les étapes clés qui permettent d'atteindre cet objectif. La figure 1.4 présente les trois principaux traitements à appliquer au signal, en vue de le comprimer :

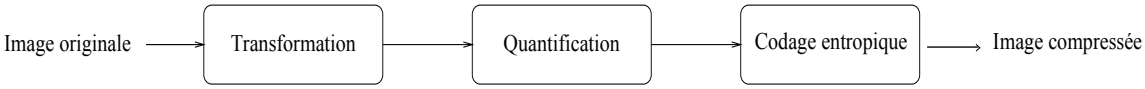



FIG. 1.4 – Schéma de principe d'un système de compression d'images

La première étape d'un schéma de compression d'images fixes cherche à décorréler le signal. La décorrélation est le traitement qui vise à réduire, voire supprimer les redondances contenues dans le signal. Les deux techniques utilisées pour décorréler les données sont le codage prédictif et les transformations.

Le codage prédictif consiste à évaluer la valeur d'un pixel courant à partir de son contexte : les pixels déjà reconstruits. La technique reposant sur une estimation de la nouvelle valeur, on introduit nécessairement une erreur. Cette erreur de prédiction se mesure par le calcul de la différence, pixel à pixel, entre la valeur source et la valeur prédite. C'est cette erreur de prédiction qui sera ensuite codée puis transmise. La quantité d'information codée est ainsi largement inférieure par rapport à un système de codage dépourvu du codage prédictif, qui code directement les données source. Le succès de la technique est par conséquent fortement conditionné à la qualité du prédicteur.

Le codage par transformée parachève le travail de décorrélation opéré en amont, grâce au codage prédictif. La transformation consiste à projeter un signal dans un autre domaine, de façon à obtenir une version plus compacte de l'information. Le signal transformé a en effet son énergie concentrée sur un nombre restreint de coefficients. Qui plus est, cela permet de distinguer les coefficients les plus significatifs de ceux qui le sont moins. Les processus ultérieurs pourront ainsi facilement éliminer les coefficients négligeables, sans pour autant dégrader sensiblement la qualité du signal reconstruit. Il est à noter que les transformées n'induisent pas directement de

compression au signal. Ce sont avant tout des outils pour le modéliser de façon propice à ce que d'autres outils puissent être appliqués pour réaliser la compression.

Appliquer la transformée directement sur toute l'image induirait un coût de calcul beaucoup trop élevé. De plus, parler de corrélations implique de travailler avec un signal qui soit stationnaire. Dans une image naturelle, il est illusoire de garantir une stationnarité à grande échelle. Il est donc nécessaire d'appliquer la transformation sur des sous-ensembles de l'image et plus précisément des blocs de pixels de faibles dimensions.

### 1.4.1 Vers un compromis : la transformée en cosinus discrète

Plusieurs transformées peuvent être utilisées. Notons que les transformations utilisées sont des transformations orthogonales afin de ne pas obtenir un signal transformé d'énergie supérieure au signal source. On peut citer la transformée de Karhunen-Loève (KLT, pour *Karhunen-Loève Transform*), la transformée discrète de Fourier (DFT pour *Discrete Fourier Transform*) ou encore la transformée discrète en cosinus (DCT pour *Discrete Cosine Transform*).

*– La transformée de Karhunen-Loève –*

Historiquement, la première transformée envisagée fut la KLT. Cette transformée consiste à rechercher les composantes principales du signal, en diagonalisant sa matrice d'autocorrélation. C'est une transformation optimale car elle permet de condenser au mieux le signal sur des composantes localisées en fréquence. Malgré son optimalité, cette transformation est peu usitée car trop complexe à mettre en oeuvre. Un autre inconvénient de cette transformée est d'être dépendante des données source puisqu'elle repose sur le calcul de l'autocorrélation du signal d'entrée.

*– La transformée de Fourier –*

La transformée de Fourier offre une bonne localisation fréquentielle de l'information et s'implémente facilement via des algorithmes rapides (FFT pour *Fast Fourier Transform*). L'expression de la DFT bi-dimensionnelle pour une image de taille $N \times N$ est donnée par la relation (1.2) ci-dessous :

$$F(k,l) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-2i\pi\left(\frac{km}{N} + \frac{ln}{N}\right)\right] \tag{1.2}$$

Un inconvénient de la DFT est de générer un signal transformé complexe, difficilement manipulable pour la suite des traitements. Un autre inconvénient est lié à la symétrisation du spectre qui induit très rapidement des effets de bord au cours des traitements ultérieurement appliqués.

*– La transformée en cosinus discrète –*

La DCT est la transformation par excellence dans le contexte du traitement d'images et de la compression vidéo. C'est la transformation qui est de loin la plus utilisée actuellement dans les normes de compression [1]. L'expression de la DCT est la suivante :

[1] Le standard JPEG2000 se distingue quant à lui par l'utilisation d'une transformée en ondelettes.

$$F(k,l) = \frac{2}{N} a(k)a(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cos\left[\frac{(2n+1)l\pi}{2N}\right] \tag{1.3}$$

$$\text{où} \quad a(0) = \frac{\sqrt{2}}{2} \quad \text{et pour} \quad k \neq 0 \quad a(k) = 1 \tag{1.4}$$

Elle est préférentiellement choisie car elle offre de nombreux atouts : elle présente une bonne localisation fréquentielle, une compaction de l'énergie qui surpasse celle obtenue avec la DFT, elle est facile d'implémentation et a notamment l'avantage de générer un signal transformé réel. Par ailleurs, c'est la DCT qui se rapproche le plus souvent de la KLT. Voici présentées en figure 1.5, les fonctions de base bi-dimensionnelles de la DCT. On peut remarquer que la première fonction de base calculée, située en haut à gauche dans cette image, correspond à une moyenne ce qui se visualise par une image uniforme. Les autres fonctions correspondent aux variations du cosinus, pour différentes valeurs de fréquences spatiales.

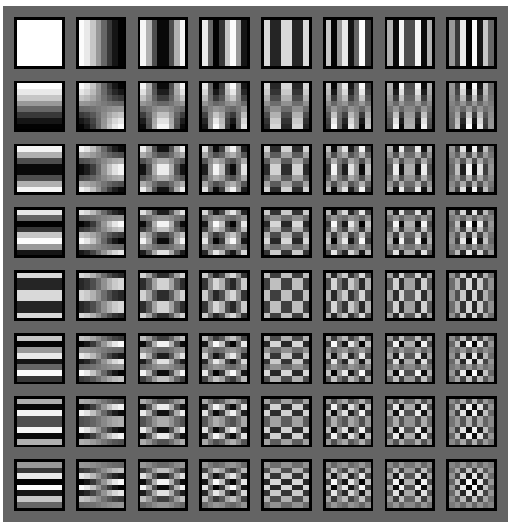


FIG. 1.5 – Fonctions de base de la transformée en cosinus discrète 2D de taille $8 \times 8$

### 1.4.2 Quantification

Quantifier un signal consiste à réduire sa précision en le discrétisant en quantités discontinues ou quanta. Un signal prenant valeur dans un alphabet de $n$ valeurs sera représenté par un nombre inférieur $m$ de valeurs. Le signal ainsi quantifié pourra être décrit avec moins de bits que la version initiale. Il existe donc, de fait, une erreur de quantification : on introduit des *pertes*, ce qui rend ainsi le processus irréversible. On distingue plusieurs types de quantification : la quantification scalaire, vectorielle et la quantification en treillis. Nous ne décrirons ici que la quantification scalaire, qui est celle utilisée dans un encodeur de type H.264 / AVC.

#### 1.4.2.1 Quantification scalaire uniforme

Soit $X$ un signal dont les valeurs sont comprises dans l'intervalle $[a, b]$ ; on note $p(x)$ la densité de probabilité associée. La quantification scalaire quantifie chaque élément $x_i$ de $X$

indépendamment des autres composantes $x_j, \forall j \neq i$. On définit l'ensemble des intervalles de quantification par $[a_i, a_{i+1}]$ pour $i \in [0, N-1]$ et avec $a_0 = a$ et $a_N = b$. Notons également $Y = \{y_0, ..., y_{N-1}\}$ l'ensemble des valeurs quantifiées possibles, appelé ensemble des niveaux de reconstruction, chacun associé à son intervalle de quantification. L'opération de reconstruction consiste à chercher l'échantillon $y_i$ le plus proche de la valeur de $x_i$ dans l'ensemble $Y$. Soit la valeur quantifiée $Q(x_i)$ :

$$Q(x_i) = \hat{x_i} = y_i \quad \text{si} \quad x_i \in [a_i, a_{i+1}] \qquad \text{avec} \quad i \in [0, N-1]$$

Les niveaux de reconstruction $y_i$ sont les espérances des valeurs connues en entrée, pour un intervalle de quantification :

$$y_i = \mathbb{E}_{p(x)}\left(x | x \in [a_i, a_{i+1}]\right)$$

On parle de quantification uniforme car tous les intervalles ont la même dimension. On note généralement $\Delta = a_{i+1} - a_i$, la longueur de cet intervalle, aussi appelé le *pas de quantification*.

#### 1.4.2.2 Quantification scalaire à zone morte

Il s'agit d'un quantificateur uniforme avec une légère modification. Les intervalles du quantificateur scalaire sont, comme nous venons de le voir, tous égaux à $\Delta$. C'est la définition même de l'uniformité. Dans le cas du quantificateur avec zone morte, on impose d'avoir un intervalle supérieur à $\Delta$ autour de la valeur $x = 0$. En général, la longueur choisie correspond à un multiple de $\Delta$ (généralement : $2\Delta$).

Ce qui a motivé la mise en place de cette zone morte est directement lié aux caractéristiques, déjà évoquées précédemment, du système visuel humain. Son utilisation trouve tout son sens lorsque la distribution du signal source est centrée en zéro. Généralement les coefficients de hautes fréquences sont de faible amplitude. On décide ainsi de négliger automatiquement ces fréquences, trop pénalisantes pour le processus, sachant que leur retrait n'induira pas de dégradation perceptible pour l'oeil humain.

### 1.4.3 Codage

Le codage consiste à passer d'une représentation du signal à une autre. Soit $X = \{x_i\}_{i \in [0,N-1]}$ une source d'information. Coder $X$ revient à trouver pour chaque élément $x_i$ un mot de code $b_i$. Pour mesurer le nombre de bits nécessaire, on définit l'entropie d'une source d'information. L'entropie de $X$, notée $H(X)$, est définie comme la limite inférieure du nombre moyen de bits par symbole nécessaire pour coder $X$ *sans perte d'information*. L'entropie est définie comme étant l'espérance de la quantité d'information des observations de la source, notée $I_X$ :

$$H(X) = \mathbb{E}_X\left(I_X\right)$$

avec $I_X(x) = -\log_2\left(P(X = x)\right)$ la quantité d'information et $P(X = x)$ la loi de probabilité des événements de la source. Soit :

$$H(X) = -\sum_{i=0}^{N-1} P(x_i) \log_2\left(P(x_i)\right)$$

Un **codeur entropique** a pour spécificité de générer des mots de code $b_i$ de longueur moyenne proche de l'entropie de la source. Afin de parvenir à ce résultat, on exploite les propriétés statistiques du signal source que l'on souhaite coder. La distribution du signal n'étant pas

nécessairement uniforme, on choisit d'attribuer un mot de code le plus court possible aux symboles qui apparaissent le plus fréquemment. On parle ainsi de codes à longueur variable (les codes VLC, pour *Variable Length Code*) dont le but est de minimiser le coût de codage. Parmi les codes les plus connus, nous pouvons citer le codage de Huffman, le codage arithmétique ou encore l'UVLC pour *Universal Variable Length Code*. Le codage entropique actuellement le plus utilisé est le codage arithmétique. Il a la particularité d'attribuer un code au message tout entier plutôt que symbole à symbole, ce qui lui permet de coder l'information sur une valeur fractionnaire de bit. Nous ne donnerons pas ici le descriptif détaillé de tous ces codes. Nous nous attacherons cependant, en section 1.6.2.3, à présenter le codage entropique spécifique à H.264 / AVC : le CABAC (*Context Adaptive Binary Arithmetic Coding*), basé sur des modèles de sources et des lois de probabilités. Comme codeur contextuel basé sur un codage arithmétique, nous pouvons également citer le codeur EBCOT (*Embedded Block Coding with Optimal Truncation Points*), utilisé dans la norme de compression JPEG2000 et adapté à une décomposition en ondelettes.

## 1.5 Compression d'images animées

### 1.5.1 Exploitation de la corrélation temporelle

Un codeur vidéo classique utilise quatre outils principaux : la prédiction, les transformations, la quantification et le codage. Nous avons exposé précédemment dans la section 1.4 les principes de ces outils valables pour les images fixes mais aussi pour des images animées. Nous allons cependant détailler ici comment est exploitée la dimension temporelle et plus exactement, la stratégie mise en place pour exploiter les redondances temporelles.

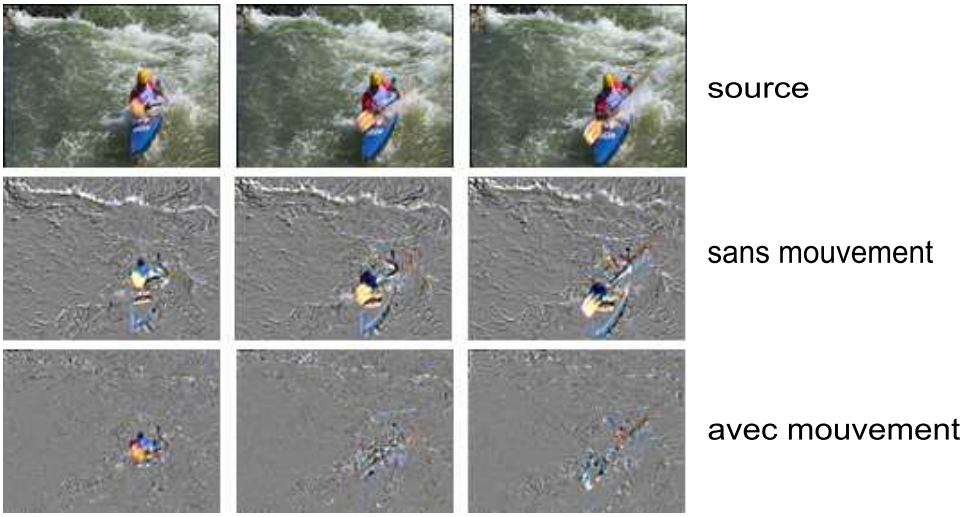



FIG. 1.6 – Exemple d'images résiduelles avec et sans compensation de mouvement

Dans la section 1.4, la prédiction présentée était **spatiale** : on estimait la valeur d'un pixel à partir des pixels voisins de la même image. La transposition à la dimension temporelle implique de chercher des corrélations entre images successives. Ainsi, la prédiction **temporelle** consiste à estimer un pixel à partir de pixels présents dans des images précédemment transmises. La présence de mouvement dans une vidéo rend la sélection des pixels dans ces images plus complexe, comme l'illustre la figure 1.6. Afin de saisir au mieux la redondance temporelle, le mouvement entre différentes images impose deux traitements : l'estimation et la compensation de mouvement.

### 1.5.2 Estimation et compensation de mouvement

L'estimation de mouvement est l'étape qui vise à mettre en correspondance des pixels de deux images voisines de la séquence vidéo. Le principe est basé sur une mesure de similarité des pixels entre eux, généralement traités par bloc de pixels. Une des techniques est connue sous le nom de *block-matching*. Les relations de correspondance ainsi trouvées sont scellées par le calcul de vecteurs de mouvement. Ces vecteurs indiquent l'emplacement des pixels retenus dans l'**image de référence**, comme étant ceux ayant le plus de corrélation avec les pixels courants. Ainsi, la précision des vecteurs mouvement va fortement conditionner l'étape suivante de compensation de mouvement, l'étape qui exploite les informations issues des vecteurs mouvement pour fournir la prédiction temporelle. On parle de prédiction **inter-images**. La figure 1.7 présente un exemple de corrélation temporelle entre deux images.

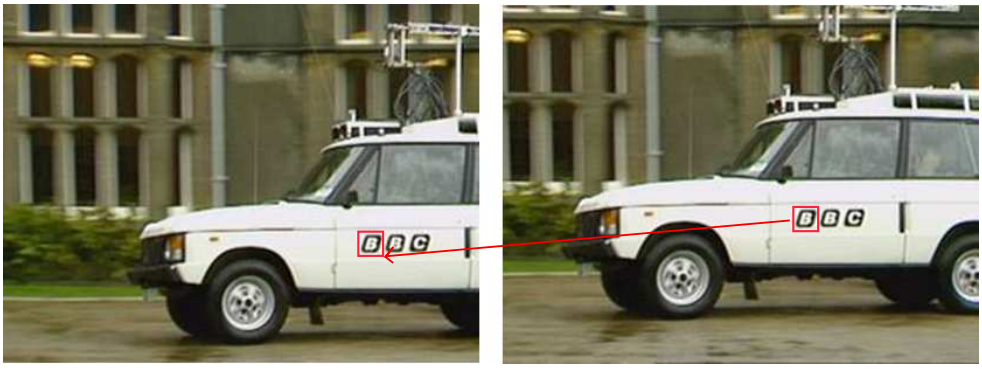



FIG. 1.7 – Recherche de corrélation temporelle

### 1.5.3 Problématiques soulevées

Trouver une prédiction à partir d'images voisines soulève de nombreuses difficultés. Entre deux images consécutives, des objets peuvent apparaître, disparaître ou être cachés par d'autres objets (phénomène d'occlusion). Comment alors prédire ces portions d'image pour lesquelles il n'y a, *a priori*, aucune information connue dans l'image voisine ? Pour contourner ces difficultés, plusieurs stratégies sont mises en place.

La première consiste à introduire périodiquement au sein de la séquence, des images codées uniquement de façon spatiale, *i.e.* elles se suffisent à elles-mêmes et sont indépendantes des autres images. Ces images sont appelées images I, pour images **intra** et sont aussi dénommées images clés ou *key frames*. Cela permet d'effectuer un rafraîchissement, particulièrement utile au décodeur quand un changement de plan est observé dans la séquence. Cette image va ensuite être le support de prédiction temporelle pour d'autres images : les images P ou prédites. Une image P exploite les corrélations temporelles au sein d'une image I ou d'une autre image P. On parle de prédiction mono-directionnelle.

Une autre stratégie consiste à prédire une image en cherchant des corrélations dans deux images de la séquence qui peuvent être situées dans le passé ou dans le futur. Il est en effet possible de «regarder» dans le futur car l'ordre de codage est différent de l'ordre d'affichage de la séquence. La compensation de mouvement bidirectionnelle est particulièrement efficace pour résoudre les problèmes d'apparition ou d'occlusion d'objets. Ces images se nomment les images B ou bi-prédites et peuvent chercher des corrélations temporelles dans plusieurs images

précédemment décodées.

Une séquence animée est donc codée selon trois types d'images : I, P et B. L'ordonnancement de ces images se choisit de telle sorte que l'on maximise le taux de compression. Nous pouvons voir en figure 1.8 un ordonnancement classique de ces images. Notons que les images B sont plus complexes en coût de calcul, comme nous le verrons par la suite, mais ce sont celles qui sont les moins coûteuses en débit.

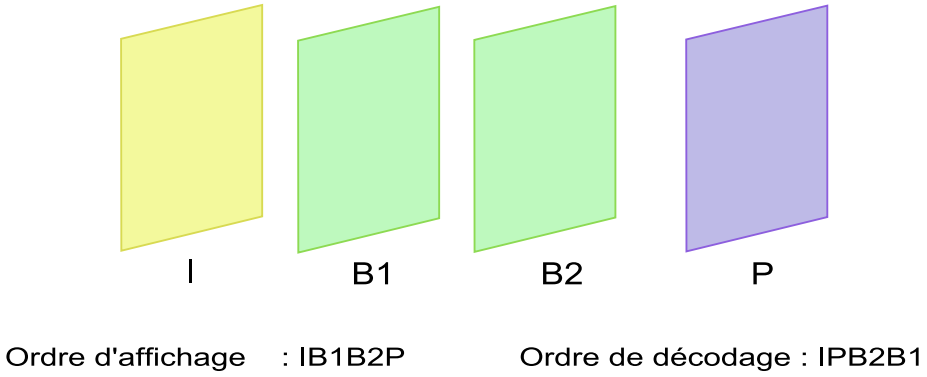



FIG. 1.8 – Exemple d'ordonnancement des images I, P et B

Jusqu'à présent, nous avons parlé de prédire une image à partir d'une autre. Rappelons ici que toutes ces images sont découpées en blocs de taille variable, ce que nous détaillerons par la suite. Le processus de prédiction temporelle s'opère sur ces différents blocs. Il est à noter que seules les images I sont entièrement prédites en spatial, mais la prédiction intra est également utilisée pour prédire certains blocs des autres images P et B. Ainsi, au sein d'une image P, des blocs sont prédits en intra et d'autres via une prédiction inter-images mono-directionnelle. Une image B contient des blocs prédits en intra, d'autres en inter-images mono-directionnelle et enfin certains via une prédiction bidirectionnelle.

## 1.6 Codage MPEG-4 AVC / H.264

### 1.6.1 Préambule

En 2003, le standard connu sous le nom de H.264 / MPEG-4 AVC a vu le jour, fruit du travail du JVT (*Joint Video Team*), constitué des groupes de l'ITU-T / VCEG et de ISO / MPEG. La norme H.264 / AVC n'est pas une rupture technologique architecturale en soi. Les modifications se situent aux différentes briques faisant partie intégrante du principe général de codage, exposé en section 1.4.

Il est communément admis qu'elle a permis d'améliorer les performances de compression de l'ordre de 50 % par rapport à l'existant [2]. Un codeur H.264 / AVC est dit *hybride* car il englobe à la fois une décorrélation par transformation spatiale (DCT) et une autre par prédiction temporelle (compensée en mouvement). La figure 1.9 illustre la structure utilisée dans un codeur H.264 / AVC.

[2] La précédente norme en vigueur était H.262 / MPEG-2.

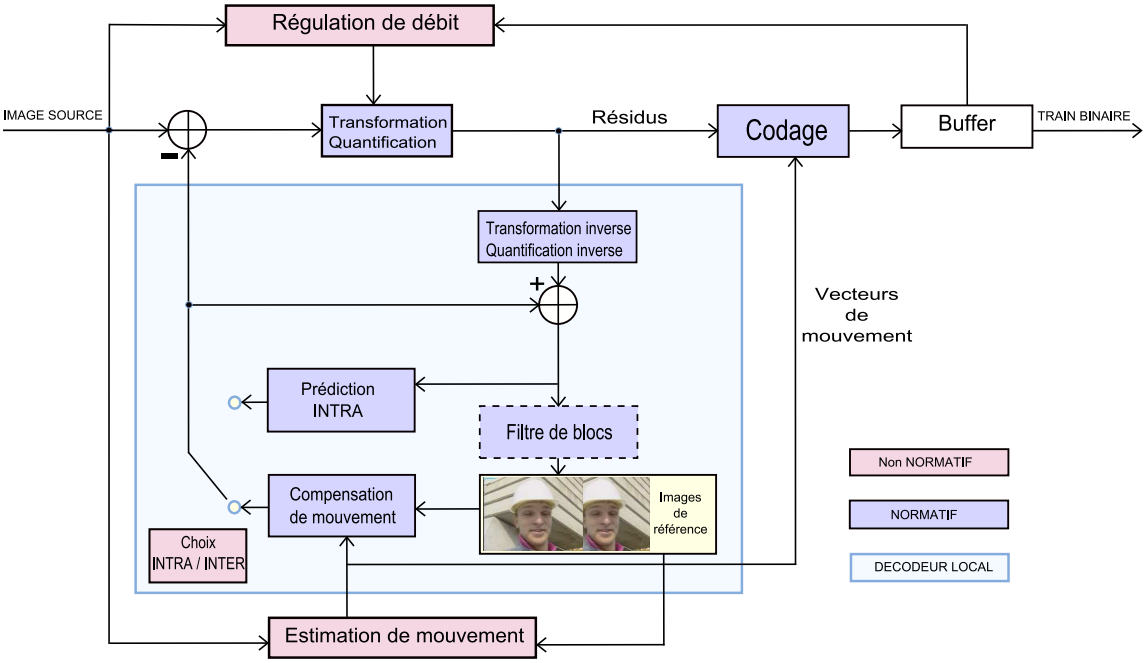



FIG. 1.9 – Schéma d'un encodeur vidéo hybride

## 1.6.2 Spécificités d'H264

### 1.6.2.1 Modes de codage

L'un des points clés des performances d'un encodeur H.264 réside dans la grande variété des modes de codage et en particulier, des modes de prédiction spatiale et temporelle. L'image est découpée en blocs de tailles variables. Le standard H.264 / AVC offre plusieurs partitionnements de l'image, à la fois pour le processus de décorrélation spatial et pour celui dédié à la prédiction temporelle.

En spatial, l'image est décomposée en trois tailles de blocs. Initialement, il y avait les **macroblocs** de $16 \times 16$ pixels composés de blocs $8 \times 8$. La nouveauté est de redécouper ces macroblocs en blocs de taille inférieure, *i.e.* en blocs de taille $4 \times 4$. En temporel, ou prédiction inter-images, la découpe des blocs est encore plus détaillée. Il existe bien sûr des blocs de taille $16 \times 16$ mais également des partitions autres, comme illustré sur les figures 1.10 et 1.11. Il existe des partitions $16 \times 8$, $8 \times 16$ et $8 \times 8$. Puis un bloc $8 \times 8$ peut être redécoupé et fournir les sous-partitions suivantes : $8 \times 4$, $4 \times 8$ et $4 \times 4$.

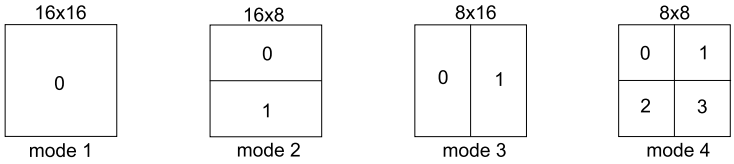



FIG. 1.10 – Partitions macrobloc : $16 \times 16$, $16 \times 8$, $8 \times 16$ et $8 \times 8$

L'avantage d'une telle décomposition est de permettre des traitements à différentes échelles et ainsi de s'adapter au contenu local de l'image. Il est plus pertinent de considérer un bloc $4 \times 4$ si le contenu de ce bloc correspond à un détail dans l'image. Tout traitement contraint à un bloc

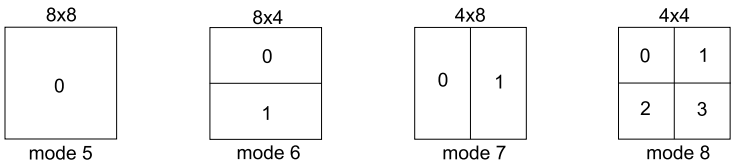



FIG. 1.11 – Partitions sous-macrobloc : $8 \times 8$, $8 \times 4$, $4 \times 8$ et $4 \times 4$

$16 \times 16$ pour représenter la finesse d'un contour, sera, très certainement, voué à l'échec.

Dans le cadre du mouvement, prenons l'exemple d'une personne se déplaçant sur un fond fixe. Si on se rapproche d'un détail, par exemple du bras, on aura un contour vertical entre deux zones bien distinctes : le fond fixe et le bras en mouvement. La partition $8 \times 16$ par exemple, permettra ici d'acquérir deux vecteurs de mouvement distincts pour chacune des zones concernées. Si seule la partition $16 \times 16$ était disponible, l'erreur serait d'autant plus grande que le mouvement relatif entre les deux zones serait important. On comprend ainsi la grande adaptativité introduite par ces partitions multiples, notamment grâce au partitionnement $4 \times 4$. La figure 1.12 présente un exemple de partitionnement d'un macrobloc $16 \times 16$ en blocs $8 \times 8$ et $4 \times 4$.

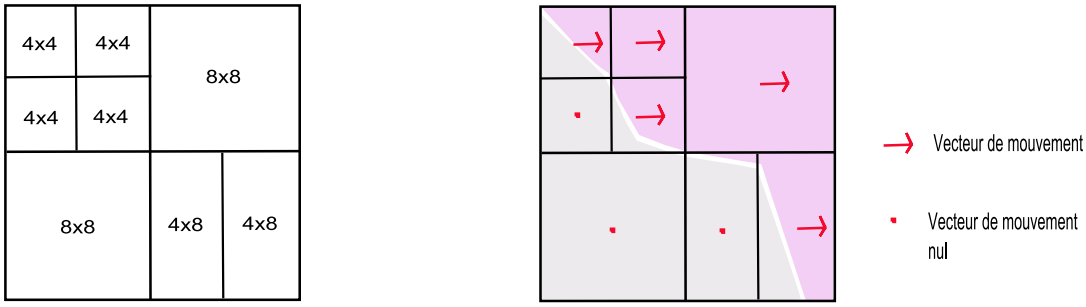



FIG. 1.12 – Exemple de partitionnement d'un macrobloc

Il faut néanmoins garder à l'esprit qu'à chaque sous-partition est associé un vecteur de mouvement doté de ses deux composantes $(dx, dy)$, qui vont demander un coût de codage. On s'aperçoit déjà qu'il faudra trouver un compromis entre le découpage en sous-partitions et le surcoût de codage.

#### 1.6.2.2 Images bi-prédites

Les images B sont prédites à partir d'images antérieures et/ou futures. Une nouveauté introduite par H.264 / AVC est d'autoriser les images B à être utilisées comme référence pour prédire d'autres images B. On les dénomme les *B-stored*.

Dans la norme précédente, il était impossible de prédire une image B à partir d'une autre image B. Seul était autorisé l'usage des images I et/ou P pour fournir la prédiction. Cependant, dans les cas de mouvement rapide, il peut être plus pertinent de s'appuyer sur les images les plus proches temporellement dans la séquence. Voici présenté en figure 1.13 un exemple d'ordonnancement des images I, P et B. Les flèches indiquent les images utilisées comme référence suivant leur type et le positionnement dans la séquence.

Cette nouvelle flexibilité, *i.e.* pouvoir coder une image B à partir d'une autre image B, a permis de mettre en place une structure dite «B-hiérarchique» offrant une amélioration de 10 à 15% des performances d'un encodeur H.264 / AVC, sans B-hiérarchique. Il s'agit d'une structuration dyadique des images I, P et B, créant alors une pyramide temporelle. On améliore ainsi la prédiction en favorisant une prédiction temporelle de proche en proche. De plus, le schéma en B-hiérarchique utilise un plus grand nombre d'images B que d'images P. Les images B sont moins coûteuses en débit car elles utilisent une prédiction bidirectionnelle qui conduit généralement à une meilleure prédiction que celle obtenue via une prédiction monodirectionnelle. Cela permet donc également d'améliorer les performances de l'algorithme de compression.

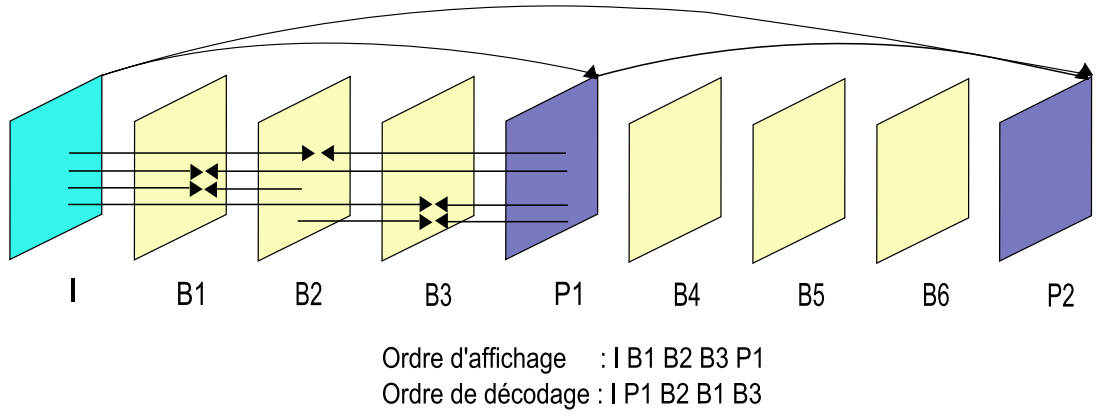



FIG. 1.13 – Schéma IPB généralement utilisé

#### 1.6.2.3 CABAC

Les principaux codeurs entropiques dans la norme H.264 / AVC sont le CAVLC (*Context Adaptive Variable Length Coding*) et le CABAC (*Context Adaptive Binary Arithmetic Coding*). L'usage de tables fixes des codes à longueur variable (VLC) ne permet pas de s'adapter aux valeurs statistiques des symboles traités au fur et à mesure du processus de codage. Il peut ainsi rester de la redondance entre les symboles qui ne pourra pas être exploitée. L'idée est donc de pouvoir s'adapter à un contexte, formé des valeurs des symboles précédemment codés, pour trouver le meilleur code du symbole courant. Le CAVLC offre ainsi la possibilité de s'adapter à l'environnement local des symboles précédemment codés.

Le CABAC, bien qu'il soit plus complexe que le CAVLC, est le processus de codage utilisé par défaut car il permet d'atteindre des performances supérieures, approximativement de 15%, par rapport au CAVLC. Le schéma 1.14 illustre les étapes clés du codeur, que nous allons détailler par la suite.

Le premier aspect remarquable du CABAC est d'être basé sur un codage arithmétique. Ceci lui permet donc de coder plusieurs symboles sur une valeur fractionnaire de bit[3]. Le CABAC tire son adaptabilité du fait que les probabilités, utilisées par le codeur arithmétique et issues du choix d'un modèle de contexte, sont mises à jour à chaque étape, à chaque traitement d'un élément binaire, un bin. En effet, la première étape du CABAC consiste à binariser les symboles en bin avant tout traitement. Puis ensuite, le codage s'opère en boucle sur chacun des bins du symbole qui vient d'être binarisé.

[3]Contrairement au codage de Huffman qui ne permettait pas de coder un symbole sur moins de un bit.

La deuxième étape consiste à sélectionner dans une table le modèle de contexte qui correspond aux données traitées. Les modèles de contextes prédéfinis ont pour but d'exploiter au maximum les redondances résiduelles entre les observations. Il existe ainsi plusieurs modèles de contextes, chacun correspondant en une certaine position spatiale des données voisines, précédemment codées. La notion de modèles se réfère à un agencement particulier dans l'espace de ces données. Au sein d'un modèle de contexte, il existe plusieurs états. Pour un agencement spatial donné, on répertorie les combinaisons de valeurs des bits des coefficients voisins. Autrement dit, on regarde l'état binaire des coefficients voisins, par plans de bits. La notion de modèle de contexte allie deux aspects : à la fois, la disposition spatiale des données traitées (au même titre que sur une image, il est pertinent de regarder ce qui a été fait pour les blocs voisins) et l'état binaire de ces données (le bit de poids fort d'un coefficient de valeur 32 sera différent d'un coefficient qui vaut 9).

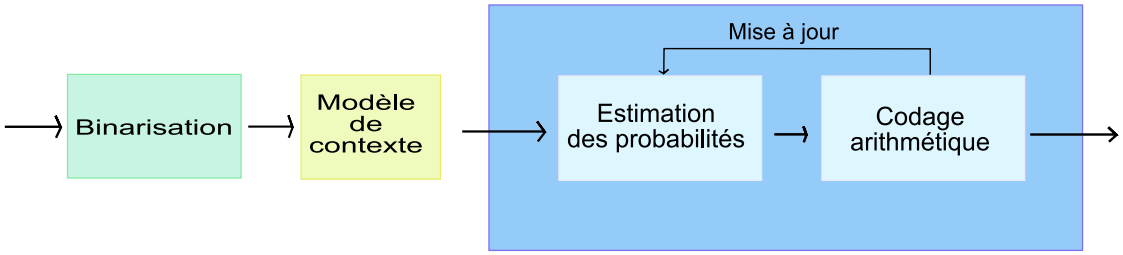



FIG. 1.14 – *Context adaptive binary arithmetic coding*

Ces modèles correspondent finalement à des tables qui proposent plusieurs états des bits des voisins. Pour chaque état du modèle de contextes, sont associées deux probabilités : la probabilité d'avoir un «1» et celle d'avoir un «0». Ce sont ces probabilités qui vont être rafraîchies au fur et à mesure. Il existe des modèles de contextes pour coder les vecteurs de mouvement, les indexs des images de référence utilisées ou encore pour coder le numéro du mode de prédiction spatiale choisi (cf section 1.6.3.1 pour la description de la prédiction intra-image).

L'étape suivante consiste à fournir au codeur arithmétique le bin courant et les probabilités liées à l'état du modèle de contexte retenu. Puis, en fonction de la valeur du bin courant, on met à jour la probabilité correspondante, pour cette instance. Si par exemple, le bin est un «1», la prochaine fois que la même instance sera sélectionnée, la probabilité relative au «1» sera légèrement plus élevée.

Nous avons présenté ici le fonctionnement général du CABAC. Sa complexité réside dans le fait que l'on tient compte pour *chaque bit* des données binarisées, de l'état d'un contexte, formé des observations précédentes. La richesse de ce codeur est rattachée à la quantité de modèles de contextes. Tous ces modèles ont été définis de manière exhaustive mais on peut noter que certains ont été abandonnés car l'observation a montré qu'il était inutile de les conserver tous. Tous ces paramètres font du CABAC un outil puissant dans un schéma de compression pour exploiter les corrélations résiduelles entre les données, qui n'auraient pas été exploitées par les précédents traitements de la chaîne de compression.

#### 1.6.2.4 Autres améliorations

Jusqu'à l'apparition d'H.264 / AVC, la transformée ne s'opérait que sur des blocs de taille $8 \times 8$ ne permettant pas une décorrélation fine du signal. Il a donc été introduit une transformation fréquentielle sur des blocs de taille $4 \times 4$. L'exploitation des corrélations spatiales résiduelles à une résolution plus fine permet d'améliorer la représentation des détails. On peut remarquer que

la transformation utilisée est définie de manière exacte (précision entière) afin d'éviter les erreurs d'arrondis.

Aux modifications présentées jusqu'à maintenant s'ajoutent encore quelques détails importants que nous présentons ci-dessous.

- La nouvelle norme permet de fournir jusqu'à seize vecteurs de mouvement par macrobloc. Jusqu'alors, seuls deux voire quatre vecteurs de mouvement étaient définis, ce qui limitait les performances de la prédiction temporelle.

- De même, le passage à une précision supérieure dans le calcul des vecteurs de mouvement a été une avancée majeure. Le principe de l'estimation de mouvement est de rechercher le meilleur prédicteur possible, au sein d'une image de référence. La recherche consiste à sélectionner un bloc de pixels, indépendamment d'une quelconque grille régulière. Afin d'améliorer la précision, sans se limiter à des valeurs entières pour les coordonnées retenues, on interpole l'image pour générer des positions sous-pixelliques. Dans la précédente norme, on ne pouvait interpoler que d'un facteur deux, ce qui engendrait une précision au demi pixel pour les vecteurs. La norme H.264 / AVC autorise une recherche au quart de pixel améliorant ainsi considérablement la précision des vecteurs de mouvement.

- Une autre spécificité d'H.264 / AVC a été l'introduction de modes supplémentaires, nommés modes directs, dont le but est de déduire les vecteurs de mouvement. Le principe consiste à éviter de les calculer en les estimant à partir des vecteurs définis pour les blocs voisins. Il existe deux modes directs : l'un spatial, l'autre temporel. En spatial, les vecteurs voisins correspondent à ceux retenus pour les blocs limitrophes au bloc courant. Le mode direct temporel utilise quant à lui, l'information du bloc du colocalisé[4] dans l'image de référence. Ces nouveaux modes sont appliqués pour les images B. Il existe également un mode direct spatial pour les images P, pour la prédiction inter $16 \times 16$. On gagne ainsi en débit puisque l'on évite d'avoir à coder l'information relative aux vecteurs de mouvement.

- La quantification a elle aussi été légèrement modifiée. On a augmenté le nombre de pas de quantification jusqu'à 52 niveaux afin d'améliorer la représentation du signal. Il a aussi été décidé de quantifier la chrominance de manière plus fine que la luminance afin d'améliorer le rendu visuel.

### 1.6.3 Prédiction d'images

#### 1.6.3.1 Intra prédiction

La prédiction intra-image se déroule dans le domaine spatial et se base sur la connaissance d'un voisinage de l'image courante en cours de reconstruction. Les blocs de l'image sont reconstruits au fur et à mesure. Si bien qu'à un instant quelconque du processus, un voisinage du bloc courant, dit **causal**, composé de trois, quatre ou cinq blocs, est disponible pour former la prédiction[5]. Cette proximité laisse tout naturellement supposer une forte corrélation entre le voisinage et le bloc à prédire, ce qui rend pertinent l'utilisation de l'information locale.

[4] Le colocalisé est un bloc ayant la même position spatiale que le bloc considéré mais dans une autre image.

[5] Pour un bloc situé sur le bord supérieur de l'image, le voisinage se limite au voisin situé à gauche. Dans le cas d'un bloc situé en bordure gauche, on restreint le voisinage aux blocs accessibles dans la partie supérieure.

*– Propagation des pixels voisins –*

La prédiction intra est basée sur de simples propagations et combinaisons linéaires de pixels. Afin de s'adapter au mieux à la texture locale, il est important de propager la texture selon un certain formalisme bien défini. Une texture peut présenter des caractéristiques très différentes, selon qu'il s'agisse d'une zone de l'image avec de faibles variations de niveaux de gris ou bien qu'il s'agisse d'une zone de variance élevée. Afin de répondre à cette problématique, plusieurs modes de prédiction existent.

*– Prédiction de la composante continue (DC) –*

Le mode DC est le mieux adapté pour modéliser les zones uniformes de l'image. Pour obtenir une prédiction DC, on calcule la moyenne des pixels limitrophes au bloc courant. L'estimation obtenue correspond à un bloc dont tous les pixels ont pour valeur la moyenne calculée. Le mode DC fournira une très bonne prédiction pour les zones de l'image où de larges plages de pixels ont, plus ou moins, les même valeurs. Ce signal n'en demeure pas moins une modélisation assez basique, non satisfaisante pour représenter des motifs plus complexes comme un contour ou une texture. Des modes directionnels furent donc créés pour synthétiser des signaux plus élaborés. Le principe est d'étendre la texture adjacente de manière judicieuse. Ces modes schématisent un sens de propagation des pixels voisins vers le bloc courant, le long d'une direction préalablement définie.

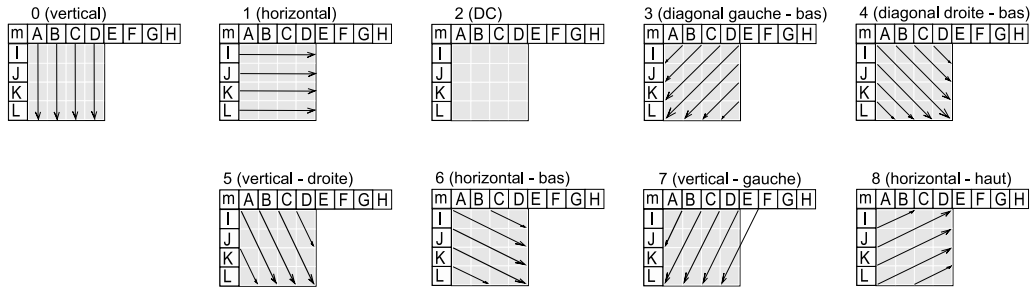



FIG. 1.15 – Présentation des 9 modes de prédiction intra $4 \times 4$

*– Modes de prédiction directionnels –*

Le panel de modes disponibles varie en fonction de la taille de bloc. Pour les blocs de taille $4 \times 4$ et $8 \times 8$, il existe neuf modes de prédiction : le mode DC et huit modes directionnels. Parmi ces derniers, on peut citer les modes horizontal, vertical, diagonal gauche-bas, diagonal droite-bas. Les modes restants correspondent à des directions dont l'angle est supérieur ou inférieur à 45°. La figure 1.15 répertorie l'ensemble des neuf modes de prédiction pour un bloc $4 \times 4$. Il est à noter que les même modes existent pour un bloc de taille $8 \times 8$.

Nous pouvons remarquer dans l'exemple exposé à la figure 1.16, que les neuf prédictions générées présentent des motifs sensiblement distincts. La prédiction intra offre ainsi un assez large panel de prédicteurs potentiels. La génération de signal via une simple méthode de recopie et de combinaisons linéaires, peut sembler être une approche naïve. Mais en dépit de leur simplicité

apparente, ces quelques orientations suffisent à représenter la plupart des directions présentes dans une image.

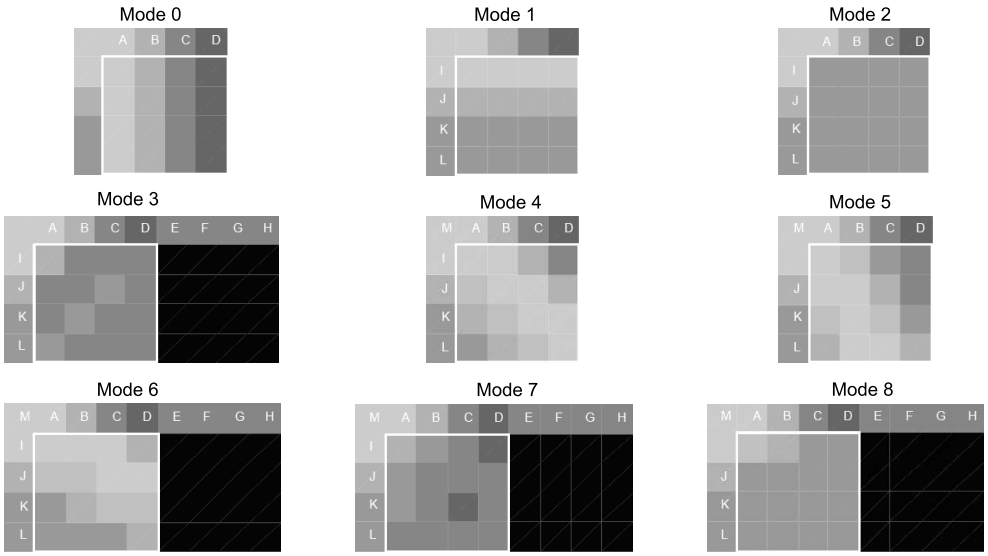



FIG. 1.16 – Prédictions intra appliquées à un bloc $4 \times 4$

Dans le cas des blocs de taille $16 \times 16$, quatre modes de prédictions ont été définis. On retrouve le mode DC pour synthétiser des textures uniformes, ainsi que deux modes directionnels : les modes vertical et horizontal. Le dernier mode se distingue des précédents par son caractère non directionnel. Il s'agit du mode *plane*. Comme son nom l'indique, la prédiction qu'il génère est un plan dont les deux vecteurs directeurs qui définissent son orientation, ont été calculés à l'aide des pixels voisins, horizontaux et verticaux. Si l'on se réfère à la figure 1.17, on constate que le mode plane permet de générer un dégradé de niveaux de gris.

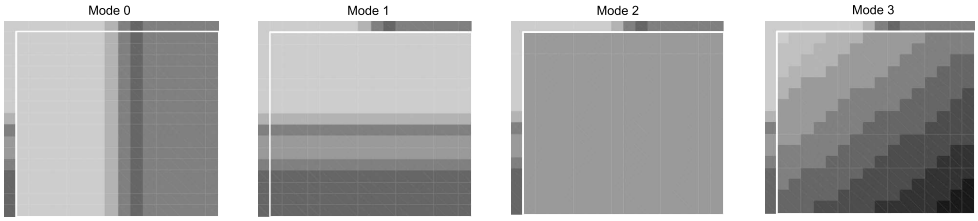



FIG. 1.17 – Prédictions intra $16 \times 16$

#### 1.6.3.2 Travaux en prédiction intra

Le but de cette section est de donner un aperçu des différentes techniques mises en oeuvre dans la littérature pour améliorer l'efficacité de la prédiction intra-image. Les méthodes présentées visent pour la plupart à améliorer la prédiction sans induire de modifications qui nécessiteraient un changement de syntaxe de l'encodeur. On peut distinguer différents types d'approche en fonction de la voie choisie pour améliorer la prédiction. Certaines se basent sur la prédiction d'H.264 / AVC mais recherchent des techniques pour améliorer la propagation des pixels. Il existe également de nombreuses recherches sur l'élaboration de nouveaux prédicteurs.

*- Nouveaux agencements pour H.264 / AVC -*

Des travaux [Wie03] ont proposé un partitionnement des blocs intra similaire à celui utilisé dans la prédiction inter (figures 1.10 et 1.11). Ils adaptent également la transformation afin qu'elle supporte les différentes tailles de bloc, nouvellement définies. La mise en place d'un tel schéma a pour vocation d'exploiter au mieux les corrélations et d'introduire une plus grande variété de choix, et donc d'adaptabilité, au niveau de la prédiction intra. Cette technique présente d'ailleurs des résultats très intéressants, aussi bien sur l'aspect intra-image que sur celui en inter. Dans [Wie03], l'impact d'une transformée adaptative en fonction des partitions, appliquée sur une prédiction inter, a aussi été évalué.

Dans le même esprit de complexification du partitionnement, [DEY+07] propose de découper le bloc suivant une géométrie particulière. Les auteurs déterminent un modèle paramétrique linéaire définissant la coupure optimale pour diviser le bloc en sous-partitions.

Les travaux [RAPP06] présentent une solution permettant de réduire notablement les corrélations résiduelles, observées au niveau du résidu lorsque la prédiction intra de la norme n'a pas suffi pour récupérer tous les contours. La technique consiste à appliquer des permutations circulaires, émulant ainsi des rotations au sein du bloc prédit, pour réordonner les pixels, qui suivent initialement des directions privilégiées prédéfinies, le long d'horizontales ou de verticales. Dans la suite du processus de codage, la DCT, transformation appliquée en ligne puis en colonne au résidu, est alors plus performante.

*- Recherche de nouveaux prédicteurs -*

Le *Template Matching* [TBS06] est une technique proche des algorithmes de *block matching* utilisés en prédiction inter-images. Il s'agit cependant de rechercher au sein de l'image courante, dans le voisinage causal, un bloc de pixels supposé représenter au mieux le bloc courant inconnu. Pour déterminer le meilleur prédicteur, on évalue une erreur entre les pixels voisins du bloc courant et des pixels au sein du passé causal, appartenant à une même forme spatiale (Figure 1.18). Cette forme est appelée *template*. Le meilleur prédicteur est celui qui conduit à l'erreur minimale entre les pixels des deux *template*. Dans la mesure où l'erreur est estimée entre des données de la même image, issues du passé causal, il n'est pas nécessaire de coder une information supplémentaire, comme cela est le cas dans le cadre d'une estimation de mouvement en inter où l'on code les vecteurs de mouvement.

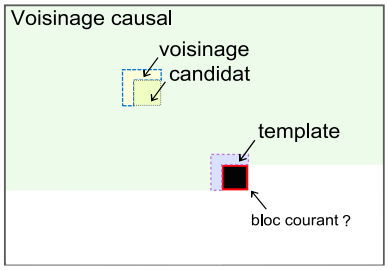



FIG. 1.18 – Principe du *Template Matching*

Comme autre exemple, nous pouvons évoquer les travaux de [PPJ07] qui prouvent la pertinence d'utiliser une approche sous-pixellique, classiquement réservée à la prédiction temporelle, pour générer de nouveaux modes intra.

Par ailleurs, de nombreux travaux proposent des solutions qui enrichissent la prédiction

intra de la norme H.264 / AVC. Certains [MTKY07] proposent d'ajouter des lignes de pixels supplémentaires à celles utilisées jusqu'à présent comme référence. D'autres solutions présentent des résultats en augmentant le nombre de prédicteurs directionnels [TYTA07] ou encore en combinant plusieurs des directions mono-dimensionnelles prédéfinies pour former des prédicteurs bi-dimensionnels [YK08]. Dans ces travaux, ils adaptent également la transformée ainsi que l'ordre des parcours des coefficients lors du processus de codage.

#### 1.6.3.3 Inter prédiction

La prédiction inter-images cherche à exploiter les corrélations temporelles entre plusieurs images. On pourra ainsi rechercher, pour chacune des partitions inter présentées précédemment, dans des images I, P ou B, les pixels les plus proches du bloc courant (au sens d'une métrique définie, cf. partie 1.7) .

L'estimation / compensation de mouvement peut se faire sur des images présentes dans le passé par rapport à l'image courante, mais aussi sur des images dans le futur. Lorsque la compensation de mouvement s'effectue vers l'avant, on parle de compensation *forward*. Ceci revient à rechercher les prédicteurs dans une image précédente. En revanche, si la prédiction retenue provient d'une image dans le futur, la compensation de mouvement se fait vers l'arrière : c'est une compensation *backward*.

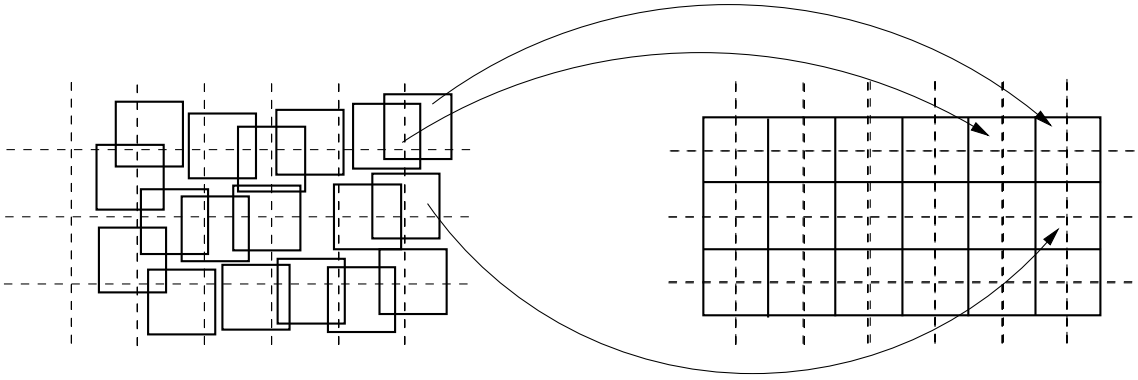


FIG. 1.19 – Prédiction temporelle *forward*

Dans la nouvelle norme H.264 / AVC, il est possible d'utiliser de multiples références pour obtenir la prédiction. Auparavant, dans le cadre de la prédiction monodirectionnelle, on utilisait une seule image de référence, qui avait, de plus, la contrainte de se situer dans le passé. A présent, pour prédire une image P, on peut rechercher le meilleur prédicteur dans plusieurs images de référence, positionnées à n'importe quel endroit de la séquence. La prédiction reste monodirectionnelle dans le sens où un seul macrobloc sera utilisé pour prédire les pixels, mais il aura été choisi parmi plusieurs images de référence. Pour les images B, l'idée est la même : le processus est toujours bidirectionnel car deux macroblocs sont utilisés mais il est autorisé de changer d'image de référence au cours de processus de prédiction de l'image courante, comme illustré sur la figure 1.20.

Ainsi deux macroblocs voisins ne seront pas nécessairement prédits à partir des mêmes images de référence. Les références sélectionnées n'ont également plus la contrainte de positionnement temporel fixé dans la séquence. Cette grande flexibilité permet notamment de résoudre efficacement les problèmes d'apparition, d'occlusion ou encore de *fading*[6].

[6]Variation de luminosité des pixels. Par exemple, un *fading* linéaire correspond à un assombrissement progressif des images, généralement utilisé pour marquer la transition entre deux scènes d'une vidéo.

Lorsque la prédiction est bidirectionnelle, celle-ci correspond à la moyenne des deux prédictions retenues pixel à pixel. Notons qu'il existe également dans la norme H.264 / AVC, une prédiction pondérée (*weighted prediction*) qui permet d'ajuster l'importance d'un macrobloc par rapport à un autre, de manière implicite ou explicite. En effet, on favorisera le bloc de prédiction temporellement le plus proche, par rapport à l'image courante.

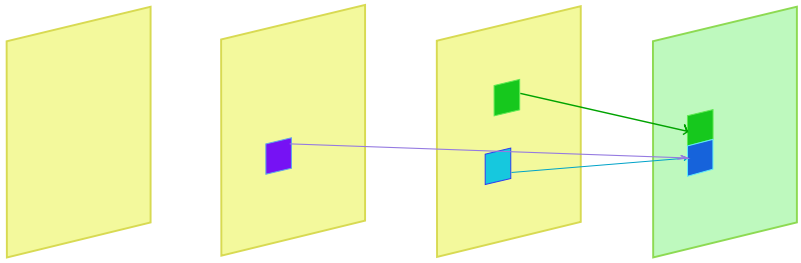


FIG. 1.20 – Références multiples

Nous pouvons remarquer que la prédiction pondérée est aussi applicable de façon explicite dans le cas d'une prédiction mono-directionnelle (pour une image P ou B) ; cela permet de réajuster la luminance des pixels. Typiquement, lorsque apparaît un flash d'appareil photo sur une image, le phénomène est très court et a de forte chance de n'apparaître que sur cette image. On peut être amené à faire référence à ces pixels localisés au niveau du flash, sans toutefois vouloir reproduire cet effet. La prédiction pondérée permet de compenser cette variation brutale de luminosité. Ces pixels de l'image de référence, ainsi compensés en luminosité, serviront ensuite comme prédicteurs pour un bloc de l'image courante.

## 1.7 Métriques de distorsion et critère d'optimisation des modes

Le processus de sélection des modes n'est pas normatif dans H.264 / AVC, ce qui signifie que l'on peut choisir n'importe quel algorithme de sélection. Pour un macrobloc donné appartenant à une image P ou B, on commence par déterminer les prédictions intra et inter images, correspondant à toutes les partitions possibles. Si l'image courante est une image I seule la prédiction intra-image sera activée. Par exemple, on peut décider de générer toutes les prédictions inter et intra, pour l'ensemble des partitions définies et ne choisir qu'ensuite le meilleur mode pour le macrobloc. Ou alors, on procède par étape en choisissant le meilleur mode pour une partition donnée et pour un type de prédiction (intra ou inter). Parmi ces meilleurs modes, on sélectionne le meilleur pour le macrobloc. Cette technique est donc nécessairement sous optimale par rapport à la précédente qui met en concurrence tous les modes, en une seule fois. La figure 1.21 présente un exemple d'algorithme de sélection des modes.

La sélection des modes se fait en général par le biais de la minimisation d'une fonction de coût. Cette fonction peut être définie selon plusieurs critères prenant en compte plus ou moins de paramètres. Nous présentons dans cette section les critères basés uniquement sur une mesure de distorsion et ceux, plus complexes, basés sur une optimisation débit / distorsion.

### 1.7.1 Mesures de distorsion

Le moyen le plus simple pour évaluer la qualité d'une prédiction est de mesurer la distorsion $D$ entre le bloc source correspondant et la prédiction. Cette mesure évalue l'écart entre deux images en sommant les erreurs faites pixel à pixel.

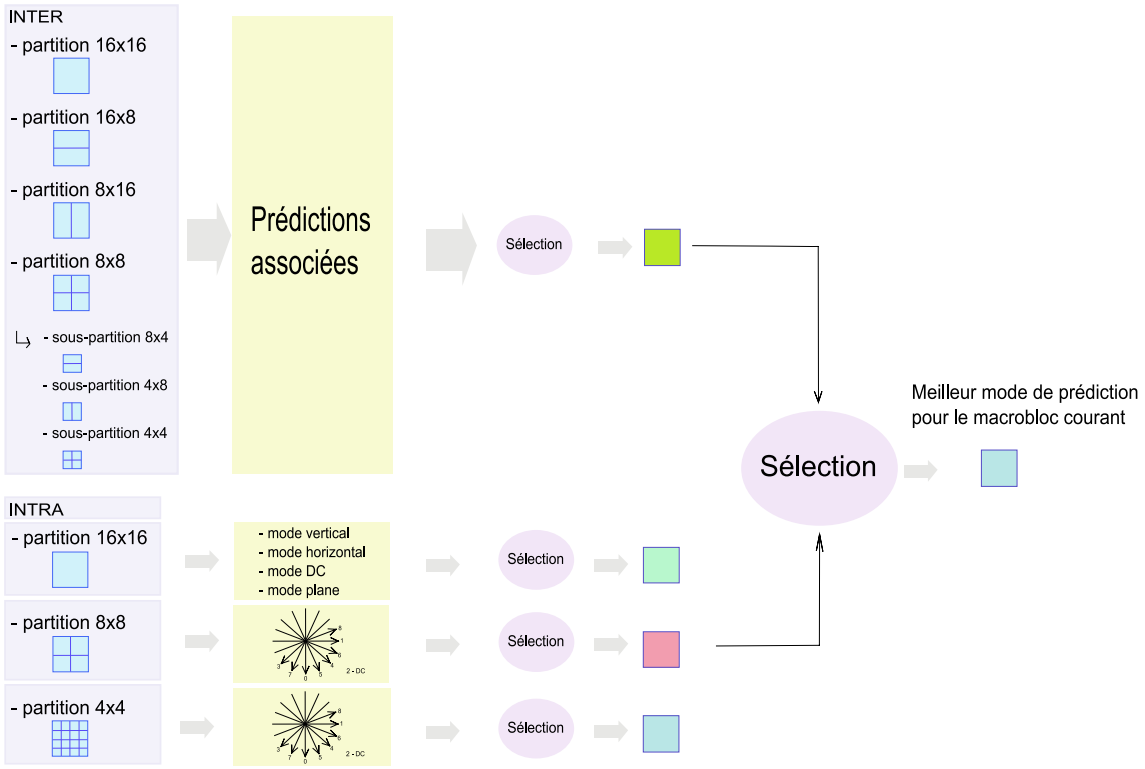



FIG. 1.21 – Exemple de sélection des modes de prédiction

Le critère historiquement utilisé pour évaluer deux images entre elles, est le PSNR pour *Peak Signal to Noise Ratio*. Cette mesure de distorsion est donnée par la relation suivante :

$$PSNR = 10.\log_{10}\left(\frac{d^2}{EQM}\right)$$

où $d$ est la dynamique du signal et l'EQM est l'Erreur Quadratique Moyenne que l'on définit de la façon suivante pour deux images $x$ et $y$, de taille $M \times N$ :

$$EQM = \frac{1}{MN}\sum_{i,j}(x(i,j) - y(i,j))^2$$

Dans les critères d'évaluation, on utilise plus simplement, la somme des erreurs au carré ou SSE pour *Sum Square of Errors* :

$$SSE = \sum_{i,j}(x(i,j) - y(i,j))^2$$

L'élévation au carré rend ces mesures particulièrement sensibles si les deux images sont trop éloignées. Un faible nombre de pixels ayant des valeurs trop différentes, suffit à perturber la mesure. Néanmoins, cette mesure de distorsion reste très efficace si la différence entre $x$ et $y$ correspond à un bruit gaussien, *i.e.* si les images ne présentent pas de différences majeures. Si ce n'est pas le cas, on préférera utiliser la SAD ou *Sum of Absolut Difference*. Cette métrique consiste à calculer la norme $l_1$ de l'image d'erreur :

$$SAD = \sum_{i,j}\left|x(i,j) - y(i,j)\right|$$

La SAD sera plus adaptée dans les cas où les deux images sont structurellement bien différentes.

### 1.7.2 Optimisation débit / distorsion

Pour affiner l'évaluation de deux images dans le cadre de la compression, il fut mis en place un processus cherchant le meilleur compromis entre la qualité de la reconstruction et le nombre de bits nécessaires pour coder puis transmettre l'information résiduelle. Il s'agit d'un problème d'optimisation connu sous le nom RDO pour *Rate Distorsion Optimization.*

Ce critère est basé sur une fonction de coût lagrangienne $J_\lambda$ que l'on cherche à minimiser :

$$J_\lambda = D + \lambda.R$$

où $D$ est une mesure de distorsion, $R$ représente le débit en bits et $\lambda$ un paramètre d'ajustement qui dépend des contraintes de quantification. La métrique de distorsion utilisée est en général la SSE ou la SAD.

Notons qu'il existe des décisions *a priori* et *a posteriori*. Dans le cadre de la décision *a posteriori*, la valeur exacte du débit $R$ doit être connue, ce qui nécessite alors de passer par le processus complet de codage (transformation - quantification). La SSE quant à elle impose la reconstruction via les étapes de quantification et transformation inverses.

La technique est performante mais reste coûteuse car cela nécessite de répéter le processus pour chaque bloc et chaque mode à tester. L'alternative est l'utilisation d'une approche moins complexe mais néanmoins sous-optimale, qui consiste à évaluer *a priori* le coût qu'aurait le bloc en fin de processus. Pour estimer le débit $R$, on se base sur des modèles empiriques d'optimisation débit / distorsion, permettant d'approcher les performances des algorithmes *a posteriori*.

## 1.8 Codage scalable

Nous présentons dans cette section le concept global d'un codeur scalable et plus spécifiquement, la norme SVC, pour *Scalable Video Coding*. Un encodeur SVC repose sur les concepts connus d'H.264 / AVC mais inclut des traitements entre les images à différentes résolutions. Nous nous limiterons dans la section 1.8.1 à exposer la philosophie générale puis nous détaillerons dans la partie 1.8.2 un point particulier lié à la prédiction spatiale inter-couches de SVC, qui a fait l'objet d'une application dans cette thèse (au chapitre 4, section 4.4).

### 1.8.1 Principe général

La scalabilité est un terme désignant un système ayant la capacité d'évoluer en performances. Dans le cadre de la compression vidéo, la scalabilité désigne l'aptitude d'un algorithme de compression à représenter une vidéo hiérarchiquement, sur plusieurs trains binaires. Parmi ceux-ci, une couche de base est indépendamment décodable des autres et permet la reconstruction des données à un niveau de qualité minimum. La prise en compte de sous-flux binaires de raffinement par le décodeur permet d'obtenir des incréments de qualité de reconstruction. Ces sous-flux peuvent être emboîtés dans un même train binaire ou diffusés sur des canaux distincts. Cette spécificité permet de s'adapter à la capacité des récepteurs, comme le schématise la figure 1.22.
Dans SVC, il existe trois niveaux de scalabilité :

1. *la scalabilité spatiale* ou scalabilité en résolution, définit une hiérarchie de résolutions spatiales. La résolution désigne la taille en pixels des images reconstruites.

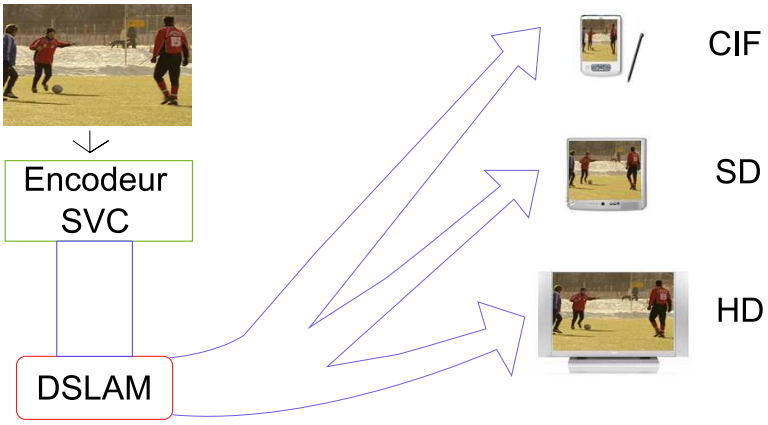



FIG. 1.22 – Exemple d'application de SVC

2. *la scalabilité temporelle* porte sur la fréquence de rafraîchissement des images : elle définit une hiérarchie de résolutions temporelles.
3. *la scalabilité en fidélité* ou scalabilité SNR (*Signal to Noise Ratio*) consiste à augmenter le rapport signal à bruit d'une couche donnée de la hiérarchie, c'est-à-dire réduire la distorsion de quantification entre images originales et reconstruites.

La figure 1.23 schématise les trois scalabilités rencontrées dans SVC. L'image en haut à gauche correspond à l'image, pleine résolution pour les trois scalabilités. L'image en haut à droite présente l'image avec un niveau de scalabilité spatiale, celle en bas à gauche, pour un niveau en scalabilité SNR et celle en bas à droite, pour un niveau de scalabilité temporelle.

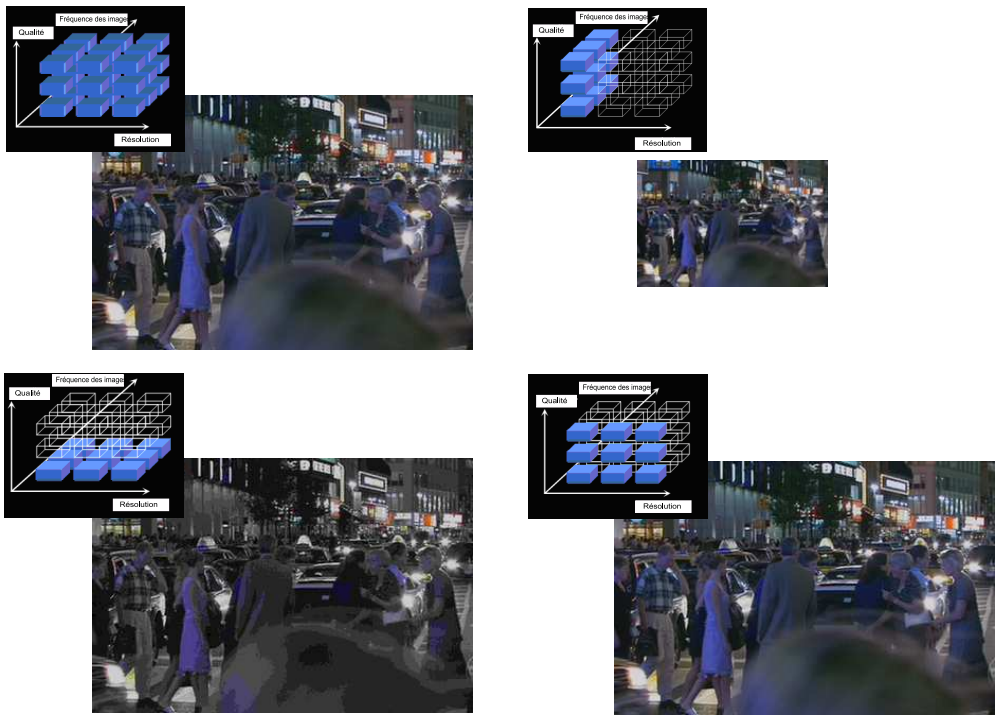



FIG. 1.23 – Scalabilités dans SVC

Une particularité de ce standard se base ainsi dans la possibilité d'exploiter les corrélations entre les images aux différentes résolutions. Une couche de base est définie comme point de départ, puis des couches d'amélioration de résolutions supérieures sont ajoutées en fonction des besoins.

### 1.8.2 Structure de prédiction pyramidale

Chaque couche est basée sur un encodage H.264 / AVC auquel se rajoute un processus de prédiction basé sur l'exploitation de données entre couches. Dans SVC, on distingue trois types de prédiction inter-couche : la prédiction spatiale, la prédiction des vecteurs de mouvement et la prédiction du résidu. Chacun de ces modes supplémentaires vise à utiliser les informations connues d'une couche de base, pour former de nouveaux prédicteurs. Nous ne présentons dans cette partie que la prédiction spatiale inter-couches.

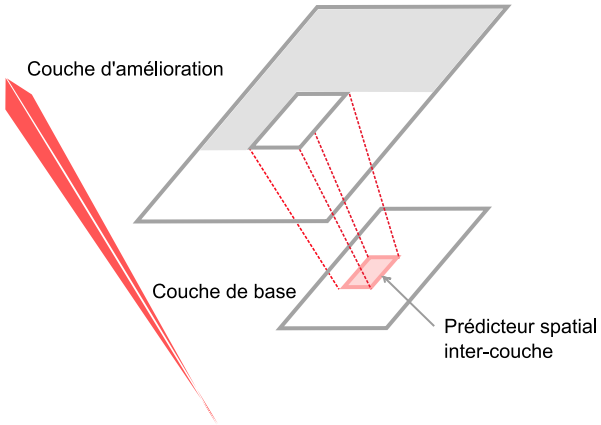



FIG. 1.24 – Prédiction spatial inter-couches

La prédiction spatiale dite intra est constituée, comme nous venons de l'exposer en section 1.6.3.1, de neuf modes de prédiction qui exploitent la connaissance d'un voisinage causal de pixels connus, limitrophes au bloc courant. A ces neuf modes, s'ajoute un mode de prédiction inter-couches basé sur l'interpolation du bloc colocalisé dans une couche de base, comme l'illustre la figure 1.24. Cette couche n'étant pas à la même résolution, il est nécessaire d'interpoler le bloc de la couche de base. Notons que la prédiction spatiale inter-couches n'est possible que dans les cas où le macrobloc dans lequel se situe le bloc colocalisé dans la couche de base, a été codé en mode intra.

## 1.9 Conclusion

Nous avons présenté dans ce chapitre les concepts généraux utilisés en compression d'images et de vidéos. Nous avons également mis en avant les motivations qui ont conduit à faire certains choix dans les différentes étapes de l'encodage. Nous nous sommes ensuite familiarisés avec certaines spécificités d'un codeur H.264 / AVC et notamment avec les concepts utilisés en prédiction intra-images, thème majeur au sein de cette thèse. Actuellement, la norme H.264 / AVC offre une grande flexibilité d'utilisation et reste difficile à surpasser en performances. Néanmoins on peut imaginer améliorer la norme par l'enrichissement des modèles de prédiction spatio-temporelle, d'estimation de mouvement ou encore réfléchir à l'élaboration d'une nouvelle structure qui viserait à repenser les modèles actuellement définis. Nous nous sommes plus particulièrement intéressés dans cette thèse à étudier des algorithmes visant à améliorer la prédiction spatiale.

# Chapitre 2

# Synthèse de texture

Nous avons présenté dans le chapitre 1 des techniques de prédiction intra-image et tout particulièrement celles mises en oeuvre dans un encodeur de type H.264 / AVC. Le coeur de nos travaux va en effet s'articuler autour de la problématique de prédiction qui peut être vue comme un problème d'extrapolation d'image ou encore de synthèse de texture. Ce chapitre a donc pour vocation de présenter un rapide survol des techniques actuelles en matière de synthèse de texture. Les notions d'extrapolation et de synthèse étant particulièrement proches, nous pouvons considérer que les techniques de synthèse soulèvent le même type de difficultés que celles d'extrapolation. L'enjeu est ici de réussir à cerner les divers angles sous lesquels a été abordé le thème de la synthèse de texture, afin de mieux en comprendre les problématiques qui y sont rattachées, ainsi que les solutions apportées dans la littérature.

Nous présenterons ici les travaux en matière de synthèse de texture bi-dimensionnelle. Bien que les travaux actuels dans la littérature s'orientent vers les applications sur la 3D ou sur la réalité virtuelle, nous n'aborderons pas les problématiques rattachées à la modélisation de texture de surface et à la définition de leur géométrie. De même, les travaux sur les textures en mouvement, incluant la dimension temporelle, ne seront pas évoqués. Nous relaterons simplement les diverses techniques employées en matière de synthèse de texture dans le plan 2D.

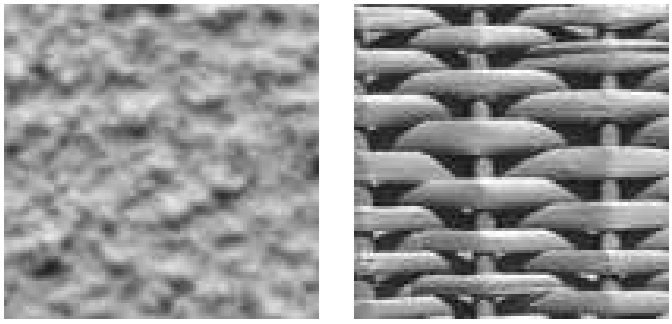


FIG. 2.1 – Texture stochastique (à gauche) et texture régulière (à droite)

## 2.1 Introduction

### 2.1.1 Qu'est-ce qu'une texture ?

Bien que nous soyons tous à même de reconnaître une texture, il est bien difficile d'en donner une définition mathématique précise. On peut néanmoins dégager certaines caractéristiques principales. Au sein d'une texture, on retrouve généralement des motifs qui vont être répétés à une ou plusieurs échelles, selon une périodicité plus ou moins régulière.

Une texture est donc constituée d'un ensemble de plusieurs pixels formant une unité particulière. Ce sont les variations de luminosité des pixels qui, de par leur caractère aléatoire ou non, permettent de distinguer les textures stochastiques de textures très régulières, dont voici un exemple en figure 2.1. Entre ces deux extrêmes, s'étale toute la plage et la richesse des textures naturelles : du sable d'une plage à un parterre de fleurs, de l'herbe d'une pelouse au pelage d'un animal (cf. figure 2.2, une classification proposée par [LLH04] ).

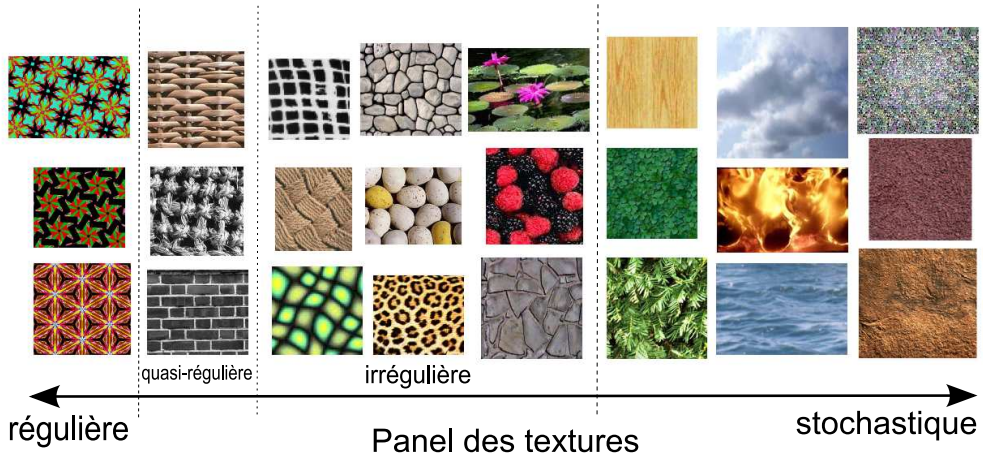



FIG. 2.2 – Exemple de textures naturelles

Il est classiquement admis de caractériser une texture comme un processus stochastique local et stationnaire : pour une échelle donnée, la texture est la même quel que soit l'endroit où on l'observe.

### 2.1.2 Axes de recherche

Toute image naturelle contient potentiellement une texture. Il apparaît donc comme inévitable, dans le vaste domaine du traitement des images, de se pencher sur la problématique de l'analyse des textures. Les principaux centres d'intérêt en la matière concernent : la synthèse de texture, la segmentation et la classification. Nous aborderons dans les sections suivantes, les techniques actuelles prédominantes pour la synthèse de texture.

La segmentation quant à elle, a pour but de trouver les contours entre différentes textures d'une image. La difficulté repose sur le fait que la nature même de la texture est inconnue. Il peut donc être très complexe de distinguer localement des pixels formant une texture plutôt qu'une autre. Quant à la classification des textures, la technique peut être vue comme une analyse complémentaire à la segmentation. L'enjeu est de réussir à classifier une nouvelle texture par le biais d'un panel connu de textures de différentes classes.

### 2.1.3 Comment juger de la qualité de la synthèse ?

L'évaluation de la qualité d'une texture obtenue par synthèse est un problème majeur. La méthodologie générale en synthèse consiste à générer une image texturée à partir d'un petit échantillon connu du motif à reproduire. Les outils mathématiques classiques sont de fait inadaptés à cette problématique. Ils ne mesurent en effet la qualité de la reconstruction qu'en calculant une erreur pixel à pixel. De plus, comme nous l'expliciterons par la suite, les algorithmes de synthèse ne cherchent pas nécessairement à obtenir une copie conforme au signal d'origine. Seule une réalisation perceptuellement semblable suffit. Par conséquent, pour juger de la qualité de la synthèse, nous ne pouvons, *a priori*, nous fier qu'à notre seule perception visuelle, avec toute la subjectivité que cela implique. C'est pourquoi des critères psychovisuels ont été introduits afin de répondre à cette problématique. Ils sont basés sur des modélisations avancées du système visuel humain qui visent à émuler les interprétations de notre cerveau face à des *stimuli* visuels [WS04].

## 2.2 Exemples de techniques de synthèse

On s'accorde en général pour distinguer deux grandes classes de texture : celles qui présentent des aspects réguliers aux motifs bien déterminés, de celles dont l'agencement est plus aléatoire et donc dépourvues de structures établies. Nous retrouvons ainsi, de façon tout à fait logique, deux catégories d'algorithmes. On distingue ceux basés sur une approche structurelle déterministe visant à restituer la régularité du motif, de ceux dont l'approche est probabiliste, cherchant à caractériser l'aspect anarchique des textures stochastiques.

### 2.2.1 Approche stochastique

La philosophie d'une approche stochastique n'est pas de reproduire la texture à l'identique. On recherche plus exactement à créer une nouvelle texture comme si elle avait été générée par le même processus aléatoire.

#### 2.2.1.1 L'image : une réalisation d'un champ de Markov

La technique basée sur les champs de Markov ou MRF (Markov Random Field) est l'une des plus anciennes du domaine [CJ83, PP93, HB95, PL95]. Le principe consiste à modéliser l'image comme un champ stationnaire, où les pixels sont des variables aléatoires. Les dépendances entre elles forment le signal texturé. L'enjeu est alors de sélectionner la réalisation ou encore l'agencement optimal des pixels, qui conduit à une restitution acceptable du motif source. Pour estimer quel sera l'ordonnancement le plus probable, il est nécessaire d'apprendre la distribution sur l'ensemble de l'image. L'approche a démontré son efficacité pour un ensemble assez vaste et divers de textures. On peut supposer que cette robustesse est liée à l'adaptativité intrinsèque de la technique qui dépend directement du signal source. Cependant, les algorithmes mis en place restent encore actuellement beaucoup trop coûteux.

#### 2.2.1.2 Structure hiérarchique

Les travaux de [Bon95] introduisent une structure hiérarchique au coeur du processus de synthèse. Une hypothèse fondamentale de la technique consiste à supposer qu'une texture est composée de plusieurs régions, qui se distinguent les unes des autres à une faible valeur près. Autrement dit, certaines caractéristiques d'une texture peuvent être visibles à une résolution sans l'être à une autre. Partant de cette hypothèse, l'analyse est basée sur une décomposition de l'image à plusieurs échelles, visant à extraire jusqu'aux composantes les plus basiques de l'image : on

obtient une décomposition en sous-bandes de différentes fréquences spatiales, un exemple est représenté en figure 2.3.

FIG. 2.3 – Décomposition en sous-bandes

Une solution proposée par Javier Portilla et Eero Simoncelli [PS00] offre de bons résultats. Leur approche est basée sur l'exploitation des corrélations entre coefficients d'ondelettes aux différentes sous-bandes de la transformation. Le modèle statistique est paramétré en tenant compte de diverses valeurs statistiques dépendantes des corrélations entre coefficients, voisins en termes de localisation spatiale, d'orientation et d'échelle.

### 2.2.2 Approches structurelles

#### 2.2.2.1 Méthodes supervisées basées pixel

Ces dernières années des méthodes très efficaces ont été développées, basées sur la recopie de pixels issus d'un patch modèle. La philosophie de ce type de méthode peut s'apparenter à une approche déterministe de la synthèse par des champs de Markov. La technique a d'abord été introduite par Alexei A. Efros et Thomas K. Leung [EL00]. Puis, les travaux de Li -Yi Wei et Marc Levoy [EL00] ont démocratisé ce type d'algorithmes. Ils génèrent la texture en recherchant des correspondances point à point, entre le voisinage de l'image en cours de synthèse et des voisinages du patch source, comme l'illustre la figure 2.4.

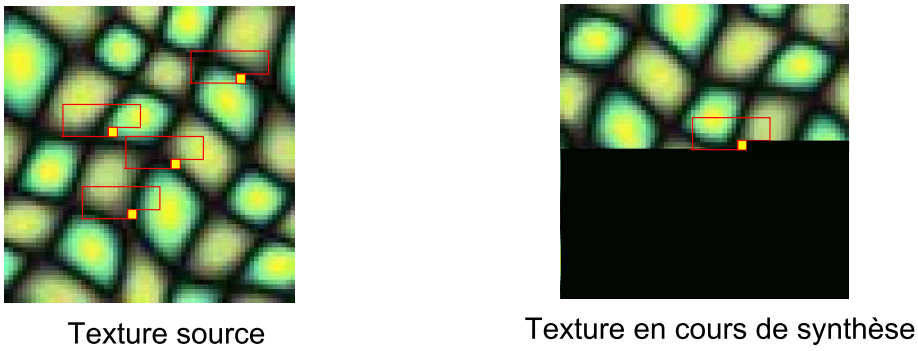



FIG. 2.4 – Principe de la synthèse de Wei et Levoy

Par la suite, des travaux, notamment ceux de Michael Ashikhmin [Ash01], se sont orientés vers une extension de la technique de Wei et Levoy. L'exploitation de stationnarités locales dans sa méthode lui permet de réduire le nombre de candidats possibles. La figure 2.5 présente quelques uns de ses résultats. Puis, les travaux dans le domaine se sont tout naturellement axés vers des techniques visant de plus en plus à recopier des morceaux complets du patch source. Citons les travaux de Vivek Kwatra et al. [KEBK05] qui mettent en place une approche d'optimisation plus

globale basée sur un algorithme de type EM[1].

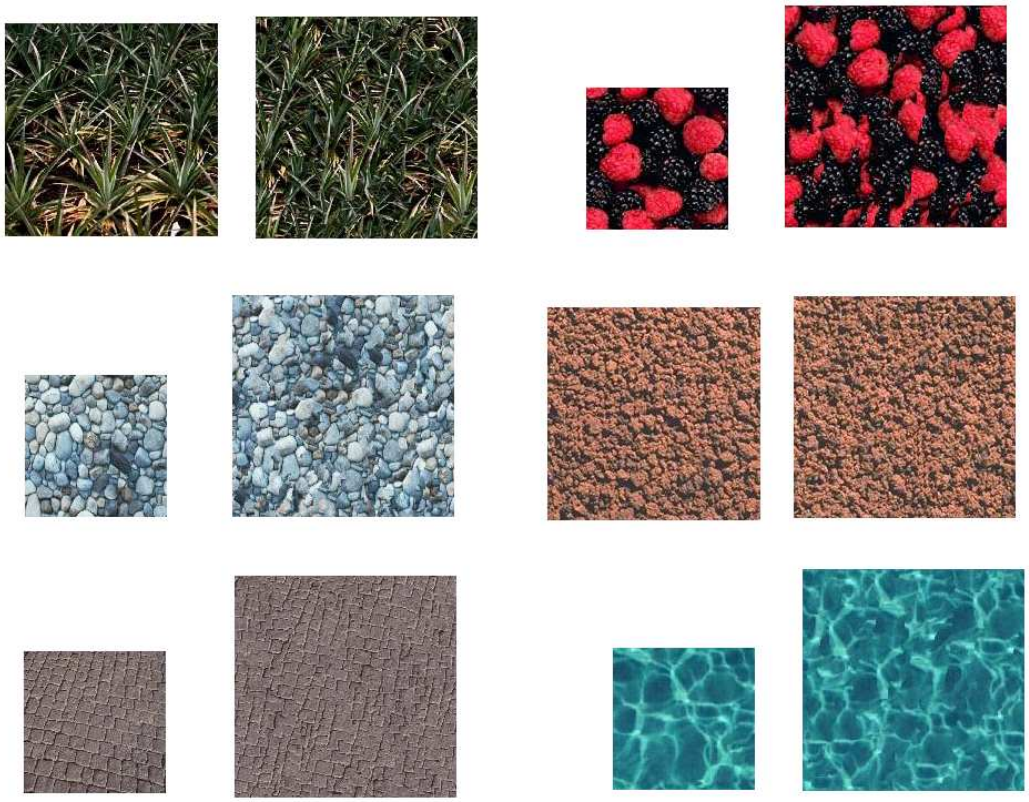


FIG. 2.5 – Exemples de synthèse obtenus avec l'algorithme d'Ashikhmin : à gauche, le patch source ; à droite, le résultat de la synthèse

#### 2.2.2.2 Méthodes basées bloc

*– Chaos mosaic –*

Cette idée originale a été développée par Ying-Quing Xu et al. [XGS00]. La technique proposée est basée sur un placement répété et aléatoire du patch source dans une nouvelle image. La technique faisant apparaître de nombreuses discontinuités entre blocs voisins, les contours sont ensuite filtrés pour éliminer ces artefacts. La figure 2.6 présente un exemple de résultats de synthèse par cette méthode. Par nature, cette technique synthétise ainsi une texture d'aspect déstructuré tout en conservant néanmoins la nature locale du motif. L'algorithme est donc efficace pour générer des textures stochastiques mais présente de faibles résultats quant à la synthèse de textures ordonnées.

*– Synthèse par graphcut –*

[1]EM pour Espérance - Maximisation [DLR77]. Cet algorithme itère les étapes E et M successivement et vise à déterminer les paramètres d'un modèle probabiliste via la recherche du maximum de vraisemblance.

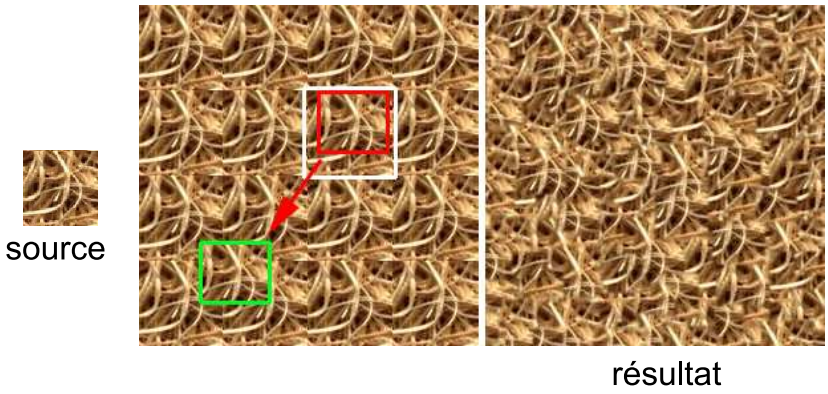



FIG. 2.6 – Synthèse par *chaos mosaic*

A. Efros et W.T. Freeman [EF01] ont cherché à combiner deux idées : celle du chaos mosaic qui recopie aléatoirement le patch source et celle qui introduit des contraintes spatiales au niveau pixellique [EL00]. L'idée consiste à recopier côte à côte les patchs source dans une nouvelle image, tout en tolérant une zone de recouvrement entre eux. La texture ainsi synthétisée est ensuite uniformisée en déterminant la couture optimale au niveau du recouvrement. La jointure optimale est calculée via des algorithmes de programmation dynamique. La figure 2.7 présente le principe de la méthode. Kwatra et al. [KSE+03] proposent d'obtenir cette couture en se basant sur la théorie des graphes plutôt que par la programme dynamique. Au niveau du recouvrement des patchs, on détermine le chemin de coût minimal[2] puis la coupe minimale, qui en découle, va définir la séparation optimale entre les blocs.

*– Synthèse par patch –*

Une des techniques la plus intuitive, qui permet également de préserver la structure globale de la texture, est la recherche au sein de l'image, d'un ensemble de pixels représentant au mieux (selon un critère défini) la texture à synthétiser. Le voisinage de pixels utilisé pour mesurer la correspondance a la contrainte d'être suffisamment grand pour capter la stationnarité de la texture. Cette synthèse revient à recopier des patchs entiers d'une texture connue. Cette technique, appelée *Template Matching* décrite au paragraphe 1.6.3.2 est généralement utilisée pour synthétiser une texture au sein d'une image.

Dans ce même esprit de synthèse basée sur des patchs exemples, les travaux de [CPT04] proposent de recopier de façon judicieuse un patch de pixels. L'ordre de recopie des pixels du patch est tel qu'il permet de privilégier les forts gradients : cela permet ainsi de propager la texture tout en conservant les structures linéiques marquées au sein de l'image, tels que des contours.

#### 2.2.2.3 Synthèse de texture inverse

La synthèse inverse de texture correspond à des travaux présentés très récemment par Wei et al. [WHZ+08]. On parle de synthèse *inverse* car elle opère dans le sens opposé à la synthèse classique que nous avons évoquée jusqu'à présent.

Le principe de la synthèse est généralement de créer une grande image de texture à partir d'un petit patch d'une texture modèle. Le but de la synthèse inverse est de générer une texture

[2]Les algorithmes utilisés peuvent être l'algorithme de Dijkstra [Dij71] ou encore l'algorithme d'Edmonds et Karp [EK72], une version améliorée de l'algorithme de Ford et Fulkerson [FF62].

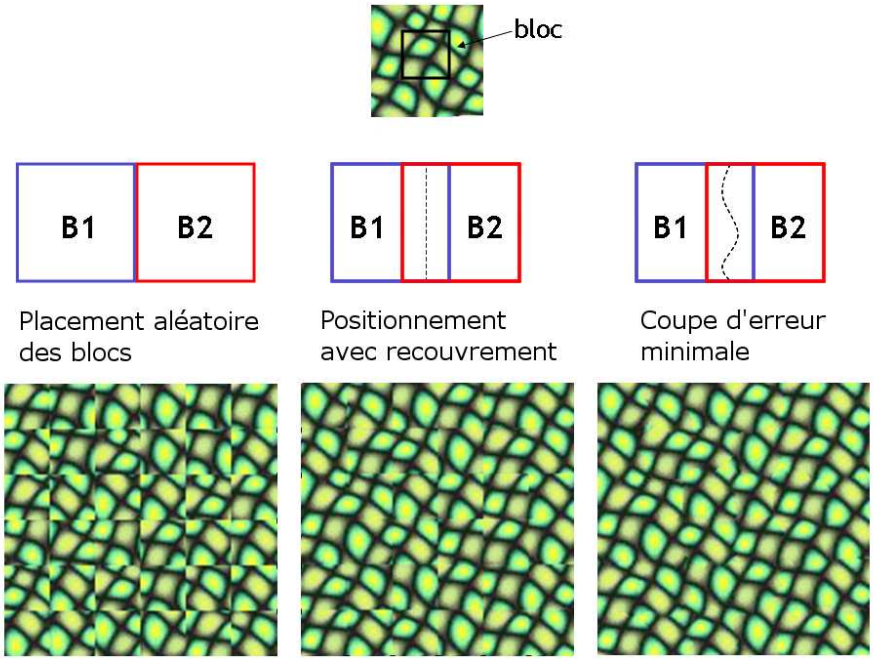



FIG. 2.7 – Synthèse par graphcut (Efros et Freeman)

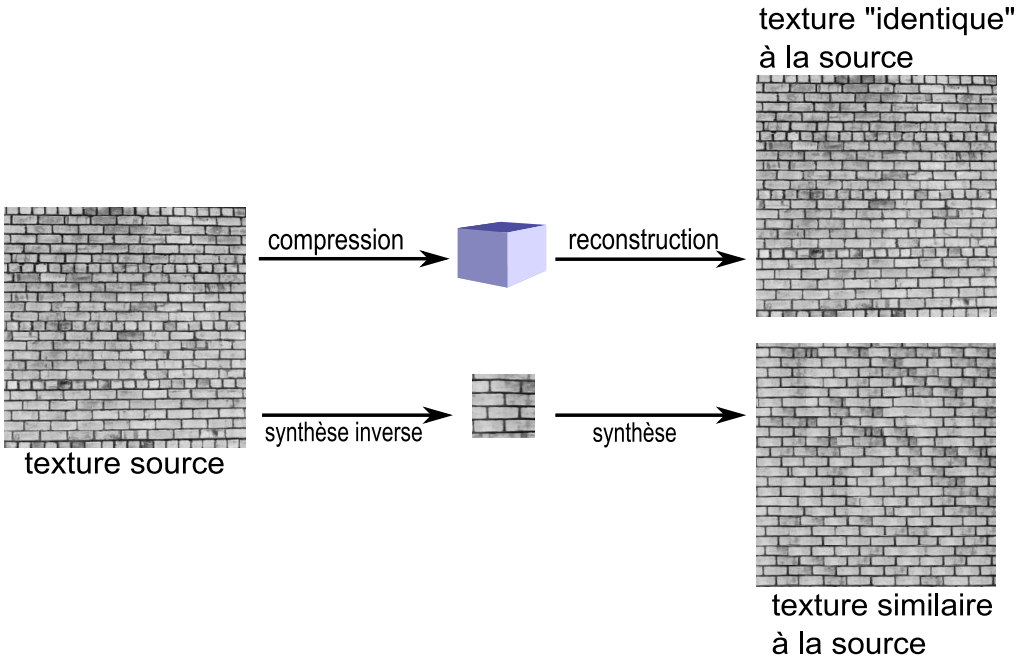



FIG. 2.8 – Synthèse inverse pour la compression

compactée, regroupant les principales caractéristiques texturelles d'une texture connue, de plus grande dimension. L'idée est bien sûr de réussir à condenser dans un petit échantillon les motifs nécessaires à la regénération d'une texture similaire de plus grande taille.

On devine aisément le potentiel que pourrait présenter cette technique dans un schéma de compression d'images. La différence majeure par rapport au schéma de compression classique, est que la texture obtenue n'est pas identique à la source (comme l'illustre la figure 2.8). Les techniques d'évaluation de la qualité basée sur une erreur pixel à pixel ne sont plus adaptées. Des critères perceptuels doivent être utilisés afin d'évaluer le degré de similarité entre la texture obtenue par synthèse inverse et la texture source.

### 2.2.3 Conclusion de la partie

Nous avons présenté différentes techniques de la littérature qui chacune met en œuvre des solutions au problème de synthèse de texture. Il ressort de l'analyse de ces travaux que les algorithmes développés sont nécessairement dépendants des textures traitées (stochastiques ou au contraire fortement ordonnées). Actuellement, il est encore complexe de s'affranchir de cette contrainte. Idéalement, il conviendrait d'effectuer une pré-analyse structurelle de la texture pour rendre la synthèse adaptative. Il existe cependant des techniques de natures différentes qui incorporent au sein même des algorithmes une analyse des composantes significatives de la texture.

*Nota bene* ·····

> *L'idée est de ne plus se contenter de recopier des pixels issus d'une texture modèle mais plutôt de «voir» la texture et ensuite de l'interpoler.*

## 2.3 L'*inpainting* d'images

### 2.3.1 Introduction

Ces dernières années, on a vu apparaître de nombreux travaux sur la thématique de l'*inpainting*. Ce domaine peut facilement se rattacher à celui de la restauration d'images. Sa dénomination illustre très bien la philosophie générale. L'*inpainting* consiste à faire le même travail que celui du peintre qui retoucherait de quelques coups de pinceaux des zones imparfaites ou manquantes de son tableau. Il s'agit ainsi de propager une texture existante, en général, sur une zone relativement restreinte, avec la contrainte que les modifications soient perceptuellement indétectables. Les techniques sont très variées, c'est pourquoi nous n'évoquerons ici que les principales.

### 2.3.2 Synthèse de la géométrie

L'approche variationnelle regroupe les méthodes par équations aux dérivées partielles (EDP) basées sur des équations de diffusion ou de transport. L'idée est de diffuser de manière anisotropique des pixels connus, au sein de la zone où l'on souhaite faire la synthèse. La propagation est anisotropique et se fait donc le long de directions particulières. Les EDP sont ainsi efficaces pour étendre une géométrie mais peu propices à la diffusion de la texture. Le signal texturé obtenu via ce genre de méthode correspond à un simple moyennage des pixels voisins (signal isotropique). Les EDP sont donc généralement exploitées en amont de la synthèse de texture pour d'abord recréer la structure géométrique de l'image.

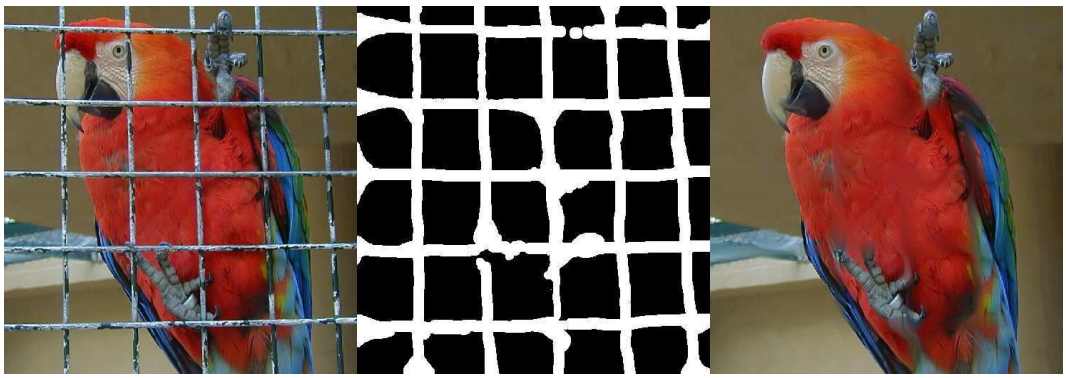


FIG. 2.9 – Exemple d'*inpainting* [TD03]. A gauche, l'image source ; au centre, le masque d'*inpainting* ; à droite, le résultat après *inpainting*

Nous pouvons citer les travaux [TD03] qui trouvent des applications dans la restauration d'images, le réajustement des couleurs, la suppression des effets blocs et bien sûr, l'*inpainting* dont un exemple de résultat est illustré en figure 2.9.

### 2.3.3 Synthèse combinée de la géométrie et de la texture

Les EDP étant limitées en terme de reconstruction de textures, les recherches se sont ensuite orientées vers la combinaison de deux approches : l'*inpainting* par diffusion pour étendre la géométrie et la synthèse de texture pour les zones texturées. La première permet de propager la structure générale de l'image et la seconde permet de créer la texture via des algorithmes de type template matching ou une méthode supervisée basée pixel. Il est particulièrement judicieux de combiner les deux approches, car il demeure complexe pour une image naturelle de bien délimiter les contours d'une texture. Ainsi, la prise en compte de la structure géométrique permet de ne pas effacer les contours lors du processus de synthèse.

FIG. 2.10 – Exemple de séparation : à gauche, l'image originale ; au centre, la géométrie ; à droite, la texture

Par exemple, les auteurs de [BVSO03] proposent de décomposer l'image en la somme de deux images fondamentales : l'une représentant les structures géométriques, tandis que la seconde ne contient que la texture (figure 2.10). Sur chacune de ces images on applique l'algorithme de diffusion et celui de synthèse respectivement. Puis, on récupère l'image restaurée en faisant la somme des deux images. Voici illustré en figure 2.11 un exemple de résultat obtenu avec cette méthode.

Les auteurs utilisent tout d'abord un algorithme de minimisation de la variation totale[3][VO03] qui a pour vocation d'extraire les contours d'une image, l'algorithme étant sensible aux variations de luminosité des pixels. Pour la partie synthèse, ils choisissent d'utiliser la technique d'Efros et Leung [EL00] évoquée en section 2.2.2.1. Bien sûr, d'autres algorithmes de synthèse peuvent être utilisés. Enfin, la dernière étape, consistant au processus d'inpainting lui-même, a été réalisée dans ces travaux par une propagation des pixels le long des isophotes, les lignes dont le changement de luminance est minimal [BSCB00].


FIG. 2.11 – Exemple de résultats d'*inpainting* par séparation des composantes fondamentales [BVSO03]

## 2.4 Étude harmonique

### 2.4.1 Seuillage itératif

Les travaux [Gul06a, Gul06b] abordent le problème de l'*inpainting* d'images comme une tâche de débruitage des données. La technique est basée sur l'étude d'une décomposition du signal local connu entaché des pixels manquants, sur une base de fonctions supposée mener à une représentation parcimonieuse. On distingue alors les coefficients nuls et ceux de faibles amplitudes, au niveau de la zone manquante.

L'enjeu de la technique consiste à déterminer la valeur de ces coefficients de faible amplitude via des seuillages itératifs, en tenant compte de contraintes de parcimonie. Les données inconnues sont alors estimées en s'appuyant sur la connaissance des données contenues dans le voisinage proche de la zone où les pixels sont manquants.

### 2.4.2 Analyse en composantes morphologiques (MCA)

Le désavantage des techniques présentées en section 2.3 est de réaliser le processus de séparation et celui de la synthèse via deux algorithmes distincts. Les travaux de [SED05] proposent une alternative à cette décomposition en combinant les deux traitements en un seul au sein d'un unique algorithme. La technique proposée repose sur une décomposition en composantes morphologiques [SED04], basée sur les représentations parcimonieuses, que nous exposons au chapitre 3.

L'hypothèse de base est de considérer que pour toute structure du signal il existe un dictionnaire de fonctions caractéristiques, à même de reconstituer cette structure particulière. Le signal est ainsi décomposé sur un méta-dictionnaire constitué d'un sous-dictionnaire pour

[3]Ils utilisent le modèle introduit par [ROF92] puis étendu par [Mey01] via un modèle original d'image texturée .

représenter la texture et d'un autre sous-dictionnaire pour récupérer la structure géométrique. Celui choisi pour représenter la texture est une base DCT. Quant à celui retenu pour récupérer les contours, il est constitué de *curvelets*. Il est important que les deux dictionnaires soient mutuellement incohérents[4] pour envisager une séparation dans de bonnes conditions. Nous verrons par la suite, au chapitre 3, que ce sont des algorithmes, comme celui du *Basis Pursuit Denoising*, qui ont pour but d'assurer la meilleure décomposition sur ce méta-dictionnaire. En l'occurrence, il s'agit de trouver parmi l'ensemble des solutions possibles, la plus parcimonieuse.

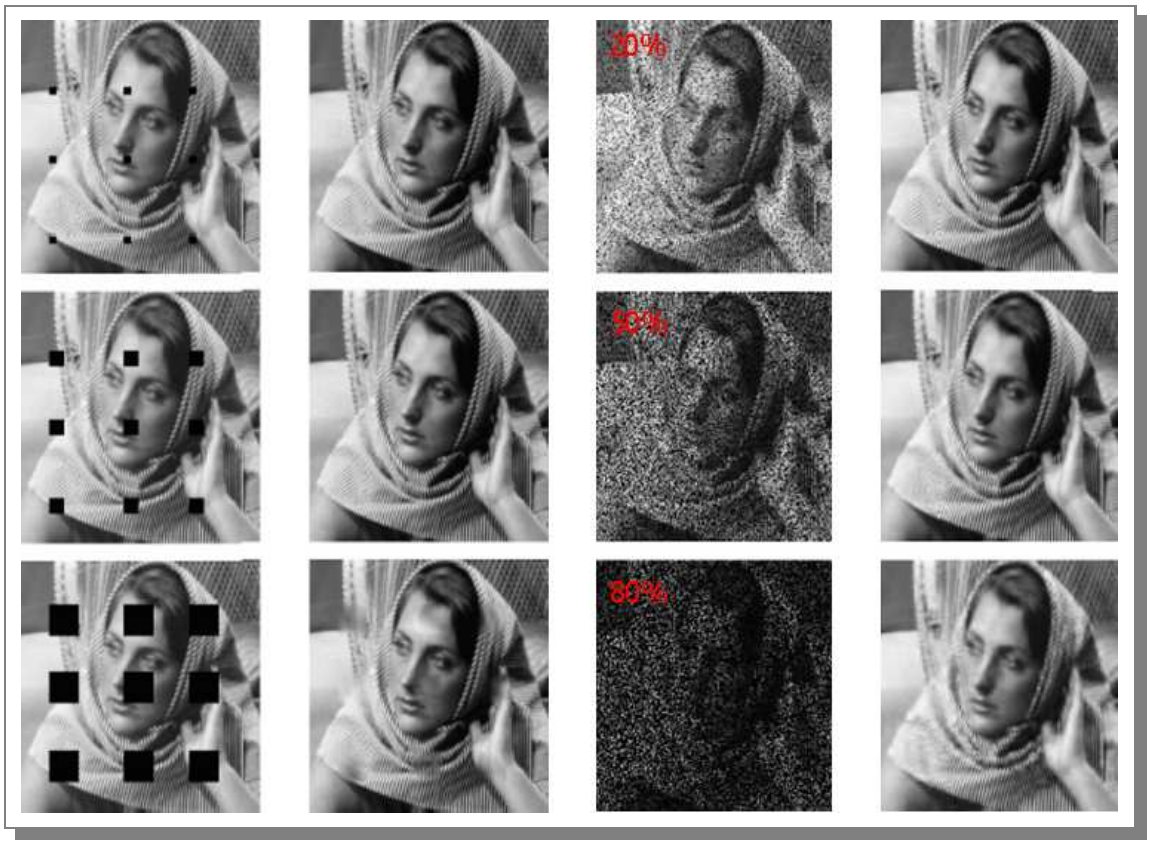



FIG. 2.12 – Exemples de résultats d'inpainting via [SED04]

### 2.4.3 Apprentissage de dictionnaire

Les performances de ce type d'approche, comme nous allons le voir tout au long de ce manuscrit, sont très dépendantes du choix des fonctions rassemblées dans le dictionnaire. Il faudrait en théorie avoir autant de sous-dictionnaires que de composantes morphologiques contenues dans une image. Les travaux de [PFS07] proposent une extension des travaux de [SED05] via l'utilisation de dictionnaires adaptés, *i.e.* appris sur un vaste panel d'images exemples, pour chacune des composantes morphologiques.

L'apprentissage de dictionnaire utilisé dans ce contexte, introduit une forte adaptabilité qui permet, à terme, de modéliser des motifs texturés complexes.

Dans le cadre des dictionnaires d'apprentissage, nous pouvons aussi citer les travaux de [EA06], basés sur un algorithme, le K-SVD, qui alterne entre une étape de décomposition

[4]On dit que deux dictionnaires sont mutuellement incohérents lorsque les corrélations entre les atomes d'un dictionnaire avec ceux de l'autre sont très faibles.

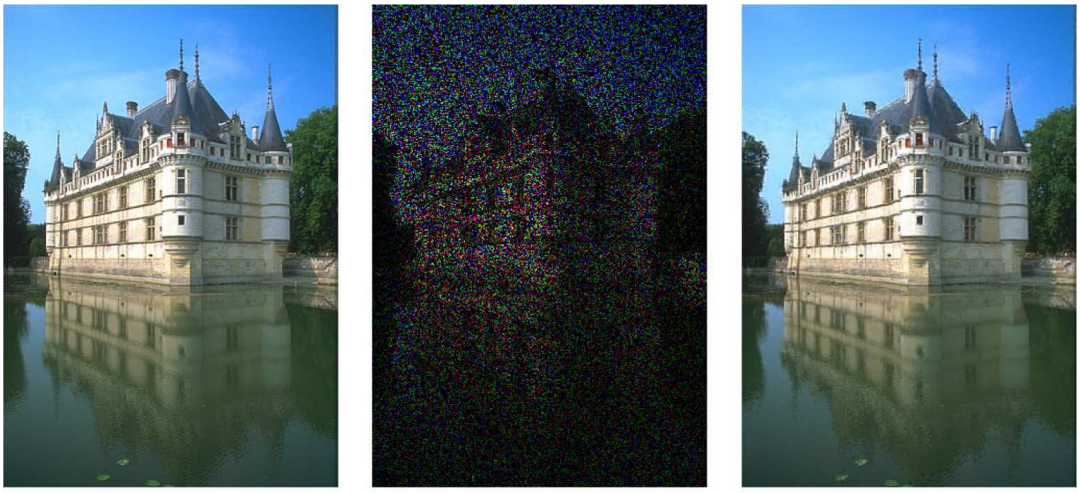



FIG. 2.13 – Exemples de résultats d'*inpainting* via le K-SVD (étendu à la couleur). L'image (a) correspond à l'image source ; (b) est l'image bruitée où 80 % des données ont été retirées ; (c) présente le résultat de cette technique d'*inpainting*

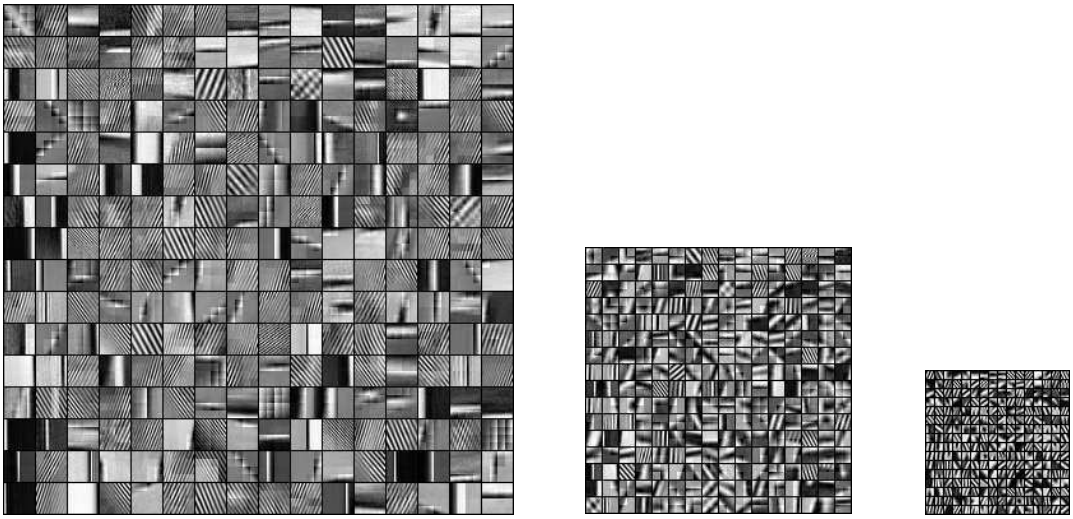


FIG. 2.14 – Dictionnaires sur trois niveaux de résolution appris sur l'image *Barbara*

parcimonieuse et de mise à jour du dictionnaire. Jusqu'à présent, ces travaux présentent des résultats essentiellement pour des applications de débruitage et certaines pour l'*inpainting* (Figure 2.13). Les auteurs ont également étendu leur technique en proposant une approche d'apprentissage multirésolution [MSE07]. La figure 2.14 présente des exemples de dictionnaires, à différentes résolutions, appris sur l'image *Barbara*.

### 2.4.4 Algorithme EM et analyse harmonique

Toujours dans la même optique, les travaux de [FS05] proposent une solution au problème d'*inpainting* d'images qui allie deux domaines : celui des statistiques et celui de l'analyse harmonique.

Leur solution est basée sur l'algorithme EM où le critère du maximum de vraisemblance est posé selon le formalisme des représentations parcimonieuses. L'approche est très similaire au K-SVD dans la mesure où l'algorithme alterne une phase de décomposition parcimonieuse puis une phase de mise à jour de l'ensemble des données estimées, en fonction des nouvelles données calculées. En revanche, il n'y a pas de mise à jour de dictionnaire car ils travaillent à dictionnaire fixé.

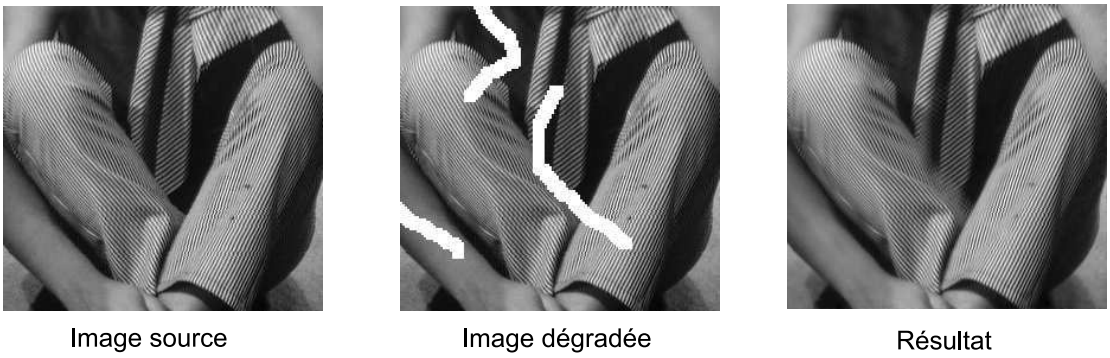



FIG. 2.15 – Exemples de résultats d'*inpainting* proposé par [FS05]

## 2.5 Conclusion

Nous avons mis en avant dans ce chapitre les différentes approches de la littérature en matière de synthèse de texture. La richesse des techniques déployées illustre à quel point il est difficile de restituer toute la complexité d'une image naturelle. Il n'est pas aisé de distinguer une méthode meilleure qu'une autre. Certaines ont une approche statistique et voient une texture comme un agencement probabiliste des pixels. L'enjeu de cette synthèse est de déterminer la *probabilité de présence* d'un pixel pour reproduire la texture. D'autres approches ont une vision plus *particulaire* de la texture : celles-ci cherchent à former un tout par un agencement pixel à pixel. Enfin, d'autres méthodes abordent la synthèse de texture sous un aspect *ondulatoire*. Le signal image peut se modéliser comme une combinaison judicieuse de formes d'ondes, qu'il reste, bien sûr, à définir. Les travaux de cette thèse se sont orientés vers cette dernière approche qui vise à percevoir une texture comme la réunion de signaux élémentaires. Nous aborderons dans le chapitre suivant la description des représentations parcimonieuses qui permettent notamment de faire une analyse harmonique d'un signal texturé.

# Chapitre 3

# Représentations parcimonieuses : état de l'art

*La prodigalité conduit à l'arrogance, et la parcimonie à l'avarice. L'arrogance est pire que l'avarice.*
Confucius

---

Dans le chapitre 1, nous avons souligné l'enjeu de la décorrélation du signal dans le domaine transformé. Nous allons voir dans ce chapitre que les représentations dites parcimonieuses s'inscrivent tout naturellement dans cette problématique. Le succès de telles représentations repose sur deux points fondamentaux. Le premier est leur habilité à extraire d'un groupe d'informations ou d'un ensemble de signaux, les structures significatives de ce signal. Le deuxième point est de réussir à générer une version compactée du signal d'origine. La technique consiste à puiser dans un vaste ensemble de signaux élémentaires, appelés **atomes** (terminologie introduite par Mallat et Zhang [MZ93]), les éléments qui linéairement combinés entre eux, formeront la représentation parcimonieuse du signal. Les atomes sont choisis au sein d'un ensemble *redondant* de fonctions que l'on nomme habituellement **dictionnaire**.

Les représentations parcimonieuses se sont révélées être une technique particulièrement performante en pratique. La recherche d'approximations parcimonieuses du signal s'est révélée fort utile dans les domaines de la compression, du débruitage, de la séparation de sources ou encore dans le domaine de l'analyse en composantes indépendantes.

## 3.1 Introduction à la parcimonie

Comme nous l'avons précédemment évoqué, la recherche de la parcimonie est utile à de nombreux domaines à des fins d'analyse, de modélisation ou encore d'identification. En effet, il est souvent pratique dans le domaine du traitement du signal de représenter l'information dans un autre espace plus propice à l'analyse ou aux manipulations diverses. Classiquement, les signaux sont décomposés dans une base de l'espace sur laquelle la décomposition est *unique*.

De manière plus générale, on définit un vecteur comme étant parcimonieux si la majorité de ses coefficients sont nuls. Ou plus exactement, qu'un ensemble de signaux de dimension $n$ de $\mathbb{R}^n$ est $k$-parcimonieux dans une base orthogonale de dimension $n >> k$, si on peut représenter avec une bonne approximation, un quelconque de ces signaux, à l'aide d'environ $k$ composantes de cette base. La figure 3.1 est une illustration schématique pour présenter quelques définitions : celle d'un signal compressé, et comparativement, celle d'un signal parcimonieux. Notons bien que pour un

signal compressé ou un signal parcimonieux, on peut obtenir des vecteurs de même dimension $k$, mais ils n'auront pas la même représentativité du signal.

On cherchera à construire la représentation parcimonieuse d'un signal à partir de fonctions définies dans un espace *redondant*, la décomposition la plus parcimonieuse. Cela permet d'obtenir une décomposition originale d'un signal sur un dictionnaire, faisant intervenir le moins d'éléments possibles.

Les représentations parcimonieuses offrent ainsi un degré de liberté supérieur aux transformations usuelles grâce à cette flexibilité inhérente à l'usage du dictionnaire et plus précisément à la grande *variété* des atomes qui le composent.

Notons bien que les représentations parcimonieuses n'offrent qu'une représentation **approximative** du signal, à la différence des transformations usuelles réversibles. L'enjeu reste néanmoins l'obtention de la solution la plus parcimonieuse, parmi celles ayant la même erreur de reconstruction, quel que soit le contexte de travail. En effet, la convergence vers la dite solution optimale peut s'avérer complexe dans le cas de signaux bruités. Cette problématique a fait l'objet d'études approfondies [Fuc04, DET06, Tro03], notamment dans le cadre de la détection de signal.

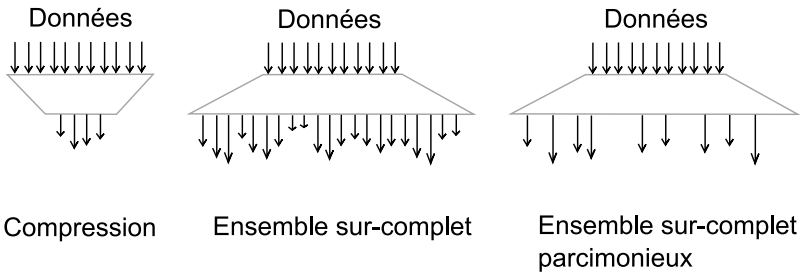



FIG. 3.1 – Modélisation de la parcimonie

## 3.2 Problème à résoudre

### 3.2.1 Problématique

Trouver la représentation parcimonieuse d'un signal consiste à obtenir la meilleure représentation du signal, *i.e.* la plus parcimonieuse parmi celles ayant la même erreur de reconstruction. Cette représentation est constituée d'un faible nombre d'atomes qui ont été choisis parmi un vaste ensemble de signaux élémentaires, le dictionnaire[1]. Générer une représentation parcimonieuse ne correspond pas, à proprement parlé, à une projection sur une base.

Notons $y$ le vecteur représentatif du signal source de dimension $m$ et $A \in \mathbb{R}^{m \times n}$ le dictionnaire, avec $m << n$. Il existe une infinité de solutions quant au choix du vecteur $x$ de dimension $n$, tel que :

$$y = Ax$$

Le but des représentations parcimonieuses est de trouver parmi l'ensemble des solutions possibles, celles qui sont parcimonieuses i.e. celles pour lesquelles le vecteur $x$ a seulement un faible nombre

[1]Le dictionnaire est aussi appelé base *redondante* car le nombre d'atomes est supérieur à la dimension de l'espace formé par le signal et sont donc linéairement dépendants. Ainsi, utiliser le mot *base* est un abus de langage par définition d'une base dans un espace linéaire.

de coefficients non nuls. Le problème à résoudre est donc le suivant :

$$\mathcal{P}_0 : \qquad \min_x \|x\|_0 \quad \text{sous} \quad y = Ax \tag{3.1}$$

où $\|.\|_0$ désigne le nombre de coefficients non nuls.

La minimisation exacte de la norme $l_0$ est un problème NP-complet [Nat95] qui n'a pas de solutions pratiques. Ceci signifie qu'obtenir la solution reviendrait à résoudre un problème combinatoire, *i.e.* tester toutes les combinaisons d'atomes possibles, méthode bien trop complexe dans un espace de grande dimension et en général, il faudrait de toute façon, $m$ composantes, ce qui est trop dans ce contexte.

On considère donc plutôt le problème suivant : il s'agit de rechercher la solution la plus parcimonieuse tout en tolérant une erreur admissible de reconstruction, notée $\rho$ :

$$\widetilde{\mathcal{P}}_0 : \qquad \min_x \|x\|_0 \quad \text{sous} \quad \left\|y - Ax\right\|_2 \leq \rho \tag{3.2}$$

où $\|.\|_2$ est la norme euclidienne $l_2$ : $\|x\|_2 = \sum_{i=1}^{N} \sqrt{|x_i|^2}$.

La solution de ce problème d'optimisation est le vecteur parcimonieux $x$, dont le nombre de coefficients non-nuls est minimal. Le vecteur $x$ conduit à une approximation dont l'erreur de reconstruction est inférieure ou égale à $\rho$. Ce problème est cependant toujours trop difficile à résoudre et on ne le considère jamais.

### 3.2.2 Distinction de deux axes d'étude

Faire face à la problématique (3.1) soulève donc potentiellement deux difficultés :

- trouver la représentation parcimonieuse pour un dictionnaire $A$ donné ;
- trouver le dictionnaire le mieux adapté au type de signal traité.

La résolution du problème sous-déterminé se fait via des algorithmes que nous décrivons à la section 3.3. Ils tendent à recouvrer une approximation de la solution exacte, la plus acceptable possible : à la fois en terme d'erreur de reconstruction (cf problème 3.2) et également en terme de parcimonie.

Au delà de la résolution mathématique du critère se pose aussi la question du choix du dictionnaire. On peut dégager à nouveau deux points essentiels. Le premier concerne la redondance du dictionnaire. Quels sont les avantages ? Quels sont les inconvénients ? Nous en discutons au paragraphe 3.4.1. Le second point porte sur la nature des atomes qui composent le dictionnaire. Les atomes doivent nécessairement se prévaloir d'une forte corrélation avec le signal pour espérer le représenter parcimonieusement. Nous présentons au paragraphe 3.4.2 quelques exemples d'atomes caractéristiques, ainsi que leurs spécificités intrinsèques.

Etant placé dans un ensemble redondant de représentations possibles, on choisit de ne garder que la plus parcimonieuse d'entre toutes. La parcimonie est directement liée au degré de corrélation des atomes et du signal.

## 3.3 Algorithmes de décomposition parcimonieuse

Nous décrivons dans cette partie les stratégies algorithmiques développées pour trouver le jeu de coefficients le plus parcimonieux, tout en respectant le critère qui a été choisi. Nous présentons

tout d'abord dans la section 3.3.1 certains algorithmes itératifs qui visent à obtenir la solution du problème $\widetilde{\mathcal{P}}_0$. Ces algorithmes ne cherchent pas exactement à résoudre le problème en trouvant l'approximation optimale. Ils raffinent progressivement l'approximation faite du signal par une procédure itérative. Le deuxième type d'algorithmes que nous présentons en section 3.3.2 sont des algorithmes de programmation linéaire et quadratique qui visent à résoudre de manière exacte le problème du *Basis Pursuit Denoising* $\mathcal{P}_{1,\lambda}^{\mathrm{D}}$ que nous allons présenter en section 3.3.2.2.

## 3.3.1 Approche sous-optimale

### 3.3.1.1 *Matching Pursuit*

Rappelons ici que nous considérons le problème général de la décomposition d'un signal $y$ sur un dictionnaire $A$ constitué d'un ensemble de vecteurs unitaires $\left\{a_\gamma\right\}_{\gamma\in\Gamma}$. Le dictionnaire est supposé redondant, ce qui laisse une grande liberté dans le choix du nombre, petit si possible, d'atomes $a_\gamma$ dont la combinaison linéaire formera une approximation du signal.

Le *Matching Pursuit* (MP) nommé ainsi par Mallat et Zhang en 1993 [MZ93] est un algorithme glouton, connu pour être une alternative à la recherche de la solution optimale. L'algorithme permet de trouver une approximation sous-optimale.

Le principe est donc de sélectionner pas à pas les atomes les plus corrélés avec le signal. Voici décrit ci-après, les étapes du *Matching Pursuit*.

– *La première approximation du signal* s'obtient en calculant le projeté orthogonal du signal observé $y$ sur l'atome qui lui est le plus corrélé, notons le $a_{\gamma_1}$. Comme $a_{\gamma_1}$ est normé, la pondération qui minimise la norme du résidu est $\left\langle y, a_{\gamma_1}\right\rangle$ et on a donc :

$$R_y^{(1)} = y - \left\langle y, a_{\gamma_1}\right\rangle a_{\gamma_1}$$

– *Lors des itérations suivantes*, on recherche l'atome le plus corrélé au résidu courant et on projette ce résidu sur l'atome sélectionné, noté $a_{\gamma_k}$. L'estimation obtenue par l'ajout de ce nouvel atome est ajoutée à l'estimation courante du résidu, pour former l'estimation courante du signal $y$. D'un point de vue plus formel, notons $R_y^{(k)}$ la valeur du résidu au pas $k$ et $\widehat{y}^{(k)}$ l'approximation courante de $y$ qui s'écrit alors :

$$\widehat{y}^{(k)} = \widehat{y}^{(k-1)} + \left\langle R_y^{(k-1)}, a_{\gamma_k}\right\rangle a_{\gamma_k}$$

Dans la procédure algorithmique que nous venons de décrire, rien n'exclut le fait qu'un atome puisse être sélectionné plusieurs fois. Le nombre d'itérations $k$ n'est donc pas nécessairement égal au nombre total d'atomes $a_\gamma$ sélectionnés pour la représentation $x$. L'intérêt majeur du *Matching Pursuit* est sa grande simplicité d'implémentation et sa relative rapidité d'exécution. Contrairement à d'autres algorithmes que nous présentons par la suite, il ne nécessite aucune inversion matricielle. Cependant, cette simplicité a un inconvénient : il peut falloir un grand nombre d'itérations pour converger vers une solution, fait d'autant plus vrai que dans certains cas un atome déjà sélectionné peut à nouveau l'être.

### 3.3.1.2 *Orthogonal matching pursuit*

L'*Orthogonal Matching Pursuit* (OMP) se base sur le même principe que le MP : sélectionner pas à pas les atomes les plus corrélés au signal pour tendre vers une approximation de la solution

**Algorithme 1** Algorithme du *Matching Pursuit*

**ENTRÉES :** Le signal source $y$, le dictionnaire $A$ et le seuil $\rho$

**INITIALISATION :** $\widehat{y}^{(0)} = 0, R_y^{(0)} = y$ et $k = 1$

**Tant que** $\left\|R_y^{(k-1)}\right\|^2 \geq \rho$ **faire**

– Recherche de l'atome le plus corrélé : $\gamma_k = \arg\max_{\gamma} \left|\left\langle R_y^{(k-1)}, a_\gamma \right\rangle\right|$

– Calcul du nouveau coefficient : $x_{\gamma_k} = \left\langle R_y^{(k-1)}, a_{\gamma_k} \right\rangle$

– Mise à jour des données :

1. de l'estimée : $\widehat{y}^{(k)} = \widehat{y}^{(k-1)} + x_{\gamma_k} a_{\gamma_k}$
2. du résidu : $R_y^{(k)} = y - \widehat{y}^{(k)} = R_y^{(k-1)} - x_{\gamma_k} a_{\gamma_k}$
3. du vecteur parcimonieux : $x[\gamma_k] \leftarrow x_{\gamma_k}$

**Fin Tant que**

au problème $\widetilde{\mathcal{P}}_0$. La différence réside dans la mise à jour des coefficients. L'objectif de l'OMP est de pallier la faille du MP, qui, comme nous venons de le voir, n'empêche pas la sélection multiple d'un même atome.

Afin d'y parvenir, l'OMP recalcule à chaque pas de l'algorithme la valeur de l'estimée $\widehat{y}$. Le MP se base sur la mise à jour du résidu : à chaque itération, on retire au résidu une contribution du *seul* nouvel atome sélectionné. Pour l'OMP, c'est différent : on évalue la reconstruction courante $\widehat{y}^{(k)}$ à chaque fois, ce qui signifie que l'on recalcule tous les coefficients jusqu'alors sélectionnés. L'entrée d'un nouvel atome dans la décomposition modifie l'espace engendré. Il est donc plus judicieux de projeter le signal $y$, non plus sur le seul nouvel atome, mais sur l'ensemble formé des atomes passés auquel s'ajoute le nouvel atome sélectionné.

Cette mise à jour de tous les coefficients pour chaque nouvel atome choisi pourrait se faire avec la procédure d'orthogonalisation de Gram-Schmidt. On génère donc avec cet algorithme une base orthogonale dont la dimension croît à chaque nouvelle sélection d'un atome. L'ensemble des atomes retenus formant une famille libre de l'espace, on exclut de fait la sélection d'un atome ayant déjà été ajouté à cette base.

Reprenons les mêmes notations que précédemment tout en introduisant deux nouvelles variables :

– $A_k$, la matrice contenant l'ensemble des atomes sélectionnés à l'itération $k$ : $A_k = \left[a_{\gamma_1} \dots a_{\gamma_k}\right]$.
– $x_k$, le vecteur contenant uniquement les coefficients non-nuls qui ont été calculés jusqu'au pas courant : $x_k = \left[x_{\gamma_1} \dots x_{\gamma_k}\right]$.

La projection du signal sur l'ensemble formé des atomes sélectionnés jusqu'au pas courant se fait via le calcul de la pseudo-inverse $A_k^+$. Comme nous l'avons précédemment remarqué, cette méthodologie empêche la sélection d'un atome qui aurait déjà été sélectionné. Cela a l'avantage de surpasser le *Matching Pursuit* en terme de nombre d'itérations mais nécessite une inversion matricielle à chaque itération. Il existe toutefois des simplifications à l'algorithme présenté qui consistent à déterminer la valeur de $A_k^+$ de manière récursive, *i.e.* en utilisant la valeur de $A_{k-1}^+$.

**Algorithme 2** Algorithme de l'*Orthogonal Matching Pursui*t

**ENTRÉES :** Le signal source $y$, le dictionnaire $A$ et le seuil $\rho$

**INITIALISATION :** $\widehat{y}^{(0)} = 0, R_y^{(0)} = y, A_0 = [.]$ et $k = 1$

**Tant que** $\left\|R_y^{(k-1)}\right\|^2 \geq \rho$ **faire**

– Recherche de l'atome le plus corrélé : $\gamma_k = \arg\max_{\gamma} \left|\left\langle R_y^{(k-1)}, a_\gamma \right\rangle\right|$

– Ajout du nouvel atome au sous-dictionnaire : $A_k = \left[A_{k-1} \quad a_{\gamma_k}\right]$

– Calcul des coefficients : $x_k = A_k^+ y$ avec $A_k^+ = \left(A_k^T A_k\right)^{-1} A_k^T$

– Mise à jour des données :

1. de l'estimée : $\widehat{y}^{(k)} = A_k.x_k$
2. du résidu : $R_k = y - \widehat{y}^{(k)}$

**Fin Tant que**

### 3.3.2 Approche globale

#### 3.3.2.1 *Basis Pursuit*

Comme la vraie parcimonie $\mathcal{P}_0$ n'est pas utilisable en pratique, on utilise la norme $l_1$ pour contraindre la parcimonie. La norme $l_1$ est définie de la manière suivante :

$$\|x\|_1 = \sum_{i=1}^{N} |x_i|$$

Cette norme est connue pour favoriser un grand nombre de coefficients nuls. Le problème se formule alors de la manière suivante :

$$\mathcal{P}_1 : \qquad \min_x \|x\|_1 \quad \text{sous} \quad y = Ax \tag{3.3}$$

Des travaux [DH01, Don04] ont montré que dans la majorité des cas, la minimisation de la norme $l_1$ permet effectivement d'obtenir la représentation la plus parcimonieuse. Ses propriétés de parcimonie n'étant toutefois pas égales à celles que l'on obtiendrait avec $l_0$. Chen, Donoho et Saunders [CDS98] ont donné le nom de *Basis Pursuit* (BP) au problème $\mathcal{P}_1$. C'est un problème d'optimisation convexe qui peut être reformulé [CDS98] sous la forme d'un programme linéaire [Dan63, GMW91].

#### 3.3.2.2 *Basis Pursuit Denoising*

La contrainte égalité du problème (3.3) est trop forte si on veut obtenir une représentation parcimonieuse. Il est par ailleurs réaliste de supposer que les données traitées sont entachées d'un bruit perturbateur. On suppose donc que les données d'observations sont de la forme :

$$y = b + e$$

où $e$ est un bruit blanc gaussien, $b$ les données sources inconnues et $y$ le signal bruité observé. Voici deux critères qui ont été proposés dans la littérature pour trouver une approximation parcimonieuse $\widehat{x}$ :

$$\min_x \frac{1}{2}\|y - Ax\|_2^2 \qquad \text{sous} \qquad \|x\|_1 \leq t \tag{3.4}$$

$$\min_x \|x\|_1 \qquad \text{sous} \qquad \|y - Ax\|_2^2 \leq \rho \tag{3.5}$$

– Le critère (3.4) fut introduit en 1996 par Tibshirani [DET96], approche dénommée LASSO pour *Least Absolute Shrinkage and Selection Operator*. La contrainte porte ici sur le nombre maximal de coefficients admis pour la représentation parcimonieuse $x$.

– Le second critère (3.5) est connu sous le nom de *Basis Pursuit Denoising* [Fuc97, Fuc98, CDS98]. A l'inverse du LASSO, on fixe l'erreur de reconstruction admissible et non le nombre maximal de coefficients. Il existe une autre formulation du *Basis Pursuit Denoising* :

$$\mathcal{P}_{1,\lambda}^{\mathrm{D}} : \qquad \min_x \frac{1}{2}\|y - Ax\|_2^2 + \lambda\|x\|_1 \tag{3.6}$$

Le paramètre $\lambda$ sert à ajuster le poids que l'on souhaite donner à chacun des deux éléments de la somme. Il permet d'établir un compromis entre la valeur de l'erreur de reconstruction (premier terme) et le nombre de coefficients non-nuls (deuxième terme).

Plus la valeur de $\lambda$ est grande, plus on privilégie la parcimonie au dépend de la qualité de la reconstruction. Inversement, si $\lambda$ est très petit, l'approximation obtenue sera de bonne qualité mais peu parcimonieuse.

#### 3.3.2.3 Algorithme du LARS

L'algorithme du LARS, *Least Angle Regression* a été utilisé et modifié par [EHJT04] pour résoudre le critère du LASSO (3.4). Nous avons vu précédemment que ce critère est une méthode d'estimation par les moindres carrés plus un terme de pénalisation, portant sur le nombre de coefficients non-nuls au sein du vecteur $x$.

L'algorithme recherche l'estimée itérativement en sélectionnant les atomes les plus corrélés au signal. A l'identique du *Matching Pursuit* ou de l'*Orthogonal Matching Pursuit*, à l'itération $k$, on raffine l'estimation du signal en ajoutant l'apport d'un nouvel atome fortement corrélé au résidu courant, $R_y^{(k)}$.

En revanche, la mise à jour du signal estimé $\widehat{y}^{(k)}$ se fait par le biais d'un critère *géométrique*. Supposons que deux atomes ont déjà été sélectionnés. La mise à jour de $\widehat{y}^{(2)}$ se fera le long de la bissectrice formée par les deux atomes sélectionnés. Ainsi au pas $k+1$, si $k$ atomes ont été retenus, l'estimée est mise à jour le long de la direction équiangulaire formée par les $k$ atomes retenus. Notons $d^{(k+1)}$ cette direction ; l'expression récursive de l'estimée est donc la suivante :

$$\widehat{y}^{(k+1)} = \widehat{y}^{(k)} + \gamma^{(k+1)} d^{(k+1)}$$

avec $\gamma^{(k+1)}$ un paramètre qui règle la contribution du nouvel atome.

A chaque itération, on choisit un nouvel atome de telle sorte que sa corrélation avec le résidu courant soit égale aux corrélations des précédents atomes sélectionnés avec le résidu courant. Pour atteindre cette contrainte, on calcule le pas $\gamma$ qui permet de déterminer de combien l'estimée doit se décaler dans la direction $d$ pour que le nouvel atome soit autant corrélé avec le résidu courant que les atomes déjà sélectionnés.

#### 3.3.2.4 Le filtre adapté global

*– Origines du filtre adapté global –*

Le filtre adapté global (ou GMF pour *Global Matched Filter*) est un algorithme initialement développé [Fuc01] pour résoudre des problèmes de détection et d'estimation, notamment pour des applications sonar. Il peut s'appliquer à partir du moment où les données d'observations $y$ peuvent se décomposer en une somme finie de $p$ termes de fonctions paramétriques connues $a(\theta_i)$, à laquelle s'ajoute un bruit $e$, supposé blanc :

$$y = \sum_{i=1}^{p} a(\theta_i)\, x_i + e$$

Pour déterminer la représentation de $y$ , on doit estimer à la fois le nombre $p$ de fonctions utiles, le scalaire ou vecteur $\theta$ qui paramétrise les fonctions $a(\theta)$ et les coefficients de pondération $x_i$ associés. La résolution de ce problème se fait en général grâce à l'estimateur au sens du maximum de vraisemblance, approche classiquement usitée en statistique pour estimer les paramètres $\theta$ dont dépendent les données. Dans le cas d'un bruit gaussien, l'estimation du maximum de vraisemblance revient à résoudre le problème suivant :

$$\min_{\theta_i, x_i} \|y - \sum_{i=1}^{p} a(\theta_i)\, x_i\|_2^2 \quad \text{pour p fixé}$$

Le GMF est une alternative à l'utilisation du maximum de vraisemblance pour traiter ce cas. L'idée est de discrétiser finement le paramètre $\theta$ afin de linéariser le problème et d'obtenir des vecteurs tels que $a_j = a(\theta_j)$. Pour un pas de discrétisation suffisamment petit, le nombre de fonctions $a_j$ devient supérieur à la dimension de l'espace formé par les données d'observations $y$. Le critère proposé pour estimer les données devient alors le suivant :

$$\min_{x \geq 0} \frac{1}{2}\|y - Ax\|_2^2 + h\|x\|_1 \quad , \quad h > 0 \tag{3.7}$$

où $h \in \mathbb{R}^+$ ajuste le degré de parcimonie de la reconstruction et $A$ est la matrice dont les colonnes sont les $a(\theta_j)$ précédents. Nous détaillerons par la suite la signification physique de ce paramètre $h$. Il a été démontré [Fuc01], que ce problème d'optimisation (3.7), est en fait équivalent au problème suivant :

$$\min_{x} \frac{1}{2}\|Ax\|_2^2 \qquad \text{sous} \qquad \left\|A^T (y - Ax)\right\|_\infty \leq h \tag{3.8}$$

qui permet, quant à lui, de faire une interprétation physique évidente. Le problème (3.8) stipule que l'on recherche la représentation d'énergie minimale qui, à l'optimum, conduit à un résidu dont la corrélation avec n'importe quel atome du dictionnaire est inférieure au seuil $h$. L'algorithme utilisé pour résoudre le critère (3.7) se base sur une approche *homotopique* et s'avère identique (ou presque) au LARS.

*– Approche homotopique –*

Dans le cadre qui nous intéresse, à savoir la recherche de la solution optimale du problème (3.7), il a été remarqué [OPT00] que la solution optimale évoluait linéairement sur des intervalles de $h$ où le nombre de coefficients non-nuls de la solution n'évolue pas.

L'idée directrice revient ainsi à trouver l'ensemble des intervalles de $h$ sur lequel la solution évolue linéairement jusqu'à l'optimum. Il faut repérer des non-linéarités qui seront le signe d'une

modification du support de la solution et donc de l'apparition d'un nouvel atome.

*– Description algorithmique –*

Pour trouver la solution au problème (3.7), une approche classique consiste à le transformer en un programme quadratique. Si on pose $x_i^+ = \max(x_i, 0)$ et $x_i^- = \max(-x_i, 0)$, on peut remplacer $x_i$ par $x_i^+ - x_i^-$ et $|x_i|$ par $x_i^+ + x_i^-$ et ainsi obtenir un programme quadratique. La technique utilisée ici est différente. Elle est basée sur l'utilisation d'une approche homotopique appliquée aux conditions d'optimalités. Soit $\partial \|x\|_1$ le sous-gradient de $\|x\|_1$ :

$$\begin{aligned}\partial\|x\|_1 &= \left\{u | u^T x = \|x\|_1, \|u\|_\infty \leq 1\right\} \\ &= \{u | u_i = \text{signe}(x_i) \text{ si } x_i \neq 0 \text{ et } |u_i| \leq 1 \text{ sinon}\}\end{aligned}$$

où $\text{signe}(x_i) = \pm 1$. Une condition nécessaire et suffisante pour que $x^*$ soit le minimum global de (3.7) est que le vecteur nul soit un sous-gradient du critère en $x^*$ :

$$\exists u \in \partial\|x\|_1 \text{ tel que} A^T (y - Ax^*) + h^* u = 0 \tag{3.9}$$

Si on connait $x^*$, cette relation donne la valeur de $u$ associée. Cette relation est ensuite utilisée pour propager l'optimum $(x^*, u)$ au voisinage de $h^*$ pour lequel l'optimum est connu. Afin de rendre utilisable en pratique la condition (3.9), distinguons les coefficients non-nuls de $x^*$ de ceux qui sont à zéro. On note $\bar{x}^*$ le vecteur extrait de $x^*$ contenant les coefficients non-nuls et $\bar{\bar{x}}^*$ le vecteur regroupant les composantes nulles. De même, notons $\bar{A}$, la matrice extraite de $A$ dont les colonnes retenues correspondent aux coefficients non-nuls de $\bar{x}^*$ et $\bar{\bar{A}}$, celles correspondant aux coefficients nuls. On a donc $Ax^* = \bar{A}\bar{x}^*$. L'équation (3.9) peut ainsi être reformulée de la manière suivante :

$$-\bar{A}^T (y - \bar{A}\bar{x}^*) = h\text{signe}(\bar{x}^*) \tag{3.10}$$

$$-\bar{\bar{A}}^T (y - \bar{A}\bar{x}^*) = h\bar{\bar{u}}^* \tag{3.11}$$

Ce que l'on cherche à déterminer, est comment évoluent $\bar{x}^*$ et $\bar{\bar{u}}$ lorsque $h$ varie localement, car la décomposition qui vient d'être faite est valide tant qu'aucun coefficient de $\bar{x}^*$ ne s'annule et qu'aucune composante de $\bar{\bar{u}}$ devienne égale à $\pm 1$ [Fuc01] . De l'équation (3.10), il en ressort une expression explicite de $\bar{x}^*$ qui met en exergue sa dépendance au terme $h$. Nous noterons ainsi $\bar{x}^*(h)$ l'optimum, plutôt que $\bar{x}^*$. Cette expression substituée dans (3.11) permet d'obtenir une formulation de $\bar{\bar{u}}$ :

$$\bar{x}^*(h) = \bar{A}^+ y - h(\bar{A}^T\bar{A})^{-1}\text{signe}(\bar{x}^*) \tag{3.12}$$

$$\bar{\bar{u}}(h) = \frac{1}{h}\bar{\bar{A}}^T y^\perp + \bar{\bar{A}}^T d \tag{3.13}$$

où $y^\perp = \left(I - \bar{A}\bar{A}^{+T}\right) y$ est le projeté de $y$ sur l'espace orthogonal à l'espace engendré par $\bar{A}$ et $d = \bar{A}^{+T}\text{signe}(x(h))$ avec $\bar{A}^+$ la pseudo-inverse de $\bar{A}$.

Les expressions (3.12) et (3.13) sont valides pour $h^*$. Cependant, tant qu'il n'y a pas de modification du support de $(\bar{x}(h), \bar{u}(h))$ les équations restent valides. L'enjeu est donc maintenant de déterminer les bornes de ces intervalles de validité.

Les bornes de l'intervalle sont obtenues à partir des équations 3.12 et 3.13. A chaque changement de support, il faut déterminer le nouvel intervalle de validité et il sera alors possible de retracer, intervalle par intervalle, l'évolution de $x(h)$ vers son optimum.

On construit l'optimum $x^*(h)$ ou plus exactement, ses composantes non nulles $\bar{x}^*(h)$ lorsque $h$ décroît. On procède de proche en proche en construisant une séquence d'intervalles adjacents $\left[h_{inf}^{(k)}\,,\, h_{sup}^{(k)}\right]$ avec $h_{sup}^{(k)} = h_{inf}^{(k-1)}$ pour $k$ croissant, dans lesquels $\bar{x}(h)$ et $\bar{u}(h)$ sont donnés par les expressions 3.12 et 3.13 avec $\bar{A} = \bar{A}^{(k)}$, signe$(\bar{x}(h)) = s^{(k)}$ et $d^{(k)} = \bar{A}^{+T} s^{(k)}$. Le critère d'arrêt est valide dès que l'on atteint l'intervalle qui contient $h^*$, valeur de $h$ fixé dans le critère.

Nous décrivons dans la procédure ci-contre, les étapes de l'algorithme permettant de déterminer les bornes de ces intervalles. Notons $n^{(k)}$ le nombre de colonnes dans $\bar{\bar{A}}^{(k)}$.

*– Interprétation physique du paramètre h –*

Le paramètre $h$ ($> 0$) est un seuil que l'on doit fixer *a priori*, en fonction du degré de parcimonie souhaité. Il ajuste en effet le taux de pénalisation introduit par le terme $\|x\|_1$ du problème (3.8). Lorsque $h$ augmente, le nombre de composantes nulles de $x(h)$ augmente. Reprenons la formulation duale (3.8) du problème :

$$\min_x \frac{1}{2}\|Ax\|_2^2 \qquad \text{sous} \qquad \left\|A^T(y - Ax)\right\|_\infty \leq h$$

A l'optimum, on a ainsi une représentation de $y$ d'énergie minimale et dont le résidu $(y - Ax)$ possède une corrélation avec tout autre atome inférieure au seuil $h$ et dont la corrélation avec les atomes sélectionnés est égale exactement à $\pm h$. Le paramètre $h$ représente une corrélation maximale autorisée. $h$ agit finalement comme un seuil de détection mais, contrairement aux algorithmes de poursuite précédemment présentés, il détecte les atomes $a_\gamma$ de manière globale ou simultanée.

Au lieu d'arrêter le *Matching Pursuit* avec $\|R_y\|_2^2 \leq \rho$, on pourrait utiliser ce critère (3.8) et arrêter d'itérer dès qu'aucun atome n'a une corrélation supérieure à $h$. La représentation donnée du MP est bien sûr bien moins parcimonieuse que celle obtenue avec le GMF.

## 3.4 Dictionnaires

### 3.4.1 Des bases orthonormales aux dictionnaires redondants

Le signal peut se représenter d'une infinité de manières via les colonnes du dictionnaire, les atomes $a_\gamma$. Le signal est décomposé sous la forme d'une combinaison linéaire pondérée de fonctions élémentaires :

$$y = \sum_{\gamma \in \Gamma} x_\gamma a_\gamma \tag{3.14}$$

où $x_\gamma$ sont les coefficients de pondération des atomes $a_\gamma$ et $\Gamma$ l'ensemble fini d'indices représentant les fonctions élémentaires choisies.

#### 3.4.1.1 Enrichir le dictionnaire

La nouveauté ces dernières années a été d'élargir le spectre des fonctions élémentaires, en se détachant des bases orthonormales usuelles [DH01, DE03, Fuc02, GN02, Dau04]. Le dictionnaire se compose alors de diverses fonctions, de natures différentes. Cette diversité est à l'origine de la

**Algorithme 3** Algorithme du *Global Matched Filter*

**ENTRÉES :** Le signal source $y$, le dictionnaire $A$ et le seuil $h$

**INITIALISATION :** – Si $h^* > \left\|A^T y\right\|_\infty$, $x^* = 0$ et arrêt de l'algorithme.

– Sinon :

- $i^{(1)} = arg\max_i \left(\left|A^T y\right|_i\right)$
- $\bar{A}^{(1)} = [a_{i^{(1)}}]$, $\bar{\bar{A}}^{(1)} = A \setminus a_{i^{(A)}}$, $n^{(1)} = n - 1$
- $s^{(1)} = \text{signe}\left[\left(A^T y\right)_{i^{(1)}}\right]$
- $h_{sup}^{(1)} = \left\|A^T y\right\|_\infty$

**ETAPE $k$ DE L'ALGORITHME :**

1. **Recherche de** $h_{inf}^{(k)} = h_{sup}^{(k)}$ :
   - CAS 1 : une composante de $\bar{\bar{u}}$ devient égale à $\pm 1$ pour $h_1^{(k)}$ :
     Pour $j = 1, 2, \ldots, n^{(k)}$ et $\epsilon = +1$ ou $-1$, on calcule :
     $$Val1_j(\epsilon) = \epsilon \frac{\left(\bar{\bar{A}}^{(k)T} y_{\perp^{(k)}}\right)_j}{1 - \epsilon\left(\bar{\bar{A}}^{(k)T} y^{(k)}\right)_j} ; \quad \begin{cases} \left(ind_1^{(k)}, \epsilon_1^{(k)}\right) = \arg\max_{j,\epsilon}\left(Val1_j(\epsilon)\right) \\ h_1^{(k)} = Val1_{ind_1^{(k)}}(\epsilon_1^{(k)}) \end{cases}$$
   - CAS 2 : une composante de $\bar{a}$ devient égale à 0 pour $h_2^{(k)}$.
     Pour $j = 1, 2, \ldots, n - n^{(k)}$ :
     $$Val2_j = \frac{\left(\bar{A}^{(k)^+} y\right)_j}{\left(\left(\bar{A}^{(k)T} \bar{A}^{(k)}\right)^{-1} s^{(k)}\right)_j} ; \quad \begin{cases} ind_2^{(k)} = \arg\max_j\left(Val2_j\right) \\ h_2^{(k)} = Val2_{ind_2^{(k)}} \end{cases}$$
   - Détermination du nouvel intervalle : $h_{inf}^{(k)} = \max\left(h_1^{(k)}, h_2^{(k)}\right)$
   - Si $\left(h_{inf}^{(k)} > h_{sup}^{(k)} \quad \text{ou} \quad h_{inf}^{(k)} < 0\right)$, prendre $h_{inf}^{(k)} = 0$.
2. **Mise à jour** :
   - **Si** $h^* \in ]h_{inf}^{(k)}, h_{sup}^{(k)}[$ alors nous sommes dans l'intervalle désiré et nous pouvons déterminer la solution qui minimise le critère :
     $$x^* = \bar{A}^{(k)^+} y - h^* \left(\bar{A}^{(k)T} \bar{A}^{(k)}\right)^{-1} s^{(k)}$$
     puis on s'arrête.
   - **Sinon**, mise à jour de $\bar{A}^{(k+1)}$, $\bar{\bar{A}}^{(k+1)}$ et $s^{(k+1)}$.
     - ◇ Si le cas 1 est valide : retrait dans $\bar{\bar{A}}^{(k)}$ de la colonne $ind1^{(k)}$, que l'on ajoute à $\bar{A}^{(k)}$ et $s^{(k+1)} = \left[s^{(k)T}, \epsilon_1^{(k)}\right]^T$, $n^{(k+1)} = n^{(k)} - 1$.
     - ◇ Si le cas 2 est valide : retrait dans $\bar{A}^{(k)}$ de la colonne $ind2^{(k)}$ et de la composante correspondante dans $s^{(k)}$, puis ajout de cette colonne dans $\bar{\bar{A}}^{(k)}$ et $n^{(k+1)} = n^{(k)} + 1$
     - ◇ puis $h_{sup}^{(k+1)} = h_{inf}^{(k)}$, $k = k + 1$ et on recommence à partir de : 1..

notion de redondance. En effet dans le cas classique d'une base orthonormale, la décomposition obtenue est unique puisque les vecteurs de base sont linéairement indépendants. En revanche, lorsqu'on enrichit le dictionnaire avec un surplus d'atomes, on s'expose au fait que le signal aura de multiples représentations possibles.

#### 3.4.1.2 Danger de la redondance

Il existe réellement un bémol quant à la recherche du dictionnaire *sur-complet.* Choisir un ensemble exhaustif de fonctions peut devenir un inconvénient si ces atomes ont une trop faible corrélation avec le signal. Dans le cas de la concaténation de deux bases orthonormales (ou non), l'intelligence repose dans le degré de complémentarité des dictionnaires concaténés. Il est judicieux de choisir comme deuxième dictionnaire un ensemble d'atomes qui permettra de reconstituer certaines des caractéristiques du signal, que l'autre dictionnaire ne sera pas en mesure de représenter parcimonieusement.

#### 3.4.1.3 Atomes corrélés au signal

Le choix des atomes dépend de l'application pour laquelle on destine l'utilisation des représentations parcimonieuses. Il s'est avéré utile, voire indispensable, d'avoir des atomes ayant les mêmes caractéristiques que le signal source que l'on cherche à modéliser. Prenons le cas des images, la modélisation d'un contour sera évidente si les atomes eux-mêmes possèdent une structure visuellement proche d'un contour. De même, se pose la question du facteur d'échelle : comment représenter efficacement *i.e.* trouver la représentation la plus parcimonieuse, d'un motif ayant cinq fois la taille des atomes ? Ainsi, plus le dictionnaire possède de fonctions structurelles variées, permettant la représentation des signaux à différentes résolutions, plus les chances d'obtenir une représentation parcimonieuse sont fortes.

#### 3.4.1.4 Dictionnaires adaptatifs

Au-delà de l'élaboration de tels dictionnaires, il existe également les méthodes d'apprentissage[PFS07]. Le dictionnaire est adaptatif car construit à partir des données sources elles-mêmes. L'avantage est bien sûr la grande liberté qu'offre l'adaptabilité pour représenter n'importe quel signal. Si l'on ne se pose plus la question du choix du type d'atomes, il se pose en revanche celle de la complexité de telles méthodes, qui, à l'heure actuelle, sont encore très lourdes.

### 3.4.2 Dictionnaires redondants

Afin de mieux cerner le caractère décisif du choix des atomes, nous présentons ci-dessous quelques atomes caractéristiques, ainsi que leurs spécificités intrinsèques.

#### 3.4.2.1 Atomes fréquentiels

*Atomes DFT*

Une représentation issue d'atomes de la transformée de Fourier génère une décomposition propice à l'analyse harmonique d'un signal. L'expression de la transformée de Fourier discrète mono-dimensionnelle, notée $F$, d'un signal $f$ de dimension $N$ est la suivante :

$$F_k = \sum_{n=0}^{N-1} f_n e^{-2i\pi \frac{kn}{N}}$$

Les atomes associés à la transformée de Fourier discrète sont les fonctions :

$$a_{n,k} = e^{-2i\pi\frac{kn}{N}}$$

Le dictionnaire ainsi créé se compose de $N$ formes d'ondes dont les atomes sont orthogonaux deux à deux. Pour obtenir un dictionnaire de Fourier sur-complet, il suffit de faire un échantillonnage plus fin des fréquences.

Cependant la plupart des algorithmes qui résolvent le problème d'optimisation (3.1) ne sont malheureusement pas adaptés au cas complexe. En général, lors de l'utilisation de la transformée de Fourier discrète, on sépare la partie réelle de la partie imaginaire ou bien le module de la phase. Généralement, les dictionnaires utilisés sont des dictionnaires de cosinus ou de sinus. Cependant de récents travaux [MBZJ08] ont présenté un algorithme de résolution de ce problème sous-déterminé dans le cas complexe, basé sur la «norme» $l_0$.

La périodicité de l'exponentielle rend de fait les atomes de la DFT particulièrement sensibles aux signaux périodiques. L'avantage est double : cela permet à la fois de modéliser avec parcimonie des signaux de type périodique mais également de propager le signal en répétant le motif élémentaire sur un plus grand support. Nous développerons ce second point dans les chapitres suivants.

*Atomes DCT*

La transformée en cosinus discrète usuelle est une transformée dont les atomes générateurs sont proches de la transformée de Fourier, puisqu'il s'agit d'un cosinus :

$$F_k = \sum_{n=0}^{N-1} f_n \cos\left[\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right]$$

Les atomes sont les fonctions :

$$a_{n,k} = \sqrt{\frac{2}{N}} \cos\left[\frac{k\pi}{N}\left(n+\frac{1}{2}\right)\right]$$

L'avantage de la DCT est d'offrir une décomposition dans l'espace des réels, contrairement à la DFT et de s'adapter ainsi parfaitement aux algorithmes développés dans le cadre des représentations parcimonieuses. Au même titre que la transformée de Fourier, la transformée en cosinus est propice à la reconstruction de signaux périodiques.

L'analyse purement fréquentielle a néanmoins quelques limitations. Elle permet certes de détecter les fréquences dominantes d'un signal, mais elle ne prend pas en compte ses caractéristiques temporelles. Les transformées de Fourier et en cosinus discrètes ne seront pas adaptées pour représenter des signaux non-stationnaires ou encore des discontinuités temporellement localisées.

#### 3.4.2.2 Dictionnaires temps-fréquence

Afin d'enrichir l'analyse qui peut être faite d'un signal, l'analyse temps-fréquence permet d'identifier à quels instants les fréquences du signal varient. Une des premières transformées utilisées pour modéliser des phénomènes non-stationnaires de durée très courte, est la transformée de Wigner-Ville [BF82]. Puis, l'usage d'atomes de Gabor fut démocratisé pour représenter des signaux caractérisés comme non-stationnaires.

Les atomes de Gabor sont des fonctions qui ont pour spécificité d'être localisées en temps et en fréquence. Ils sont définis comme étant une fréquence oscillant sur une courte période, localisation obtenue par fenêtrage :

$$a_{s,t,f}(\tau) = \frac{1}{s} w\left(\frac{\tau - t}{s}\right) e^{2i\pi f(\tau - t)}$$

où $\tau$ représente la localisation temporelle, $s$ est le paramètre qui définit la largeur de la fenêtre $w$, dans le domaine fréquentiel. Le choix de la fenêtre n'est pas contraint. On choisit en général des fenêtres aux propriétés fréquentielles satisfaisantes, comme une fenêtre de Hamming.

Le fenêtrage induit une découpe de l'onde en petits morceaux : ce sont les *ondelettes* de Gabor. Alors que l'analyse de Fourier est un outil efficace pour représenter des signaux stationnaires, les ondelettes de Gabor sont quant à elles utilisées pour représenter des signaux quasi-stationnaires. Comme le fenêtrage permet de segmenter le signal, on peut ainsi modéliser des zones où apparaissent des phénomènes non-stationnaires.

#### 3.4.2.3 Dictionnaires temps-échelle

Pour l'analyse de signaux images, il semble pertinent de s'intéresser aux informations contenues dans l'image à toutes les échelles observables. L'idée est d'allier l'analyse fréquentielle à une analyse par échelles. Les premières fonctions répondant à cette problématique sont les ondelettes de Morlet [JMG82]. Pour traiter des signaux non-stationnaires, il eut l'idée de raccourcir la fenêtre utilisée dans les ondelettes de Gabor. Il partit d'une fonction mère $\Psi$, il la décala dans le temps, et il changea d'échelle. Il obtint des fonctions $\Psi\left(\frac{t-b}{a}\right)$ où $a$ est le facteur d'échelle que l'on choisit petit pour représenter de manière efficace des phénomènes quasi-instantanés.

Plus tard, il fut introduit [Mur89] un paramètre supplémentaire pour prendre en compte la direction dans laquelle l'ondelette oscille. Afin de pouvoir contrôler l'orientation de cette oscillation, on introduit un paramètre $\theta$ qui permet de faire varier la direction des oscillations et donc la direction pour laquelle l'ondelette détecte les variations rapides :

$$\Psi\left(r_\theta . \frac{t-b}{a}\right) \quad \text{où} \quad r_\theta = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix}$$

Cet ensemble de fonctions forme des ondelettes directionnelles.

## 3.5 Conclusion

Analyser un signal signifie le décomposer en ses éléments constituants. Les représentations parcimonieuses est l'un des outils d'analyse permettant notamment d'extraire les composantes principales d'un signal. La particularité des représentations parcimonieuses est alors de choisir, parmi l'ensemble des représentations possibles, la plus parcimonieuse pour une erreur de reconstruction fixée. Nous avons choisi, dans ces travaux de thèse, d'utiliser les représentations parcimonieuses pour extrapoler des textures au sein des images. Le choix s'est porté sur les représentations parcimonieuses car nous souhaitions exploiter la souplesse dans le choix des outils de représentation, à savoir les dictionnaires. Les atomes qu'ils contiennent seront les briques primordiales nous permettant de «*recréer la matière*». Même si les atomes du dictionnaire sont, en pratique, en nombre limité, ils sont destinés à être choisis en fonction de leur corrélation avec l'image à représenter, ce qui permet alors d'effectuer, au coeur même du processus, une analyse des données connues. Cette analyse est somme toute limitée par le nombre et la variété des fonctions de base du dictionnaire.

# Chapitre 4

# Représentations parcimonieuses adaptées à la prédiction d'image

Nous venons de présenter le domaine des représentations parcimonieuses comme étant un outil majeur pour notamment acquérir une version réduite, compactée, d'un signal donné. Il s'agit là de la première utilisation évidente des représentations parcimonieuses. Nous nous sommes cependant orientés vers une autre utilisation de cet outil : la prédiction d'images. Nous avons vu au chapitre 1 que la prédiction spatiale d'un encodeur de type H.264 est notamment basée sur l'utilisation de recopies et de combinaisons linéaires de pixels limitrophes, le long de directions privilégiées. L'approche est peu complexe et permet d'obtenir des résultats satisfaisants. Cependant, cette technique trouve ses limites lorsqu'il s'agit d'étendre un signal bi-dimensionnel, plus complexe. Nous avons exposé au chapitre 2 les différentes techniques abordées dans la littérature pour synthétiser une texture. Les outils utilisés consistent principalement à extraire de la texture source, ses composantes principales, puis, les connaissant, à synthétiser une texture de dimensions voulues. Nous avons utilisé la même philosophie par le biais des représentations parcimonieuses dans le cadre de la prédiction d'image.

Ce chapitre présente la méthode de prédiction parcimonieuse que nous avons mise en place. Nous détaillons les problématiques soulevées et les solutions que nous proposons. La section 4.3 présente la mise en place de la technique dans un encodeur de type H.264 / AVC et en section 4.4, dans un formalisme de type SVC pour le cas de la prédiction spatiale inter-couches. Les résultats expérimentaux que nous avons obtenus sont rapportés en section 4.5.

## 4.1 Introduction

Cette introduction a pour vocation de tisser les liens entre les trois domaines recoupés par les travaux présentés dans cette thèse. Ces domaines sont : l'extrapolation de signal, la synthèse de texture et les représentations parcimonieuses.

Les notions d'extrapolation et de synthèse sont très proches et pourraient fusionner en un seul domaine. Cependant, l'idée ici est de présenter la philosophie générale de l'extrapolation de signal, et plus spécifiquement d'un signal mono-dimensionnel plutôt qu'un signal image bi-dimensionnel, afin d'en dégager les points clés.

L'extrapolation de signal consiste à étendre un signal, notons le $f \in \mathbb{R}^m$, au-delà des données d'observations connues, *i.e.* on recherche une expression de $f$ sur $\mathbb{R}^n$ avec $n >> m$. L'idée classique en matière d'extrapolation est de supposer que le signal à étendre peut se modéliser par une fonction analytique formelle que l'on choisit, mais dont les paramètres sont inconnus.

Notons la $g_{a,b,\theta}$ où $a, b, \theta$ sont, par exemple, les paramètres à estimer à partir des observations dont on dispose. Une des manières d'ajuster la valeur de ces paramètres est de minimiser l'énergie de l'erreur $r$, entre les données source et l'état courant $k$ du modèle, restreint à l'espace de dimension $m$ : $r = f - g^{(k)}_{a,b,\theta}|_{\mathbb{R}^m}$.

Grâce à la fonction analytique ainsi obtenue, on peut calculer les valeurs de nouveaux échantillons sur un ensemble de définition plus vaste, ici $\mathbb{R}^n$, que celui du signal original. Toute la difficulté réside dans la détermination des paramètres inconnus de la fonction analytique de référence.

Reprenons maintenant le formalisme des représentations parcimonieuses. On recherche la représentation parcimonieuse $x \in \mathbb{R}^n$ d'un signal $y \in \mathbb{R}^m$, en choisissant parmi un ensemble de fonctions de base, ou atomes, notées $a_j \in \mathbb{R}^m$, les colonnes du dictionnaire $A \in \mathbb{R}^{m \times n}$.

Si on fait le parallèle avec la méthodologie dédiée à l'extrapolation de signal, nous pouvons faire l'analogie suivante : un atome correspond à une fonction formelle, dont les paramètres sont fixés. Comme nous n'estimons plus les paramètres des fonctions de base, ce sont les algorithmes que nous avons présentés précédemment (comme le MP ou le GMF) qui évaluent le degré de correspondance du signal avec les fonctions de base.

Une fois trouvée, la représentation $x$ de dimension $n$ nous permettra d'obtenir une extension de $y$ de la dimension $m$ à la dimension $n$, sachant que la définition des $a_j \in \mathbb{R}^m$ est aussi connue sur $\mathbb{R}^n$.

## 4.2 Prédiction parcimonieuse

### 4.2.1 Principe

Soit le vecteur $y$ de dimension $m$ formé des observations locales. On distingue au sein de $y$, $n$ échantillons non nuls et donc $m - n$ valeurs nulles, représentant les données inconnues à prédire. Le principe va être de se baser sur la connaissance des $n$ échantillons non nuls contenus dans $y$ pour extrapoler les $m - n$ données inconnues, par le biais de fonctions de base contenues dans le dictionnaire $A$. On suppose ainsi qu'une bonne approximation des $n$ échantillons connus conduira à une bonne extrapolation des $m - n$ données inconnues.

On forme donc la matrice $A$ de dimension $m \times m$ où chaque colonne correspond à une fonction de base de dimension $m$. Afin de créer une base redondante, notée $A_c$, on masque les $m - n$ lignes de $A$ correspondant aux échantillons nuls contenus dans $y$. Le dictionnaire $A_c$ est alors de dimension $n \times m$. Le degré de redondance de ce dictionnaire est ainsi directement lié au nombre $m - n$ d'échantillons nuls. On procède de la même manière en retirant les échantillons nuls de $y$ et ainsi générer le vecteur $y_c$ contenant uniquement les $n$ observations non nulles.

Le but est ensuite de rechercher une approximation du signal $y_c$, en terme de fonctions de base contenues dans le dictionnaire $A_c$. Une fois la solution la plus parcimonieuse $x$ de dimension $m$ obtenue par le biais d'algorithmes comme le MP, l'OMP ou le GMF, on génère le vecteur $y_e$ de

dimension $m$, à l'aide du dictionnaire initial, $A$.

$$y_e = Ax$$

Le signal $y_e$ est ainsi composé de $m$ échantillons non nuls, représentant une approximation des $n$ observations non nulles et une extrapolation des $m - n$ données de valeur nulle, présente initialement dans le vecteur $y$.

*– Mise en place dans le cadre de la prédiction spatiale –*

Nous proposons d'étendre un signal image par le biais de fonctions de base bi-dimensionnelles connues, stockées dans le dictionnaire $A$. La figure 4.1 présente quelques atomes 2D issus d'une DCT. Ces atomes sont ensuite vectorisés en les parcourant ligne par ligne pour former les colonnes du dictionnaire.

FIG. 4.1 – Exemple d'atomes bi-dimensionnels extraits d'une DCT

Nos données d'observation sont les pixels connus issus du voisinage proche de la zone spatiale à prédire. Nous travaillons sur des blocs, donc prenons l'exemple d'une zone d'observation constituée de trois blocs de pixels connus, le quatrième étant celui que nous voulons prédire (cf figure 4.2). Tous les pixels de ce dernier bloc sont nuls.

Par exemple, sur la figure 4.2 le bloc $P$ constitué de $n \times n$ pixels à prédire à l'aide du voisinage $V$ de taille $3n^2$. On associe ensuite à la zone $L = V \cup P$ contenant 4 blocs et étant ainsi de dimension $2n \times 2n$, les fonctions de base choisies pour former le dictionnaire $A$. Les colonnes de la matrice $A$ correspondent ainsi à $4n^2$ atomes de dimension $4n^2$.

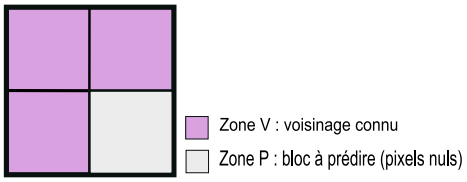



FIG. 4.2 – Voisinage de trois blocs

Notons $y$ le vecteur de dimension $4n^2$ contenant tous les pixels de la zone $L$ parcourue ligne par ligne. Soit $x$ le vecteur contenant les coefficients de la représentation de $y$ en termes de fonctions de base :

$$y = Ax$$

Seul les pixels non nuls de la zone $L$ sont pertinents comme support de prédiction. On modifie ainsi la matrice $A$ en masquant les lignes correspondant aux pixels nuls ne faisant pas partie de la zone $V$. On obtient alors une matrice compactée, notée $A_c$, dont la taille est $3n^2 \times 4n^2$. Les pixels correspondants contenus dans le vecteur $y$ sont également supprimés pour former le vecteur compacté $y_c$ de dimension $3n^2$. Les algoritmes de type *Matching Pursuit*, *Orthogonal Matching Pursuit* ou *Global Matched Filter* sont ensuite appliqués à $A_c$ et $y_c$.

Remarque: *L*e fait de mettre bout à bout des pixels non corrélés entre eux, génère des discontinuités qu'il va falloir être à même de représenter. En compactant de la même manière, les atomes du dictionnaire, on introduit des discontinuités aux mêmes endroits rendant, les atomes mieux corrélés aux observations bruitées par les discontinuités et plus susceptibles de conduire à une représentation qui soit parcimonieuse.

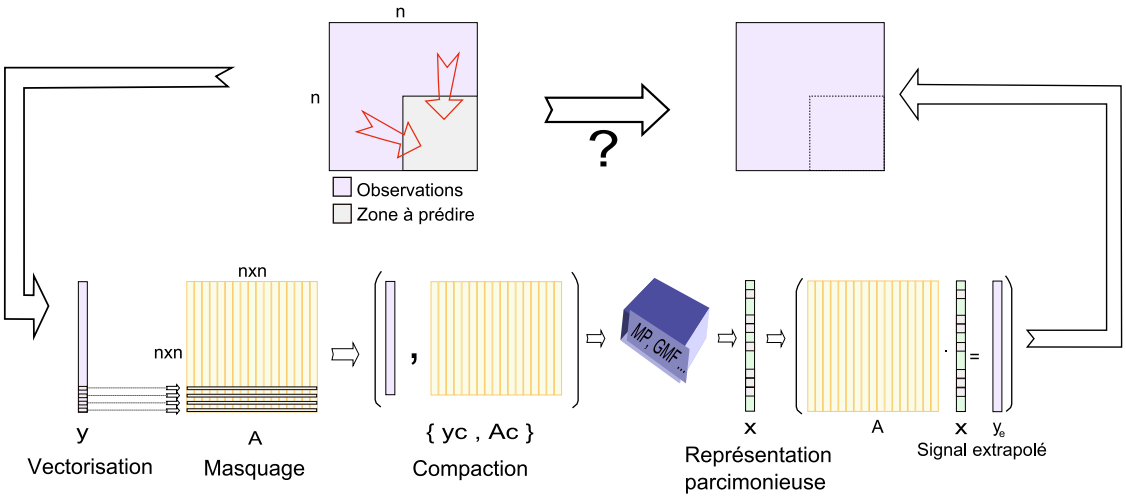



FIG. 4.3 – Principe de la prédiction parcimonieuse

Le schéma 4.3 résume le principe général de la prédiction parcimonieuse. Une fois obtenue la représentation parcimonieuse $x$, on utilise le dictionnaire non compacté, $A$, pour former le vecteur $y_e$ contenant les données extrapolées.

Remarque: *L*a problématique ne s'arrête pas à l'obtention de la représentation parcimonieuse. On souhaite avant tout extrapoler les données $y_c$ sur un ensemble plus vaste. Nous verrons d'ailleurs par la suite que la représentation la plus parcimonieuse n'est pas nécessairement la meilleure prédiction

### 4.2.2 Motivations théoriques

La prédiction parcimonieuse repose sur plusieurs aspects théoriques bien précis, centrés sur deux points clés : le voisinage d'approximation, support de la prédiction et la composition du dictionnaire, à savoir, la nature des fonctions de base. On part de l'hypothèse classique en traitement d'images, que le signal est considéré comme stationnaire sur un environnement local, de faibles dimensions. Ainsi, cela a du sens de s'appuyer sur un voisinage où les pixels ont de fortes chances d'être toujours corrélés entre eux pour réussir à générer une texture ayant les mêmes caractéristiques structurelles. Cependant la complexité des images naturelles est telle, que le signal est rarement stationnaire. Nous verrons par la suite quelques outils qui permettent néanmoins de contourner ce problème. Un autre aspect important réside dans la nature même des fonctions de base utilisées. Elles doivent en effet remplir deux fonctions primordiales dans le cadre de la prédiction : représenter *et* étendre le signal.

Il faut tout d'abord qu'elles puissent représenter le plus grand nombre de motifs contenus dans une image. Ces fonctions doivent également avoir un support, au sens mathématique, suffisamment étendu dans l'espace de définition. En termes plus concrets, si la fonction de base a toutes ses valeurs non nulles (l'information utile) sur un petit sous-ensemble de son domaine de définition, elle ne pourra pas étendre le signal sur l'ensemble du domaine de définition (les

valeurs nulles de la fonction de base étant évidemment inutiles pour modéliser le signal recherché).

Nous évoquerons plus en détails dans le chapitre 6, les difficultés que soulève le choix des fonctions de base et quelles pistes d'analyse nous avons explorées.

### 4.2.3 Choix et optimisation du voisinage d'approximation

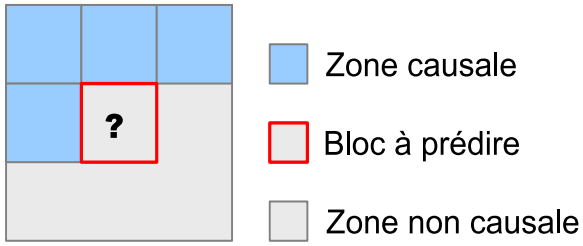



FIG. 4.4 – Exemple de voisinage d'approximation

Le voisinage que nous avons décidé d'utiliser correspond à la réunion des blocs limitrophes au bloc courant (un exemple est présenté en figure 4.4). Ce voisinage constitue un assez bon compromis entre complexité et qualité de reconstruction. La quantité d'information récupérée est suffisante tout en conservant une forte corrélation avec les pixels les plus proches des pixels inconnus. La propagation d'un motif bi-dimensionnel pourrait s'avérer délicate si nous ne disposions pas d'un support connu suffisamment étendu.

#### 4.2.3.1 Fonction de pondération

Fixer le voisinage engendre quelques problèmes. Il n'est pas toujours pertinent d'utiliser l'ensemble des observations du voisinage, tout particulièrement lorsque l'hypothèse de stationnarité du signal n'est pas respectée. Pour pallier ce problème, nous avons testé une solution qui consiste à appliquer une fonction de pondération aux pixels du voisinage, en donnant plus de poids aux pixels proches de la zone à prédire. On a utilisé la fonction suivante :

$$w(n,m) = \rho^{\sqrt{(n-N/2)^2+(m-N/2)^2}}$$

avec $N$ la dimension de la zone étudiée et $\rho \in ]0,1[$ un paramètre à fixer.

Comme l'illustre la figure 4.5, on constate que les pixels les plus éloignés deviennent négligeables par rapport à ceux entourant la zone à prédire. Ainsi, au sein des algorithmes, on a introduit la matrice $W$, obtenue en diagonalisant la fonction $w$, au niveau de l'évaluation de l'erreur de reconstruction :

$$W.(y - Ax)$$

#### 4.2.3.2 Segmentation du voisinage

Lorsque la fonction de pondération n'est pas suffisante, on peut envisager une autre méthode pour améliorer l'extrapolation du signal en cas de non-stationnarité. L'idée consiste à extraire du voisinage d'approximation, les zones où le signal est localement stationnaire. Cela soulève le délicat problème de la sélection des pixels pertinents et plus exactement de l'évaluation de la pertinence de ce choix. Le plus sûr moyen de sélectionner l'information utile serait d'analyser le contenu local, notamment par le biais d'algorithmes connus en analyse / synthèse de texture.

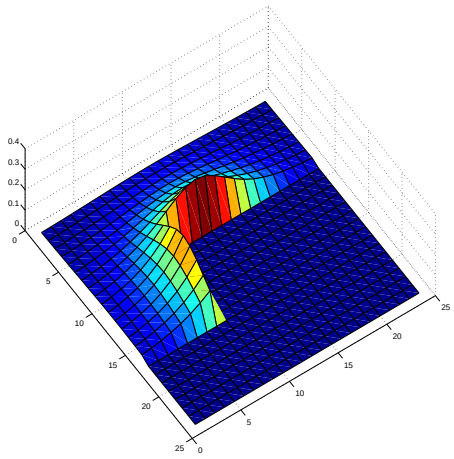



FIG. 4.5 – Fonction de pondération

L'idée générale étant de ne conserver que les pixels formant une même unité, les autres pixels s'apparentant alors à du bruit parasite. Ce type de technique peut s'appliquer pour distinguer aussi bien des zones homogènes que texturées, ainsi que les contours d'une image.

Dans le même esprit mais de complexité inférieure, nous avons choisi de segmenter le voisinage d'approximation en plusieurs zones. Les diverses zones ainsi créées deviennent les nouveaux supports de prédiction, moins riches en information mais potentiellement moins bruités.

Chaque prédiction formée peut ensuite être évaluée en terme de qualité de reconstruction : la meilleure sera retenue comme prédiction définitive. Ici encore se pose le problème de l'évaluation de la qualité de reconstruction, problème majeur notamment dans le domaine de la compression d'images. Il apparaît donc comme primordial d'avoir des critères d'évaluation ajustés aux intérêts qui nous préoccupent : la qualité de la prédiction obtenue, ainsi que l'impact de cette prédiction dans la boucle de codage, en terme de débit / distorsion.

### 4.2.4 Critère d'arrêt

Une problématique intrinsèque à notre méthode est liée au fait que nous pouvons garantir une bonne représentation du voisinage local mais cela ne présuppose pas une bonne prédiction du bloc courant. En effet, la recherche du vecteur parcimonieux est réalisée sur la zone causale et les algorithmes (MP et OMP) s'arrêtent lorsque l'erreur entre l'approximation de cette zone causale et les données d'observations est inférieure à un certain seuil, fixé à l'avance (cf figure 4.6, à gauche). Afin d'améliorer la prédiction du bloc courant, nous avons ajouté une condition portant sur la qualité de la prédiction et non uniquement sur la qualité de l'approximation obtenue pour le voisinage local, comme l'illustre la partie droite de la figure 4.6.

Nous nous sommes intéressés à différents critères pour évaluer la qualité de la représentation à chaque itération des algorithmes, mais également son coût de codage. Nous distinguons un critère basé sur l'erreur quadratique moyenne (EQM) et un autre basé sur une optimisation débit / distorsion (D/R).

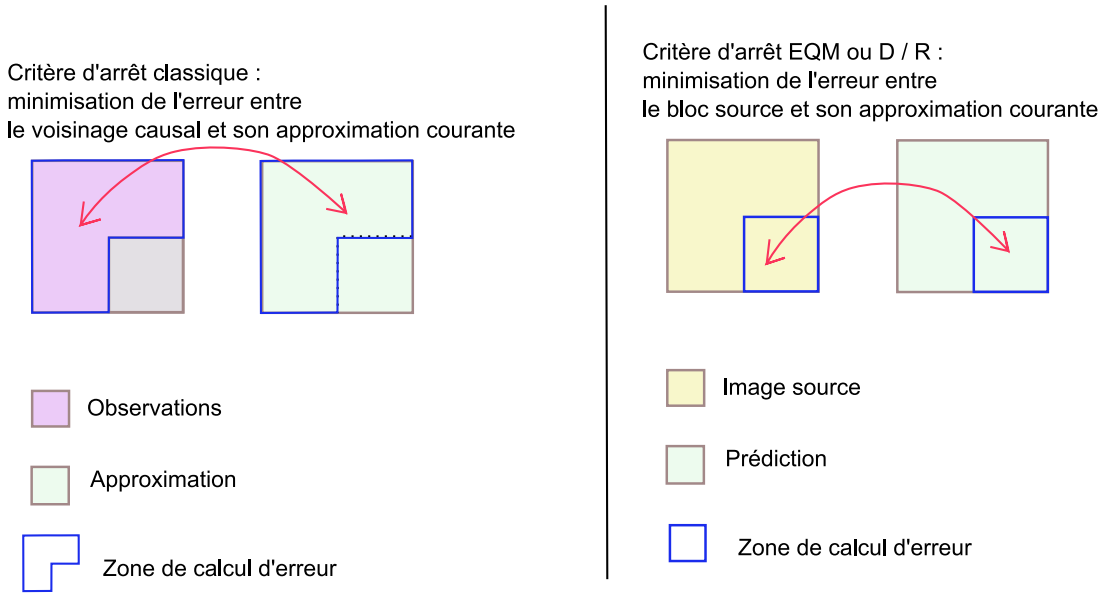



FIG. 4.6 – Illustration des différents critères d'arrêt

#### 4.2.4.1 Erreur quadratique moyenne

Le premier critère testé est le PSNR. On a mesuré les erreurs faites, d'une part, entre notre estimation de la zone causale et les données sources, et d'autre part, entre notre prédiction du bloc courant et le bloc source.

La figure 4.7 présente schématiquement, pour ces deux zones, l'évolution du PSNR en fonction du nombre d'itérations des algorithmes. On en déduit que l'erreur faite sur la zone causale décroît exponentiellement, au fur et à mesure que le nombre de coefficients non nuls augmente dans la représentation. Alors que l'erreur entre la prédiction et les pixels sources commence par décroître rapidement, puis ensuite fluctue aléatoirement.

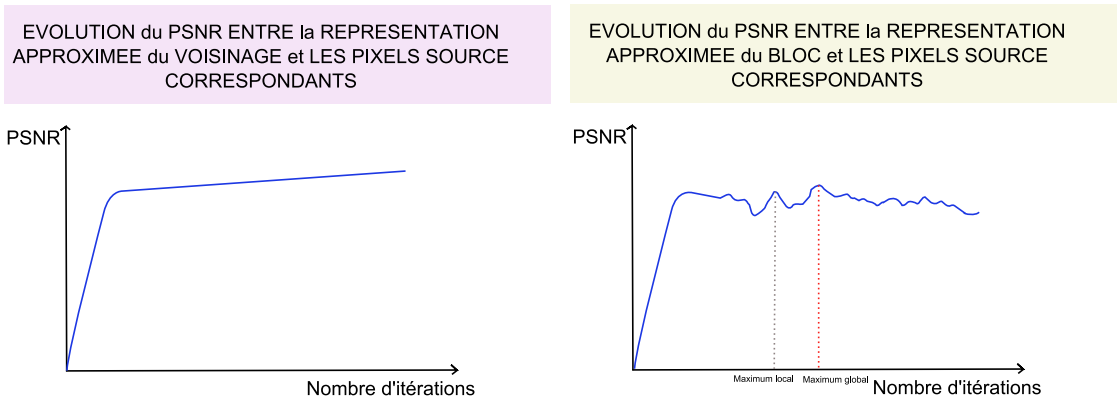



FIG. 4.7 – Evolution du PSNR en fonction des zones représentées

On veut donc appliquer aux algorithmes un critère d'arrêt qui tend à minimiser l'erreur de reconstruction dans la zone à prédire.

Les algorithmes sont implémentés de telle manière qu'ils génèrent une séquence de représentations $x_k$ de complexité croissante et pour chaque $x_k$, on calcule l'énergie de l'erreur de reconstruction :

$$\left\| y_c - A_c x_k \right\|_2 \tag{4.1}$$

où les notations précédemment introduites sont à nouveau utilisées : $y_c$ le vecteur contenant uniquement le voisinage causal et $A_c$ le dictionnaire dont les atomes ont été compactés.

On devrait s'arrêter dès que cette erreur de prédiction, qui généralement commence à décroître, augmente. Mais comme il n'y a aucune raison pour qu'une représentation plus complexe conduise obligatoirement à une erreur de reconstruction qui augmente systématiquement, on procède différemment en se basant sur une méthode en deux temps :

– Tout d'abord, l'algorithme itère jusqu'à ce que le seuil, préalablement fixé, sur l'erreur de reconstruction (4.1) soit atteint et l'ensemble des séquences $x_k$ est stocké en mémoire. La valeur du seuil est fixée de manière à ce que la représentation finale ait un assez grand nombre de coefficients non nuls, disons $K$ éléments.

– Dans un second temps, on sélectionne la représentation optimale comme étant celle qui conduit à la plus petite erreur de reconstruction dans la zone $P$ :

$$k_{opt} = \min_{k \in [1,K]} \left\| y_P - A_P x_k \right\|$$

#### 4.2.4.2 Fonction de coût lagrangienne

Ce critère vise à générer une optimisation débit / distorsion. Nous l'avons présenté plus en détail, au chapitre 1, à la section 1.7.2. Nous rappelons simplement ici que ce critère est basé sur une fonction de coût lagrangienne $J_\lambda$ que l'on cherche à minimiser :

$$J_\lambda = D + \lambda . R$$

où $D$ est une mesure de distorsion, $R$ représente le débit en bits et $\lambda$ un paramètre d'ajustement qui dépend des contraintes de quantification. Idéalement pour nos travaux, le débit $R$ doit tenir compte, à la fois du nombre de bits pour coder le résidu, noté $R_{res}$, et du nombre de bits nécessaire pour coder le nombre d'itérations $R_{it}$ (cf section précédente). Il faudrait donc considérer la fonction de coût suivante :

$$J'_\lambda = D + \lambda . (R_{res} + R_{it})$$

Nous n'avons cependant pas inclus le surcoût $R_{it}$ au sein de la décision. Les estimations qui ont été faites de cette pénalisante reposent sur un calcul *a posteriori*. En ce qui concerne la valeur du débit $R_{res}$, elle peut peut correspondre : soit au débit vrai, soit à un débit estimé par un modèle afin de réduire le temps de calcul. Nous avons choisi de prendre en compte le débit réel codé avec CAVLC.

### 4.2.5 Choix du dictionnaire

Nous avons précédemment évoqué l'importance du choix des fonctions de base. Rappelons ici que nous ne cherchons pas uniquement la représentation parcimonieuse d'un signal. Nous voulons en plus, l'étendre sur un support plus grand. Ainsi les fonctions de base se doivent d'avoir elles-mêmes un support suffisamment étendu.

Les fonctions de base périodiques sont un cas particulier de fonctions non locales. Elles sont particulièrement efficaces pour étendre des motifs réguliers et répétitifs. Les textures présentes

dans une image naturelle sont très variées mais les fonctions de base fréquentielles de type DCT ou DFT répondent relativement bien à la problématique qui nous occupe. Nous nous sommes tout naturellement intéressés à ces dictionnaires, comme première approche de nos travaux, pour former la prédiction parcimonieuse.

## 4.3 Application à la prédiction intra dans un codeur H.264 / AVC

### 4.3.1 Pourquoi vouloir améliorer la prédiction intra de la norme ?

La prédiction intra image formée au sein d'un encodeur de type H.264 / AVC est relativement simple à mettre en oeuvre. Nous avons préalablement détaillé ce type de prédiction dans la section 1.6.3.1. La figure 4.8 illustre sur un exemple basique les neuf prédictions obtenues à partir des modes intra de la norme.

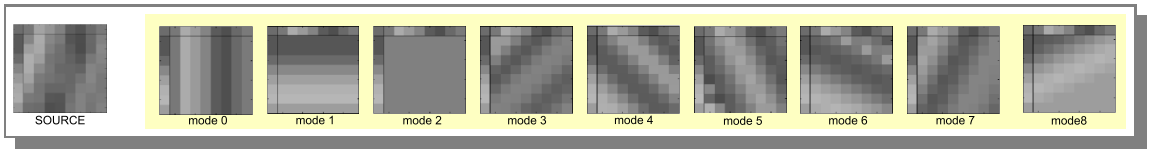



FIG. 4.8 – Exemple de prédiction intra H.264 / AVC

Si on compare attentivement l'image source à toutes les prédictions, on constate qu'aucune ne retranscrit le motif dans toute sa complexité.

La technique est limitée de par sa définition, à ne générer des signaux que dans une seule direction (excepté le mode DC). C'est l'observation de ce phénomène qui a principalement motivé notre recherche de méthodes plus élaborées pour reproduire des signaux images aux motifs bi-dimensionnels.

### 4.3.2 Support de prédiction : le passé causal

Le codeur transmet les données au décodeur par macroblocs successifs. Si bien qu'à un instant quelconque du processus, le décodeur n'a reconstruit que partiellement l'image, comme illustré en figure 4.9. La zone grisée présentée dans cette figure constitue un état possible de l'image en cours de reconstruction.

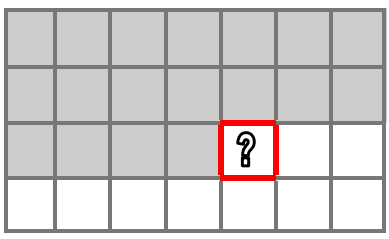


FIG. 4.9 – Découpage en macroblocs de taille $16 \times 16$ pixels

La notion de causalité intervient au niveau du codeur. Lorsqu'un traitement effectué au sein du codeur, ne fait appel qu'aux pixels précédemment reconstruits, le décodeur pourra réaliser la même opération sans qu'on n'ait besoin de transmettre des informations supplémentaires. Le

traitement est dit *causal*. On appelle ainsi le passé causal, l'ensemble des pixels formé par les blocs précédemment reconstruits.

Nous avons choisi de travailler dans un encodeur H.264 / AVC avec pour paramètre de parcours des blocs, le mode *raster scan*. Il s'agit du parcours classique horizontal, ligne par ligne. Notons qu'il existe également le parcours vertical et d'autres parcours plus spécifiques, tous regroupés sous le terme FMO pour *Flexible Macroblock Ordering*. Le processus d'encodage de l'image s'effectue par macroblocs de taille $16 \times 16$ pixels, pour lesquels il existe un découpage plus fin en blocs de tailles $8 \times 8$ et $4 \times 4$.

Ainsi au fur et à mesure du processus, le voisinage causal disponible varie en fonction des différentes tailles de blocs mais aussi de l'ordre de codage des blocs au sein d'un macrobloc. Les flèches en zigzag de la figure 4.10 indiquent l'ordre dans lequel on parcourt les différentes sous-partitions d'un macrobloc. On peut donc s'appuyer sur un voisinage de trois, quatre ou cinq blocs adjacents pour former la prédiction, comme l'illustre la figure 4.10.

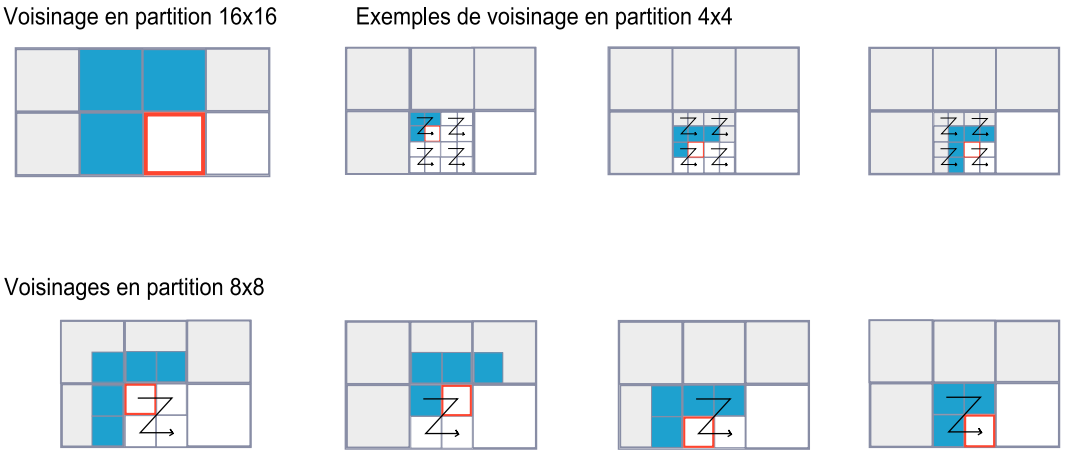



FIG. 4.10 – Exemples de voisinages en fonction des tailles de blocs et de l'ordre de codage

### 4.3.3 Codage du nombre d'itérations

Les critères d'arrêt que nous avons introduits pour garantir une meilleure prédiction dépendent des données source. Or l'image source n'étant pas accessible au décodeur, le nombre optimal d'itérations effectuées, $k_{opt}$, doit impérativement lui être transmis afin de pouvoir former la même prédiction.

Remarque: *Il s'agit bien du nombre d'itérations qu'il faut envoyer et non simplement le nombre d'atomes utilisés. Il arrive qu'un atome déjà sélectionné soit à nouveau choisi (algorithme du MP) et qu'il apporte ainsi une nouvelle contribution, inférieure à la précédente. Le coefficient correspondant dans la représentation $x$ est alors ajusté par un terme correctif, à la hauteur de cette nouvelle contribution.*

## 4.4 Application à la prédiction inter-couches dans SVC

### 4.4.1 Raffinement de la prédiction SVC

Nous proposons un mode de prédiction spatiale basé sur les représentations parcimonieuses, qui vise à exploiter simultanément l'information locale d'une couche d'amélioration et l'information issue d'une couche de base.

L'information support extraite de la couche d'amélioration correspond au voisinage causal tel que défini précédemment pour la norme H.264 / AVC, dans la section 4.3.2. La nouveauté consiste à compléter la zone inconnue, *i.e* la zone **non causale**, par les données colocalisées issues de la couche de base. Les deux images étant de résolutions différentes, les échantillons de la couche de base sont interpolés afin de correspondre à la résolution courante.

Inclure ces données pour la prédiction parcimonieuse a deux avantages.

- Le premier est bien sûr, l'apport d'information non causale qui jusqu'alors était inconnue dans le cas d'une prédiction intra image.
- Le second avantage est que la technique utilisée va naturellement débruiter le signal interpolé. Il peut s'opérer un réajustement des niveaux de gris des pixels bruités grâce à la connaissance des pixels issus de la zone causale. On espère ainsi améliorer la prédiction obtenue par le biais de la prédiction spatiale inter-couches du standard SVC.

### 4.4.2 Mise en place de la prédiction

Nous avons pris le parti de tirer profit de toute l'information non causale au niveau de la couche de résolution supérieure. On aurait pu imaginer interpoler uniquement la zone correspondant au bloc à prédire et laisser les autres pixels à zéro. Il s'avère néanmoins plus performant de compléter toute la zone non causale avec l'information issue de la couche de base interpolée.

Ceci a pour conséquence que le vecteur d'observation $y$ ne contient pas de valeurs nulles. Rappelons que nous ne cherchons plus ici à extrapoler le signal, nous souhaitons simplement le raffiner. En l'état, nous avons donc les données d'observation $y$, vecteur de dimension $n$ pour lequel nous cherchons une représentation $x$ à l'aide d'un dictionnaire appartenant à $\mathbb{R}^{n\times n}$. Ainsi défini, cela correspond à décomposer le signal sur une base de fonctions, le dictionnaire n'étant plus redondant. Afin de créer de la redondance dans le dictionnaire, il est nécessaire d'ajouter, par exemple, $p$ ($> 0$) fonctions de base. Ainsi le dictionnaire $A$ appartenant à $\mathbb{R}^{n\times(n+p)}$ est une base redondante.

## 4.5 Résultats expérimentaux dans le cadre H.264 / AVC

### 4.5.1 Substitution à un mode de la norme H.264 / AVC

Pour évaluer notre prédiction, nous devons la mettre en concurrence avec les modes de prédiction intra-image existant dans l'encodeur. L'ajout d'un mode supplémentaire nécessite cependant de modifier la syntaxe du train binaire. Afin de s'affranchir des modifications longues et fastidieuses que cela implique, nous avons choisi une approche plus immédiate qui consiste à substituer notre prédiction à un des modes existants.

Pour une image donnée, nous évaluons tout d'abord le pourcentage de sélection de chaque mode intra-image, dans le cas d'un encodage H.264 / AVC classique, *i.e.* sans notre prédiction. Ensuite, nous remplaçons la prédiction H.264 / AVC la moins utilisée par la prédiction parcimonieuse. Le processus de sélection des modes reste par ailleurs le même.

### 4.5.2 Analyse des performances

L'analyse visuelle est un des premiers outils permettant de jauger de la qualité de notre prédiction. Cela nous permet de constater l'efficacité de notre technique, sur des zones texturées complexes, qui sont difficilement prédites via les modes intra-image de la norme.

Pour commencer, nous avons comparé visuellement l'image de prédiction générée par les neuf modes de la norme et celle obtenue par notre prédiction parcimonieuse.

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... MP
* Dictionnaire ..... DCT
* *QP* ..... 21
* Partition ..... $8 \times 8$
* Sélection des modes intra ..... SAD
* Seuil ..... 1
* Critère d'arrêt ..... EQM

Dans ce contexte, les pourcentages de sélection des modes intra $8 \times 8$, dans le cas d'une prédiction classique de la norme, sont répertoriés dans le tableau 4.1. Le mode horizontal bas est celui qui est en moyenne le moins utilisé. Nous avons donc choisi de substituer notre prédiction au mode 6. Dans le cadre de ce test, notre prédiction est choisie à hauteur de 30%.

| Modes intra $8 \times 8$ | Pourcentages (%) | Modes intra $8 \times 8$ | Pourcentages (%) |
|---|---|---|---|
| 0 (vertical) | 20.3 | 0 (vertical) | 15.4 |
| 1 (horizontal) | 11.1 | 1 (horizontal) | 9.72 |
| 2 (DC) | 14.7 | 2 (DC) | 8.47 |
| 3 (diagonal gauche-bas) | 6.93 | 3 (diagonal gauche-bas) | 5.39 |
| 4 (diagonal droite-bas) | 7.23 | 4 (diagonal droite-bas) | 6.47 |
| 5 (vertical droite) | 13.4 | 5 (vertical droite) | 8.49 |
| 6 (horizontal bas) | 4.44 | 6 (prédiction parcimonieuse) | 30.0 |
| 7 (vertical gauche) | 15.4 | 7 (vertical gauche) | 10.4 |
| 8 (horizontal haut) | 6.57 | 8 (horizontal haut) | 5.64 |

TAB. 4.1 – Pourcentages de sélection des modes intra $8 \times 8$ sur l'image *Barbara*, sans et avec notre prédiction

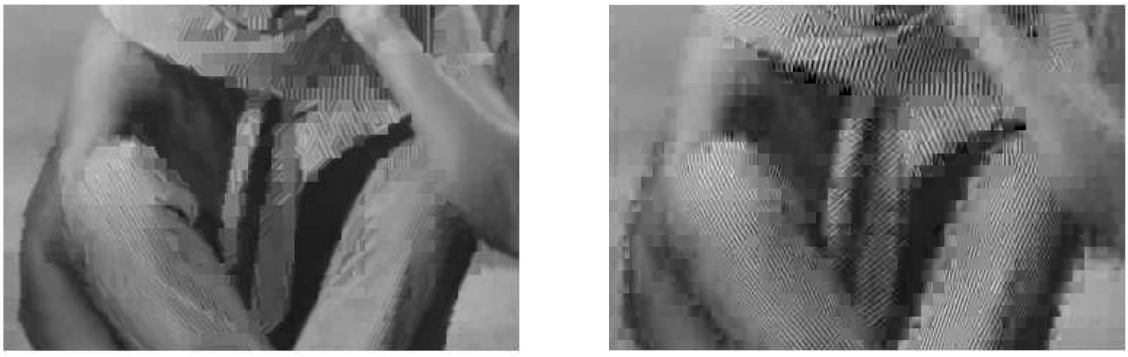


FIG. 4.11 – Détail de prédiction intra $8 \times 8$ de la norme à gauche ; prédiction parcimonieuse appliquée à des blocs $8 \times 8$ à droite

La figure 4.11 présente un détail extrait de la prédiction de l'image *Barbara* : à gauche, celle

de la norme, à droite, celle obtenue uniquement avec notre méthode. Les structures linéiques du pantalon et du foulard sont correctement restituées avec notre méthode. En revanche, la prédiction issue de la norme ne parvient pas à reconstituer les structures obliques de la texture des vêtements, n'ayant pas le mode directionnel correspondant à ces orientations.

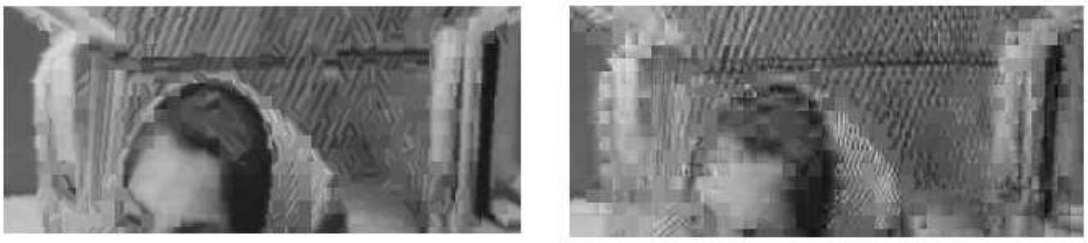


FIG. 4.12 – Détail de la prédiction intra $8 \times 8$ de la norme à gauche ; prédiction parcimonieuse appliquée à des blocs $8 \times 8$ à droite

D'autre part, la texture ajourée du fauteuil est un motif bi-dimensionnel assez complexe à restituer à partir des modes mono-directionnels de la norme. La figure 4.12 présente la restitution de ce motif via la prédiction de la norme et grâce à notre méthode. Cet exemple permet de jauger les limites des modes intra d'H.264 / AVC qui n'ont pas la capacité de représenter des textures structurellement trop complexes. En revanche, notre prédiction basée sur les représentations parcimonieuses permet une restitution plus fidèle du motif original, dans la mesure où la texture est étendue de manière bi-directionnelle.

### 4.5.3 Analyse en terme débit / distorsion

Nous avons ainsi utilisé des outils basés sur le critère Bjontegaard [TWL03] s'appuyant sur les courbes débit / distorsion , couramment utilisées en compression d'images pour évaluer les performances d'un encodeur. Ces courbes représentent l'évolution de la qualité, évaluée par le PSNR en décibels (dB), en fonction du débit, évalué en bits/s. Le critère Bjontegaard fournit une évaluation du gain en qualité à débit constant, ainsi que le gain en débit, pour une qualité donnée.

Remarque: *Pour générer ces gains, le critère se base sur la connaissance de quatre valeurs de couples* {*qualité, débit*}*, choisis sur une plage de valeur relativement grande pour évaluer les performances sur la plus grande plage de débit possible.*

La variable permettant de choisir le débit désiré est directement liée au pas de quantification, qui détermine également la dégradation induite au signal. Il s'agit du paramètre de quantification, appelé *QP* (*Quantization Parameter*) dont la plage de valeur s'étend de 0, pour une image finement quantifiée à 51, pour une image très fortement quantifiée. Dans notre étude, nous avons choisi de générer les résultats de mesures, de type Bjontegaard, à partir des valeurs de *QP* suivantes : 21, 26, 30 et 35.

Le tableau 4.2 résume les résultats obtenus sur l'image *Barbara*, pour les trois types de prédictions intra-image, en comparant la prédiction intra-image de la norme à celle obtenue en substituant notre prédiction à un des modes d'H.264 / AVC.

| | Avec critère EQM | | Sans critère EQM | |
|---|---|---|---|---|
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| Sans la fenêtre de pondération | | | | |
| $4 \times 4$ | + 0.72 | - 8.55 | + 0.36 | - 4.24 |
| $8 \times 8$ | + 0.43 | - 5.61 | + 0.23 | - 3.04 |
| $4 \times 4$ et $8 \times 8$ | + 0.59 | - 7.56 | + 0.30 | - 3.89 |
| $4 \times 4$, $8 \times 8$ et $16 \times 16$ | + 0.59 | - 7.60 | + 0.27 | - 3.59 |
| Avec la fenêtre de pondération | | | | |
| $4 \times 4$ | + 0.78 | - 9.22 | + 0.33 | - 3.97 |
| $8 \times 8$ | + 0.56 | - 7.27 | + 0.27 | - 3.49 |
| $4 \times 4$ et $8 \times 8$ | + 0.72 | - 9.21 | + 0.33 | - 4.21 |
| $4 \times 4$, $8 \times 8$ et $16 \times 16$ | + 0.72 | - 9.33 | + 0.32 | - 4.18 |

TAB. 4.2 – Résultats Bjontegaard de la prédiction parcimonieuse, pour les trois tailles de bloc, avec et sans fonction de pondération. A gauche : avec le critère d'arrêt supplémentaire, à droite, sans.

Pour les prédictions intra $4 \times 4$ et $8 \times 8$ , le mode statistiquement le moins utilisé ici est le mode 6. Pour la prédiction $16 \times 16$, il s'agit du mode 1. Le tableau présente les gains obtenus via le critère de Bjontegaard pour une prédiction parcimonieuse sans et avec ajout de la fenêtre de pondération. Nous constatons d'après les résultats retranscrits ici, que la méthode présente des résultats encourageants. Sans l'ajout du critère d'arrêt supplémentaire basé sur l'EQM, nous observation une diminution de débit, associée à une augmentation du PSNR, par rapport à la prédiction d'H.264 / AVC. De plus, l'apport du critère d'arrêt supplémentaire permet de réduire de manière significative le débit de l'ordre de 7% voire 8% lorsque l'on ajoute la fonction de pondération.

Cependant, ces résultats n'incluent pas le coût de codage induit par le critère d'arrêt supplémentaire que nous avons introduit. Comme nous sélectionnons la meilleure représentation $x_k$ en fonction du PSNR évalué entre le bloc source et le bloc que nous prédisons, il est impératif d'informer le décodeur du nombre d'itérations finalement retenu. Pour évaluer ce coût, nous avons calculé l'entropie à partir de l'histogramme formé des nombres d'itérations finalement sélectionnées pour chacun des blocs prédits par notre méthode de prédiction.

### 4.5.4 Évaluation du coût de codage du nombre d'itérations

L'histogramme permet de connaître la fréquence $f_i$ d'utilisation des symboles à coder, ici le nombre d'itérations $k$, pouvant prendre jusqu'à $K$ valeurs possibles. Le coût de transmission d'une itération est alors obtenu en calculant l'entropie $H_{it}$ suivante :

$$H_{it} = -\sum_{i=0}^{K-1} f_i \log_2(f_i)$$

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... Matching Pursuit
* Dictionnaire ..... DCT

* *QP* ..... 21 – 26 – 30 – 35
* Partition ..... $8 \times 8$
* Sélection des modes intra ..... SAD
* Seuil ..... 1
* Critère d'arrêt ..... EQM

La figure 4.13 présente l'histogramme des valeurs prises par le nombre d'itérations retenu pour former la meilleure prédiction, dans le cas hypothétique où tous les blocs sont prédits par notre méthode. Soit $N = 4096$ blocs $8 \times 8$ pour cette image de taille $512 \times 512$. Nous obtenons des valeurs comprises entre 1 et 79, pour cette simulation faite à un *QP* de 21.

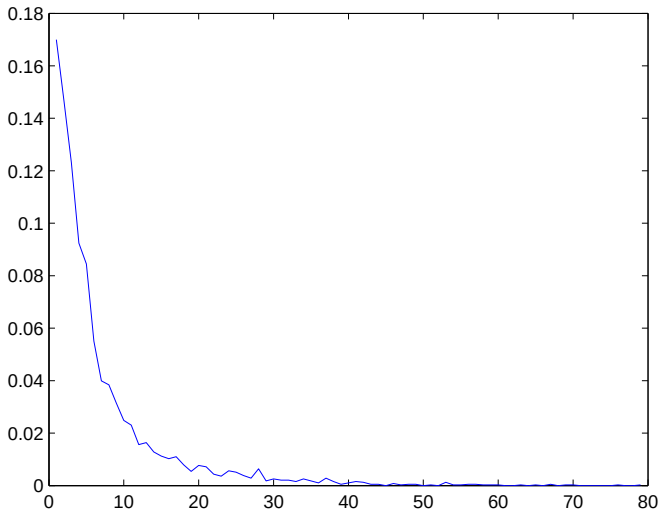



FIG. 4.13 – Histogramme des valeurs du nombre d'itérations après choix de la meilleure représentation, selon un critère EQM

Cependant, pour connaître le nombre réel de bits à transmettre, il faut tenir compte du pourcentage $p$ de sélection de notre prédiction, lors du choix du meilleur mode intra. On forme ainsi l'histogramme, présenté en figure 4.14, qui tient uniquement compte des valeurs des itérations dans les cas où notre prédiction a été sélectionnée comme la meilleure pour le bloc courant.

Remarque: *L'histogramme 4.14 ne présente pas la même allure que celui en figure 4.13. On constate que l'évolution des valeurs prises par le nombre d'itérations n'est pas monotone et tout particulièrement lorsque le nombre d'itérations est faible. Ceci s'explique par le fait que la prédiction parcimonieuse est moins efficace que celle de la norme lorsqu'un seul coefficient est retenu, i.e. lorsqu'on calcule une moyenne. Comme nous moyennons des pixels d'un vaste voisinage, cette valeur est dans la plupart des cas moins bonne que le mode DC de la norme, qui calcule une moyenne sur une faible bande de pixels, très proche du bloc à prédire. Ainsi, notre prédiction est moins souvent retenue dans les cas où nous formons une moyenne. Ceci se caractérise sur la courbe 4.14 par une forte chute en amplitude symbolisant la faible fréquence observé pour les valeurs faibles du nombre d'itérations.*

Nous obtenons une entropie de $H_{it} = 4.6164$ bits/bloc pour transmettre le nombre d'itérations

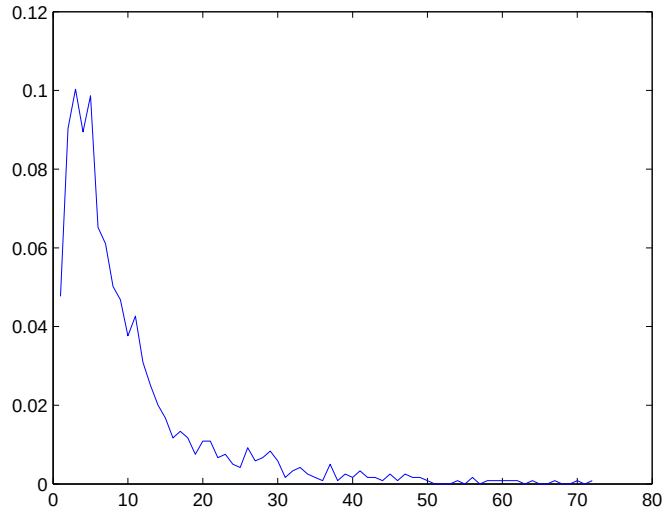



FIG. 4.14 – Histogramme des valeurs du nombre d'itérations pour les blocs sélectionnés comme meilleure prédiction

nécessaire pour former notre prédiction. Le coût total des itérations à transmettre est le suivant :

$$\text{coût}_{it} = H_{it} * N * p \tag{4.2}$$

Le débit obtenu sans coder cette information supplémentaire est, dans le cadre de cette expérimentation, égal à 551 160 bits. L'application numérique de la formule (4.2) de calcul du surcoût donne la valeur $\text{coût}_{it} = 5\ 673.61$ bits, correspondant à une augmentation relativement faible de 1.03 % en débit.

Cependant, plus on augmente la valeur du pas de quantification, plus le coût de signalisation que nous introduisons peut devenir pénalisant. A fort *QP*, l'image est très quantifiée et le débit est sensiblement réduit. En outre, notre prédiction continue à être sélectionnée à hauteur de 30 % ce qui augmente l'impact de notre surcoût relativement au nombre de bits total à transmettre. Le tableau 4.3 répertorie pour plusieurs valeurs de *QP*, le surcoût introduit par notre méthode.

| *QP* | Sélection de notre prédiction (%) | $H_{it}$ (bits/bloc) | Surcoût (%) |
|---|---|---|---|
| 21 | 30.0 | 4.62 | 1.03 |
| 26 | 29.2 | 4.57 | 1.61 |
| 30 | 28.5 | 4.60 | 2.21 |
| 35 | 27.8 | 4.46 | 3.46 |

TAB. 4.3 – Surcoût de codage en fonction de la valeur du *QP*

***Remarque importante :***

*Nous faisons ici une évaluation du surcoût a posteriori puisque nous l'évaluons après que l'image est encodée, pour connaître la répartition des valeurs du nombre de boucles retenu pour chaque bloc. Idéalement, il faudrait que les décisions prises pour chaque bloc tiennent compte du surcoût lié à notre prédiction. Il faudrait ainsi avoir une décision lagrangienne sur les modes*

*intra et un processus de calcul du nombre de bits nécessaire pour notre prédiction, qui se fasse à la volée et, pourquoi pas, de manière contextuelle, à la manière du CABAC (cf section 1.6.2.3).*

Nous avons ensuite calculé les gains Bjontegaard en tenant compte du coût de signalisation du nombre de boucles réalisées.

L'impact de cette signalisation supplémentaire n'est pas négligeable. Nous pouvons voir, d'après le tableau 4.4 que cela réduit les performances de 0.14 dB à débit constant et de 1.86 % en débit, pour une qualité donnée. Cependant les résultats obtenus en tenant compte du coût de signalisation restent légèrement supérieurs à ceux obtenus sans critère d'arrêt basé EQM et ne nécessitant aucune signalisation supplémentaire.

| | Gain (dB) | Gain (%) |
|---|---|---|
| Sans prise en compte de coût$_{it}$ | + 0.43 | - 5.61 |
| Avec prise en compte de coût$_{it}$ | + 0.29 | - 3.75 |
| Sans critère d'arrêt EQM | + 0.23 | - 3.04 |

TAB. 4.4 – Résultats Bjontegaard en tenant compte du surcoût de codage

### 4.5.5 Résultats sur différentes images

Le tableau 4.5 présente les résultats obtenus avec l'algorithme du *Matching Pursuit* pour différentes images et différentes configurations de l'encodeur. Nous avons utilisé les images présentées en figure 4.15.

Les résultats sont variables et dépendent bien évidemment du contenu de l'image à encoder. Les meilleurs résultats sont obtenus pour les images contenant de nombreuses textures bi-dimensionnelles complexes. Par exemple, sur l'image *Laine*, notre approche de prédiction basée sur les représentations parcimonieuses, réussit particulièrement bien à générer la texture attendue, là où la prédiction de la norme a des lacunes (cf image 4.16). Nous constatons une réduction de débit de près de 10 %, associée à un gain en qualité de 0.79 dB. Nous réussissons en effet, à reproduire le motif avec une certaine finesse dans les détails alors que les modes directionnels de la norme sont particulièrement mis en difficulté face aux signaux texturés de cette image.

Quant aux images *Boule* et *Barbara*, les gains en qualité et en débit sont également significatifs, bien qu'inférieurs. La structure texturée présente en grande proportion dans l'image *Boule* permet de surpasser les performances obtenues avec l'image *Barbara*.

| | Sans calcul du surcoût | | Avec calcul du surcoût | |
|---|---|---|---|---|
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| *Barbara* | + 0.43 | - 5.61 | + 0.29 | - 3.75 |
| *Boule* | + 0.77 | - 8.38 | + 0.54 | - 5.94 |
| *Laine* | + 1.00 | - 12.75 | + 0.79 | - 10.06 |
| *Lena* | + 0.08 | - 1.68 | - 0.01 | + 0.26 |
| *Zèbre* | + 0.03 | - 0.41 | - 0.03 | + 0.36 |
| *Raven* | + 0.12 | - 2.15 | + 0.06 | - 1.05 |

TAB. 4.5 – Résultats Bjontegaard de la prédiction parcimonieuse en $8 \times 8$, pour différentes images

En figure 4.17, nous présentons une comparaison visuelle des images de prédiction : celle

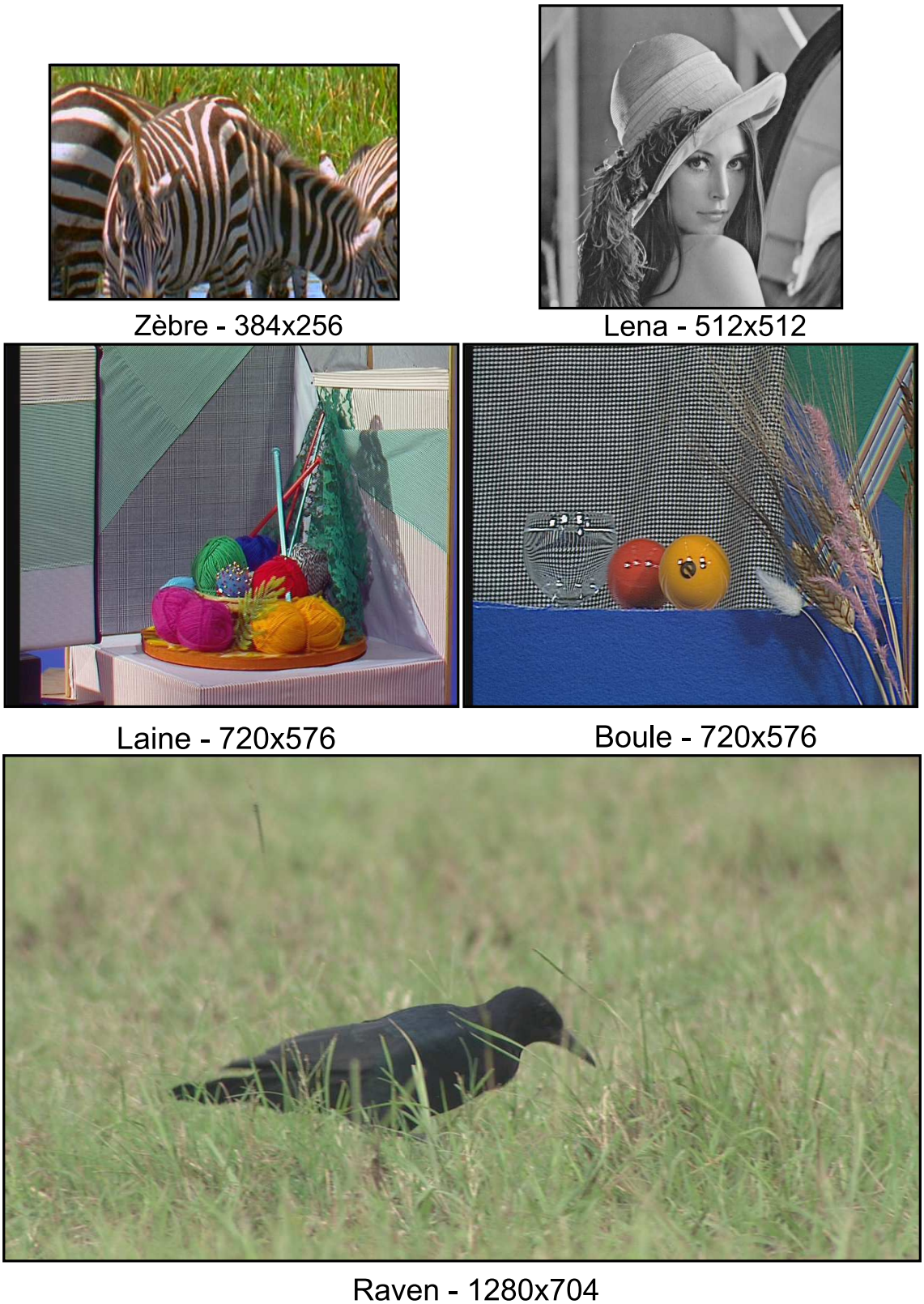



FIG. 4.15 – Images de tests

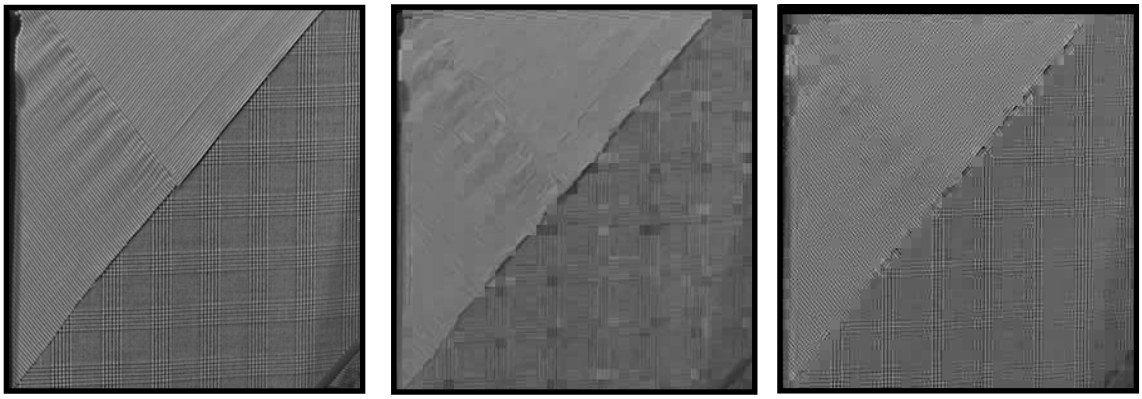



FIG. 4.16 – Détail de prédiction sur l'image *laine*

obtenue à partir des seuls modes de la norme, celle obtenue à partir de notre méthode et enfin, celle correspondant à la prédiction obtenue lorsque nous substituons notre prédiction à un mode AVC. On constate que la prédiction de la norme est performante sur les motifs directionnels en haut, à droite de l'image mais également sur les épis de blé et sur la texture granuleuse de la table, présente en avant-plan. En revanche, la structure périodique du fond de l'image n'est pas du tout récupérée. En *parfaite* complémentarité, notre prédiction peine sur les structures directionnelles ou sur la granularité aléatoire de la table mais retranscrit avec succès la structure périodique du fond. On peut apprécier la qualité de l'image de prédiction globale, qui inclut les modes de la norme mais également notre prédiction parcimonieuse, présentant ainsi une amélioration visuelle sensible de la qualité de l'image de prédiction.

Les images *Lena*, *Zèbre* et *Raven* sont de plus grands défis pour notre méthode. Nous n'avons obtenu aucun gain significatif sur ces images. L'image *Lena* est composée de nombreuses zones assez plates, uniformes. Comme le mode DC de la norme forme une moyenne plus précise que nous ne pouvons le faire (nous avons évoqué cette problématique en section 4.5.4), notre prédiction ne suffit pas améliorer les performances de la prédiction du standard. Sur l'image *Zèbre*, on aurait pu s'attendre à une reconstruction avantageuse par notre méthode, des rayures des animaux, mais ce n'est pas le cas. Les directions des motifs semblent correspondent assez correctement à certains des modes de la norme H.264 / AVC, notamment le mode 0 (vertical) ou encore le mode 6 (horizontal-bas). La prédiction de l'image *Raven* est elle aussi particulièrement difficile. Dans la mesure où notre approche peut difficilement reproduire cette texture stochastique des brins de l'herbe, difficulté accentuée par le fait que nous avons testé en blocs $8\times8$, nous peinons à surpasser la norme du fait de notre surcoût de codage.

### 4.5.6 Influence du codage entropique sur les résultats

Les expérimentations présentées dans les sections précédentes ont été réalisées avec pour configuration du codage entropique, un codeur UVLC. Pour des raisons de rapidité de simulations, nous avons en effet choisi d'effectuer la majorité des tests avec un codeur UVLC et non le CABAC (présenté en section 1.6.2.3). Il est toutefois incontournable d'évaluer notre méthode de prédiction parcimonieuse, dans le contexte d'un codage entropique basé sur le CABAC. Son efficacité est redoutable et risquerait d'inverser la tendance des résultats obtenus jusqu'à présent.

Nous constatons d'après le tableau 4.6 que la tendance reste identique. Les résultats en qualité diminuent approximativement de 0.1 dB et augmentent de 1 % en débit. Cependant, nous

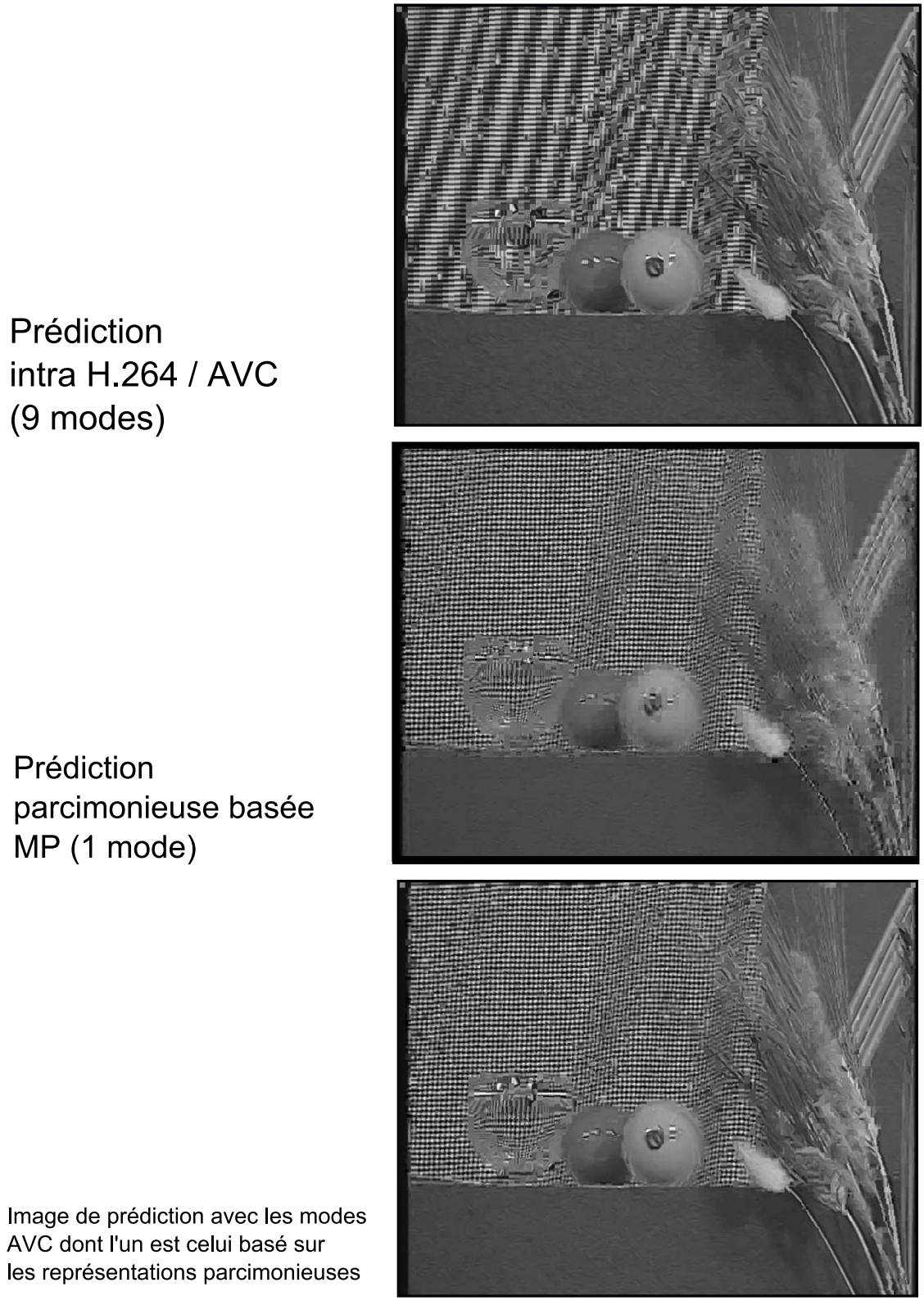



FIG. 4.17 – Exemple d'images de prédiction

| | Sans calcul du surcoût | | Avec calcul du surcoût | |
|---|---|---|---|---|
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| Codage UVLC | + 0.43 | - 5.61 | + 0.29 | - 3.75 |
| Codage CABAC | + 0.35 | - 4.80 | + 0.20 | - 2.74 |

TAB. 4.6 – Impact du codage entropique et du surcoût de codage en prédiction $8 \times 8$

parvenons toujours à améliorer la prédiction intra image de la norme, même si le codage entropique utilise le CABAC. Les évolutions relatives au changement de codeur restent similaires pour les différentes configurations proposées (cf tableau 4.7).

| | Sans calcul du surcoût | | | |
|---|---|---|---|---|
| | Codage UVLC | | Codage CABAC | |
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| $4 \times 4$ | + 0.72 | - 8.55 | + 0.59 | - 7.50 |
| $8 \times 8$ | + 0.43 | - 5.61 | + 0.35 | - 4.80 |
| $4 \times 4$ et $8 \times 8$ | + 0.59 | - 7.56 | + 0.48 | - 6.52 |
| $4 \times 4$, $8 \times 8$ et $16 \times 16$ | + 0.59 | - 7.60 | + 0.48 | - 6.58 |

TAB. 4.7 – Impact du codage entropique pour différentes partitions

### 4.5.7 Comparaison des algorithmes

Le tableau 4.8 présente les performances obtenues pour les différents algorithmes suivants : le *matching pursuit*, l'*orthogonal matching pursuit* et le *global matched filter*. Rappelons ici que les deux premiers algorithmes ont une procédure itérative qui recherche la meilleure représentation en cherchant à minimiser l'erreur résiduelle. Le GMF correspond au *basis pursuit* dont le critère introduit un terme de pénalisation sur la parcimonie de la représentation. Notre but est de voir comment se comportent ces algorithmes dans le cadre de la prédiction.

On constate que les performances sont sensiblement similaires pour les trois algorithmes. On peut cependant remarquer que le MP a des gains légèrement supérieurs. La première remarque que l'on peut avancer est que le dégré de parcimonie de la représentation n'influe par directement sur la qualité de la prédiction. L'OMP et le GMF fournissent en effet des représentations plus parcimonieuses que le MP, mais c'est ce dernier qui conduit dans la majorité des cas à de meilleurs résultats Bjontegaard lorsqu'il est appliqué à la prédiction. Dans la suite de nos travaux, nous utiliserons par défaut le *matching pursuit* car c'est celui qui, en l'état, répond le mieux à notre problématique.

| | MP | | OMP | | GMF | |
|---|---|---|---|---|---|---|
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| $4 \times 4$ | + 0.72 | - 8.55 | + 0.64 | - 7.65 | + 0.62 | - 7.26 |
| $8 \times 8$ | + 0.43 | - 5.61 | + 0.41 | - 5.36 | + 0.44 | - 5.74 |
| $4 \times 4$ et $8 \times 8$ | + 0.59 | - 7.56 | + 0.54 | - 6.95 | + 0.55 | - 6.95 |
| $4 \times 4$, $8 \times 8$ et $16 \times 16$ | + 0.59 | - 7.60 | + 0.55 | - 7.19 | + 0.53 | - 6.91 |

TAB. 4.8 – Comparaison des résultats Bjontegaard entre la prédiction parcimonieuse basée sur l'algorithme du MP, celle basée sur celui de l'OMP et celle obtenue via le GMF (sans prise en compte du surcoût).

### 4.5.8 Critère d'arrêt lagrangien

Afin de favoriser la représentation parcimonieuse qui conduira au meilleur compromis, entre qualité de reconstruction et coût de codage, nous avons introduit un critère d'arrêt basé sur une optimisation débit / distorsion, pour remplacer celle sur l'EQM. Pour être cohérent avec les décisions prises par la suite lors de la sélection des modes intra, nous activons la décision lagrangienne sur le choix du meilleur mode. Le paramètre *RDOPT* de l'encodeur utilisé permet d'activer cette décision en mettant sa valeur à 1.

La figure 4.18 présente plus précisément les différentes décisions que nous évoquons. La sélection du meilleur mode intra peut se faire de deux façons, soit :

* selon un critère uniquement basé sur une mesure de distorsion (ici, la SAD) et dans ce cas, $RDOPT = 0$
* selon un critère débit / distorsion, activé par $RDOPT = 1$

Concernant notre algorithme, nous avons le choix entre deux critères d'arrêt :

* soit une mesure de distorsion : l'EQM
* soit un critère lagrangien pour une optimisation débit / distorsion

Lorsque nous choisissons, un critère d'arrêt basé sur une optimisation débit / distorsion, nous activons également la décision lagrangienne pour le choix du meilleur mode intra.

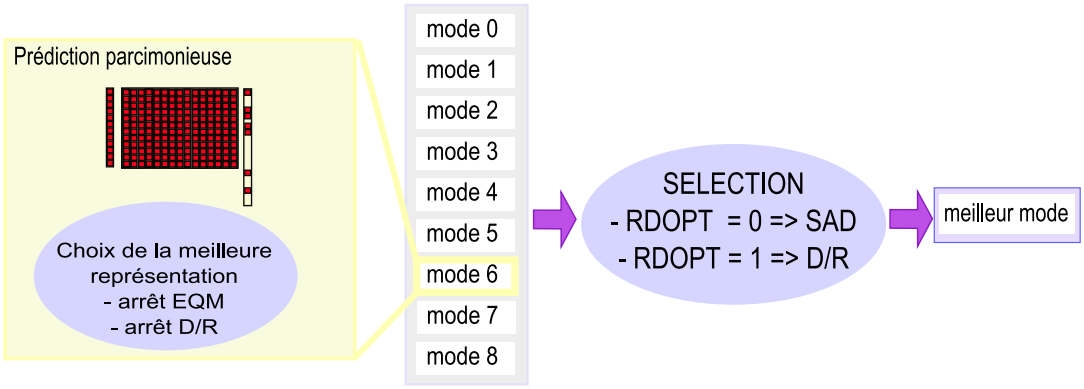



FIG. 4.18 – Décisions au sein des algorithmes et celles du meilleur mode intra

Nous répertorions dans le tableau 4.9, les performances obtenues en fonction des critères d'arrêt choisis. La décision lagrangienne au sein de l'algorithme améliore sensiblement les résultats. De plus, la technique reste compétitive même lorsque l'on tient compte du coût de codage de l'information supplémentaire à transmettre.

| | Sans surcoût | | Avec surcoût | | Surcoût moyen (%) |
|---|---|---|---|---|---|
| Configuration | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) | |
| MP - RDOPT = 0 | + 0.23 | - 3.04 | - | - | - |
| MP - RDOPT = 1 | + 0.22 | - 2.93 | - | - | - |
| MP - arrêt EQM - RDOPT = 0 | + 0.43 | - 5.61 | + 0.29 | - 3.75 | 2.08 |
| MP - arrêt EQM - RDOPT = 1 | + 0.40 | - 5.32 | + 0.26 | - 3.43 | 2.11 |
| MP - arrêt $D + \lambda R$ - RDOPT =1 | + 0.57 | - 7.38 | + 0.37 | - 4.84 | 2.86 |

TAB. 4.9 – Comparaison des performances selon le critère d'arrêt sélectionné

### 4.5.9 Optimisation du voisinage d'approximation

Pour améliorer la prédiction, nous avons vu la pertinence de l'ajout d'une fonction de pondération lorsque le signal n'est pas stationnaire au sein du voisinage choisi. On constate par ailleurs, d'après différents cas exposés en figure 4.19, qu'il n'est pas toujours pertinent d'utiliser l'ensemble des blocs limitrophes au bloc courant, comme support de prédiction. Dans certains cas, il peut être plus avantageux de n'utiliser qu'un seul bloc, choisi en fonction de la direction du motif ou du type de texture à reproduire. En vert sur la figure, une proposition de voisinage formé d'un seul bloc, *a priori* le plus corrélé aux données sources.

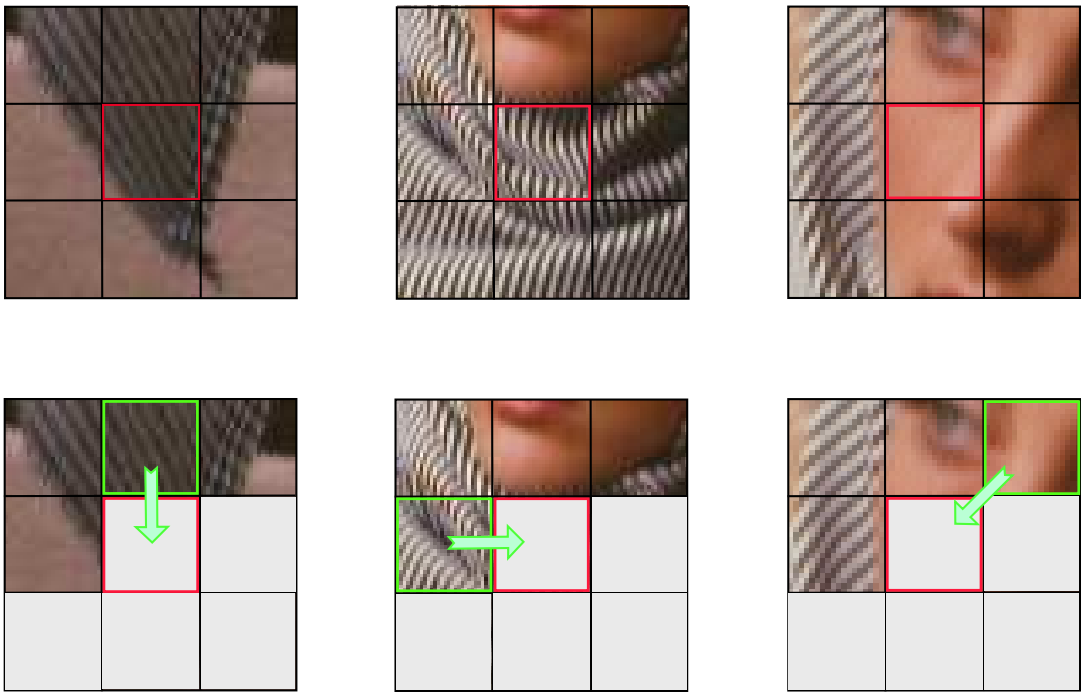


FIG. 4.19 – Exemple de voisinages basés sur un bloc

Comme nous ne pouvons savoir quel bloc du voisinage sera le plus adéquat sans faire au préalable une analyse structurelle du contenu, nous proposons de générer autant de prédictions qu'il y a de blocs dans le voisinage, chacune étant issue d'une extrapolation des pixels d'un seul bloc.

Parmi ces prédictions, on choisit ensuite la meilleure en se basant sur un critère lagrangien. Afin de ne pas se pénaliser dans les cas où le signal est stationnaire, on propose de mettre également la prédiction habituelle, basée sur l'ensemble des blocs du voisinage, en compétition avec ces nouvelles prédictions, comme l'illustre la figure 4.20.

Pour ensuite évaluer les performances de cette approche, nous devons prendre en compte deux coûts de signalisation :

* le coût sur le nombre d'itérations puisque l'on s'arrête sur un critère utilisant l'image source (ici le critère lagrangien).

* le coût de signalisation du meilleur mode que nous avons formé via un voisinage d'un seul bloc.

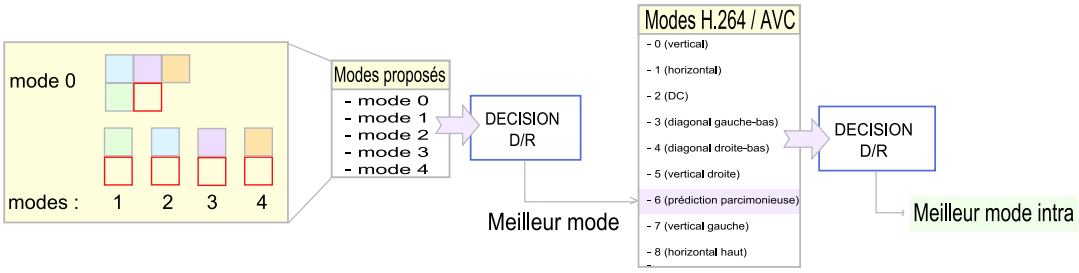



FIG. 4.20 – Prédiction parcimonieuse basée sur un voisinage d'un seul bloc

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... Matching Pursuit
* Dictionnaire ..... DCT
* *QP* ..... $21 - 26 - 30 - 35$
* Partition ..... $8 \times 8$
* Sélection des modes intra ..... Optimisation D / R
* Seuil ..... 1
* Critère d'arrêt ..... Optimisation D / R

Bien que le surcoût soit conséquent, 3 % en moyenne sur les quatre *QP*, l'approche par segmentation permet de gagner 0.10 dB en qualité et 1.24 % en débit (cf tableau 4.10) en tenant compte des surcoûts de signalisation, par rapport à la prédiction parcimonieuse classique faite sur le voisinage complet.

| | Sans surcoût | | Avec surcoût(s) | |
|---|---|---|---|---|
| Configuration | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| MP - voisinage complet - arrêt $D + \lambda R$ - RDOPT =1 | + 0.57 | - 7.38 | + 0.37 | - 4.84 |
| MP - voisinage un bloc - arrêt $D + \lambda R$ - RDOPT =1 | + 0.77 | -9.98 | + 0.47 | - 6.08 |

TAB. 4.10 – Résultats de la prédiction parcimonieuse avec segmentation du voisinage

### 4.5.10 Application : raffinement de la prédiction inter - image

#### 4.5.10.1 Débruitage de la prédiction inter

Nous proposons de tester la prédiction parcimonieuse dans le cadre hybride où l'on exploite simultanément une information causale issue de l'image courante et une information supplémentaire issue du processus inter-images.

L'application consiste à raffiner la prédiction inter-images en se basant sur la connaissance des pixels voisins de l'image courante, utilisés jusqu'alors comme support de prédiction intra-images. La figure 4.21 est une illustration de la méthode : on recopie au sein du bloc courant, la prédiction issue de l'estimation / compensation de mouvement faite à partir d'une image de référence. On utilise ensuite l'information du voisinage causal et la prédiction temporelle positionnée dans le

bloc courant pour former une nouvelle prédiction.

Pour évaluer les performances, il est nécessaire de prendre en compte le coût de codage des composantes des vecteurs de mouvement, au même titre qu'une prédiction inter-images classique. De plus, il faudrait considérer cette prédiction mixte comme un mode inter supplémentaire. Cela impliquerait l'ajout d'un second mode pour la partition inter considérée, un processus de sélection du meilleur de ces deux modes proposés et la prise en compte du coût de signalisation du choix qui a été retenu. Pour des raisons de simplification de mise en place, nous avons procédé différemment en élaborant un processus qui permet d'évaluer, en relatif et non de manière absolue, la qualité de notre prédiction.

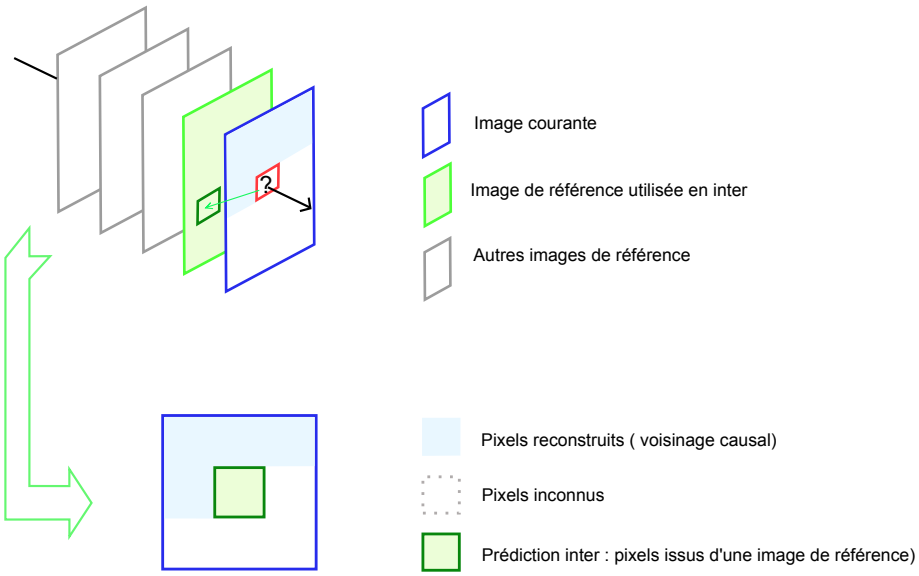



FIG. 4.21 – Prédiction parcimonieuse mixte

Pour une séquence donnée, nous nous sommes placés dans un schéma *IPPP*, en autorisant uniquement la prédiction inter $8 \times 8$ (les autres partitions inter étant inactives). L'image *I* est codée en intra avec uniquement les modes de la norme, sans prédiction parcimonieuse. Quant aux images *P*, voici comment nous avons procédé.

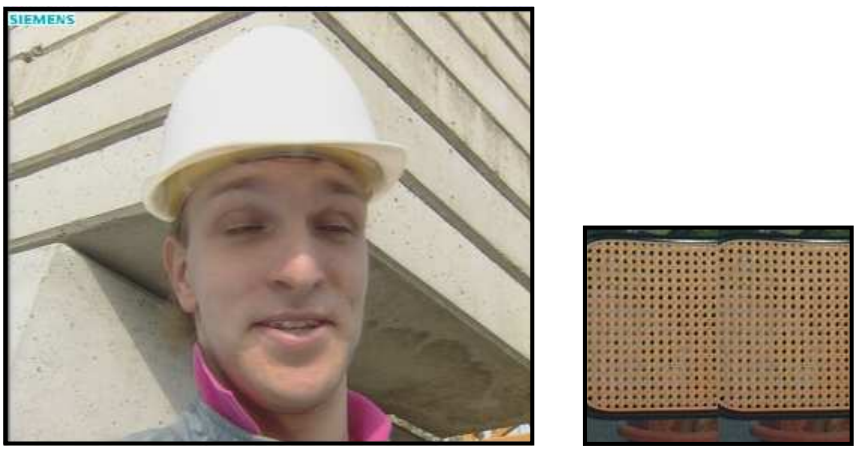



FIG. 4.22 – Séquences *foreman* ($352 \times 288$) et *panslow* ($176 \times 144$)

Nous avons substitué la prédiction issue d'une estimation / compensation de mouvement en partition $8 \times 8$, à un mode *intra*, en neutralisant le processus de sélection inter. Ce protocole a constitué notre référence : la prédiction inter est alors insérée dans le processus de sélection des modes intra-images.

Pour ensuite juger de la qualité de notre prédiction mixte, nous avons substitué notre prédiction à un autre mode intra. Ainsi, les deux prédictions : inter $8 \times 8$ de la norme et notre prédiction mixte sont mises en concurrence au sein du processus de sélection intra, si bien que la question du coût de codage des vecteurs de mouvement est éludée. Il s'agit principalement d'évaluer l'apport du raffinement en terme de prédiction sur la prédiction inter-images classique.

Nous nous sommes intéressés à trois séquences différentes : la séquence *foreman*, *panslow* (cf figure 4.22) et *laine*. La figure 4.23 présente une image de taille $720 \times 576$ extraite de la séquence *laine*. Nous avons travaillé sur une imagette $176 \times 144$ extraite de cette séquence, correspondant une partie texturée dotée d'un mouvement de translation vertical. La séquence *panslow* correspond à la juxtaposition de deux textures de même type animées chacune d'une faible translation horizontale.

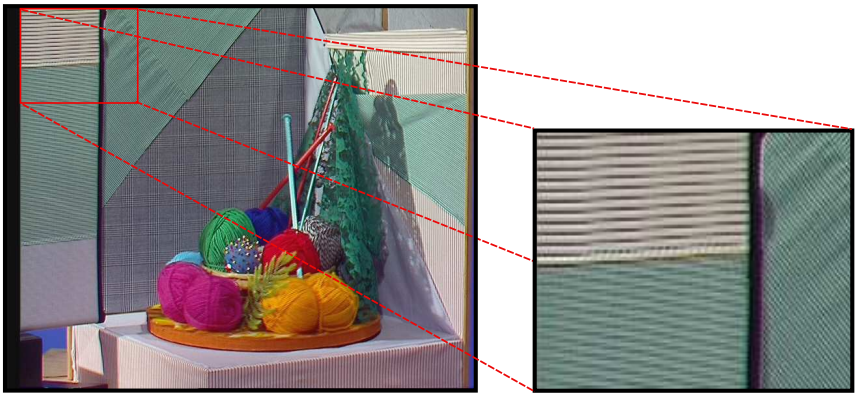


FIG. 4.23 – Séquence *laine* et zone de travail $176 \times 144$ présentée à droite

*Configuration de test* :

* Séquences sources ..... *Laine, Foreman, panslow*
* Algorithme ..... Matching Pursuit
* *QP* ..... 21 – 26 – 30 – 35
* Nombre d'images à encoder ..... 10
* Schéma ..... *IPPP*
* Partition ..... Inter $8 \times 8$
* Sélection des modes intra ..... SAD
* Seuil ..... 1
* Critère d'arrêt ..... EQM

Les meilleurs résultats (cf tableau 4.11) ont été obtenus sur la séquence extraite de la séquence *laine*. Pour ce type de signal, bruité par un phénomène d'*aliasing* [1], notre technique semble corriger des erreurs où l'estimateur de mouvement n'aurait pas réussi à sélectionner le bon motif.

En revanche, les résultats sur la séquence *panslow* présentent une détérioration de la qualité associée à une augmentation du débit, ce qui signifie que l'on ne réussit pas à raffiner la prédiction

[1] Effet de crênelage se traduisant par une pixellisation parasite au sein de l'image.

inter : on la détériore en rajoutant du bruit. La prédiction inter de la norme est très performante pour cette séquence. En effet, le signal est très périodique, l'image ne subit aucune déformation géométrique (le mouvement est translationnel, horizontal), ainsi le processus d'estimation / compensation de mouvement, dont la précision s'étend au quart de pixel, récupère très facilement le bon motif.

| | Sans surcoût | | % de sélection | | |
|---|---|---|---|---|---|
| Séquence | Gain (dB) | Gain (%) | AVC | Prédiction parcimonieuse | Autres modes |
| *laine* | + 0.29 | - 5.23 | 60 | 30 | 10 |
| *foreman* | + 0.09 | - 2.03 | 75 | 15 | 10 |
| *panslow* | - 0.03 | + 0.60 | 98 | 2 | 0 |

TAB. 4.11 – Résultats bjontegaard et pourcentages de sélection des prédictions temporelles de la norme et basée représentations parcimonieuses

#### 4.5.10.2 *Dé-crossfading*

Les *fading* sont des effets appliqués sur plusieurs images, en général pour marquer des transitions entre différentes scènes d'une vidéo. Les corrélations temporelles entre images successives sont alors très faibles, ce qui peut perturber le processus de prédiction inter-images.

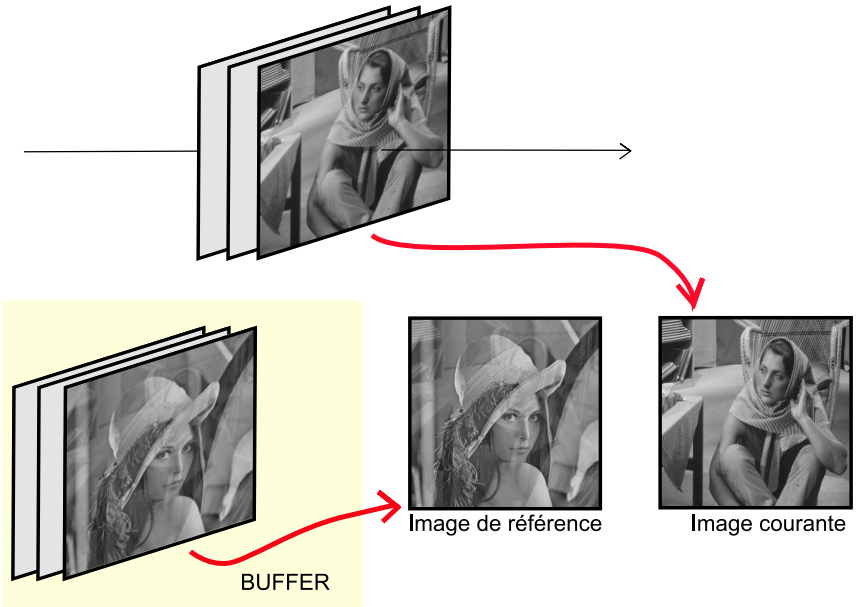



FIG. 4.24 – Situation de crossfading

Prenons l'exemple expérimental suivant présenté en figure 4.24. L'image courante est l'image *Barbara*. L'image de référence que l'on suppose être utilisée par l'estimation de mouvement correspond à une image où les images *Lena* et *Barbara* ont été mélangées en proportion choisie. Sur cette figure, l'image *Barbara* est pondérée d'un facteur de 0.25 par rapport à *Lena*. Si bien que l'image de référence ici présentée, correspond à un *crossfading* de ces deux images.

L'objectif de cette étude est d'évaluer les performances des représentations parcimonieuses, dans le cadre d'une application de *dé-crossfading* pour améliorer la prédiction inter-images. Dans

ces cas particuliers de *crossfading*, la norme ne possède comme outil que la prédiction pondérée (évoquée en section 1.6.3.3), laquelle est plutôt dédiée au phénomène de fading. En ce qui nous concerne, l'idée est de corriger la valeur des pixels issus de la référence, bruitée par le *crossfading*.

Nous proposons une approche similaire à l'application précédemment présentée, le débruitage de la prédiction inter. L'enjeu est de réussir à extraire du bloc de prédiction inter bruité, les données correspondant uniquement à l'image courante, ici *Barbara*, en se basant sur la connaissance du voisinage local.

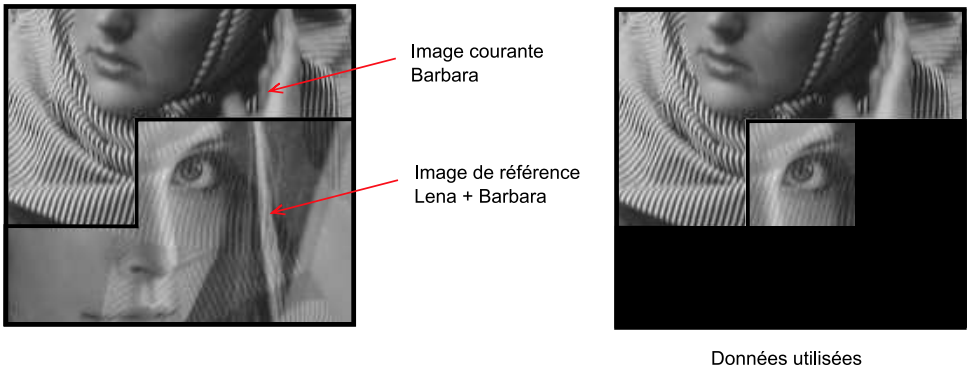



FIG. 4.25 – Principe du *dé-crossfading* basé sur les représentations parcimonieuses

Nous nous sommes positionnés dans un schéma non-réaliste dans le sens où nous avons rapatrié les pixels *colocalisés* de l'image de référence et non la prédiction inter issue de l'estimation / compensation de mouvement. Le but de cette expérimentation est principalement de présenter une autre application possible de la prédiction parcimonieuse. La figure 4.25 montre sur un exemple, les données utilisées pour générer notre prédiction : on exploite simultanément les pixels du voisinage causal de l'image courante (*Barbara*) et ceux correspondant au bloc à prédire, en colocalisé dans l'image de référence (ici, *Lena* et 0.25 % de *Barbara*).

Nous présentons en figure 4.26, les images de prédiction obtenues pour différentes valeurs de mélange : 15 %, 25 % et 75 % de *Barbara*. Pour évaluer de manière approfondie, les performances de cette technique, il faudrait mettre cette prédiction en concurrence avec les méthodes couramment employées pour résoudre ce type de problème. L'avantage de notre approche est de faire une compensation d'illumination fine. Le processus de sélection des atomes permet de ne représenter que les données pertinentes, tout en corrigeant des défauts d'illumination par le biais des facteurs de pondération, contenus dans la représentation parcimonieuse $x$. On peut supposer que les résultats soient meilleurs que ceux obtenus via des techniques de compensation globale dont le terme correctif est unique pour l'ensemble des pixels du bloc.

Nous avons néanmoins voulu obtenir des résultats permettant de jauger le potentiel de la technique, dans un schéma de compression d'images. Une première approche pour évaluer l'apport de la technique est de se comparer à la prédiction inter image *crossfadée*. Dans le même esprit que pour le raffinement de la prédiction inter (section 4.5.10.1), nous avons substitué la prédiction extrait de l'image de référence *crossfadée* à un mode intra de la norme.

Les résultats que nous obtenons, présentés dans le tableau 4.12, sont assez modestes. Les tendances les plus intéressantes ont été observées, sans grande surprise, en prédiction $4 \times 4$. Pour un facteur de *crossfading* de 0.25, nous observons un gain de 0.31 dB en qualité et une diminution de débit de 3.53 %. La petite taille des blocs permet de récupérer de manière assez précise le signal utile, en l'occurence, *Barbara.* Cependant le surcoût lié à notre prédiction est prohibitif au vue du nombre de blocs $4 \times 4$ présents dans cette image de dimensions $512 \times 512$.

Afin d'évaluer la méthode, sans être pénalisé par le surcoût, nous avons également calculé


FIG. 4.26 – Résultat du *dé-crossfading* basé sur les représentations parcimonieuses en prédiction $4 \times 4$

| | | | Sans surcoût | | Avec surcoût | |
|---|---|---|---|---|---|---|
| Arrêt EQM | Prédiction | Facteur | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| | $4 \times 4$ | 15 % | + 0.23 | - 2.27 | - | - |
| | | 25 % | + 0.31 | - 3.53 | - | - |
| | | 75 % | + 0.40 | - 5.15 | - | - |
| | $8 \times 8$ | 15 % | + 0.19 | - 2.53 | + 0.13 | - 1.72 |
| | | 25 % | + 0.23 | - 2.96 | + 0.15 | - 1.90 |
| | | 75 % | - 0.72 | + 5.35 | - 1.07 | + 9.65 |
| Sans arrêt EQM | Prédiction | Facteur | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| | $4 \times 4$ | 25 % | + 0.20 | - 2.38 | - | - |
| | $8 \times 8$ | 25 % | + 0.12 | - 1.53 | - | - |

TAB. 4.12 – Résultats Bjontegaard concernant le *dé-crossfading*

les performances, sans critère d'arrêt supplémentaire, ici basé sur l'EQM. Rappelons que dans ce cas de figure, l'algorithme itère jusqu'à ce que l'énergie de l'erreur de reconstruction, sur le voisinage causal, soit inférieure à un seuil, préalablement fixé. Dans ce contexte, les performances en prédiction $4 \times 4$, avec ce seuil fixé à 35, diminuent pour n'avoir plus que 0.20 dB de gain en qualité et une diminution de 2.38 % en débit.

Les performances du *dé-crossfading* parcimonieux en prédiction $8 \times 8$ sont encore plus discutables. Après prise en compte du surcoût, on atteint une hausse de 0.15 dB en qualité et une perte de 1.90 % en débit, pour un facteur de *crossfading* de 0.25.

D'un point de vue purement visuel, les images de prédiction *dé-crossfadée* que nous avons obtenues, pouvaient laisser présager des résultats intéressants dans le cadre de la compression. Il s'avère que les gains Bjontegaard obtenus ne sont pas aussi élevés que nous pouvions l'espérer. On peut néanmoins imaginer utiliser cette technique dans un autre cadre, différent de la compression où l'enjeu majeur serait la séparation de signaux ou encore des applications de débruitage.

## 4.6 Résultats expérimentaux dans le cadre SVC

Nous présentons dans cette partie, les résultats que nous avons obtenus dans le cadre de l'amélioration de la prédiction spatiale inter-couches du standard SVC. Notre expérimentation consiste à utiliser la couche de base sur-échantillonnée comme information supplémentaire pour former une prédiction spatiale mixte basée sur les représentations parcimonieuses. La figure 4.27 présente les différents voisinages utilisés : celui de la couche courante et celui de la couche de base. Nous avons utilisé les filtres recommandés par la norme SVC pour sous-échantillonner et sur-échantillonner l'image. Le filtre de sous-échantillonnage $h$ est un sinus cardinal fenêtré (proposition JVT - R006). Celui du sur-échantillonnage, $g$, est normalisé et correspond à un filtre de Lanczos (proposition JVT - U042).

Cependant nous ne nous sommes pas positionnés dans un encodeur SVC : nous avons émulé son fonctionnement en proposant des prédicteurs issus d'une couche de base. Comme cette couche de base est commune dans nos manipulations, le débit pour coder cette couche est commun et peut ne pas être pris en compte. Dans ce cadre là, nous n'avons pas les valeurs de débit SVC à proprement parler mais la comparaison des débits pour coder la couche d'amélioration suffit à évaluer les performances des prédictions proposées.

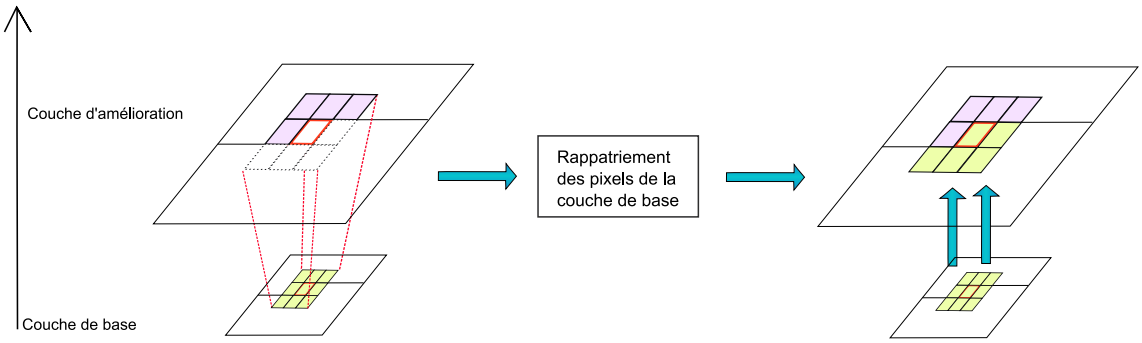



FIG. 4.27 – Voisinages utilisés dans le cadre de l'application de la prédiction spatiale inter-couches

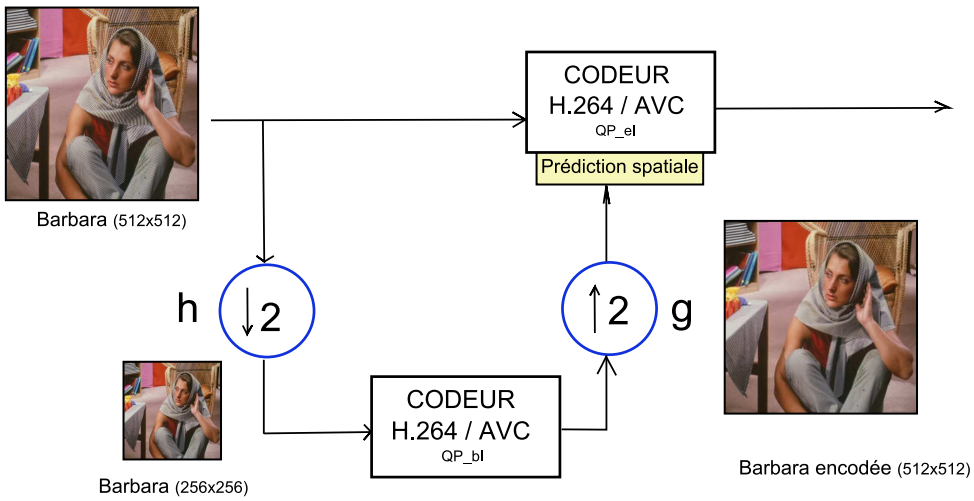



FIG. 4.28 – Emulation de SVC

Le processus est décrit sur le schéma 4.28. Nous générons la version sous-échantillonnée de l'image de la couche courante via le filtre $h$. Ensuite, nous encodons cette couche de base à l'aide d'un encodeur H.264 / AVC, pour différents $QP_{bl}$, nécessaires pour évaluer les gains Bjontegaard. Ces couches de base encodées sont ensuite sur-échantillonnées à l'aide du filtre $g$ pour correspondre à la résolution de la couche courante. La deuxième étape consiste à générer notre prédiction parcimonieuse, pour différents $QP_{el}$, en utilisant les voisinages décrits plus haut en figure 4.27, faisant référence à la couche de base encodée (au $QP_{bl}$ correspondant) et sur-échantillonnée.

Pour évaluer la qualité de notre prédiction, nous devons avant tout former ce qui sera notre référence, dans cette émulation de SVC. La prédiction spatiale inter-couches du standard SVC correspond au bloc colocalisé dans la couche de base sur-échantillonnée. On substitue cette prédiction à un mode intra de l'encodeur H.264 / AVC. Dans un vrai encodeur SVC, la prédiction spatiale inter-couches correspond à un $10^{\text{ième}}$ mode intra. Ici, toujours pour des raisons de simplification et de rapidité d'implémentation, nous nous contentons de remplacer un des modes de la norme pour éviter de modifier la syntaxe.

Nous proposons la manipulation suivante pour évaluer notre prédiction par rapport à celle du standard SVC. Nous substituons la prédiction de la norme et notre prédiction à deux des modes intra. Les deux prédictions sont ainsi mises en concurrence directe. Le dictionnaire utilisé est un dictionnaire contenant deux fois plus de fonctions de base que pour une prédiction parcimonieuse classique.


FIG. 4.29 – Comparaison des images de prédiction dans le cadre de la prédiction spatiale inter-couches

Nous avions évoqué en section 4.4.2, qu'il était pertinent d'ajouter des fonctions de base au dictionnaire, pour qu'il soit redondant (il a effectivement été expliqué précedemment en section 4.2.1 que la redondance du dictionnaire est directement liée aux nombres d'observations nulles). Nous avons choisi de former un dictionnaire de taille double, à partir d'une DCT et d'atomes directionnels que nous évoquerons plus tard, en section 6.2.2.

La première conclusion que nous pouvons tirer, d'après les résultats présentés dans le tableau 4.13, est que nous améliorons sensiblement la prédiction spatiale inter-couche de la norme. Le raffinement que nous proposons permet de gagner 0.64 dB en qualité et de réduire le débit de 8.58

%. En revanche, le surcoût est prohibitf et pénalise grandement les performances de la méthode. Les représentations obtenues sont en effet peu parcimonieuses. Il semble donc que la convergence vers la solution du problème soit laborieuse.

| | Gain (dB) | Gain (%) |
|---|---|---|
| Sans calcul du surcoût | + 0.64 | - 8.58 |
| Avec calcul du surcoût | + 0.24 | - 3.09 |

TAB. 4.13 – Résultats Bjontegaard dans le cadre de la prédiction spatial inter couche de la norme SVC

Remarque: *La flexibilité introduite par les valeurs de seuil, inhérent aux algorithmes liés aux représentations parcimonieuses, permettrait de générer une scalabilité en SNR. En modifiant la valeur du seuil, on pourrait choisir le degré de raffinement désiré et ainsi être en adéquation avec la philosophie d'un encodage scalable.*

## 4.7 Discussion sur le critère d'arrêt non-causal

Il est certain que l'ajout d'un critère d'arrêt basé sur l'erreur quadratique moyenne ou sur une optimisation lagrangienne débit / distorsion, nous permet d'améliorer significativement les résultats. Cela peut cependant devenir un réel désavantage lorsque l'on prend en compte le coût de transmission de l'information supplémentaire, pour des encodages à fort pas de quantification ou pour certaines applications qui nécessitent un nombre d'itération plus important.

Pour remédier à ce type de problématique, il est classiquement envisagé de calculer la valeur des seuils des algorithmes, à partir d'un voisinage connu de pixels. Ainsi, pour chaque nouveau bloc traité, cette valeur de seuil est adaptée en fonction de la nature du signal local. On peut envisager différentes méthodes : calculer l'énergie du signal voisin, se baser sur un modèle faisant intervenir la variance du signal ou même encore, générer un modèle d'évolution du seuil en fonction du pas de quantification fixé.

Au cours de nos travaux, nous n'avons cependant pas fait de recherche approfondie concernant cette problématique. Nous avons pris le parti d'évaluer les diverses techniques présentées pour leur capacité à restituer une prédiction de bonne qualité, notamment concernant la restitution de motifs bi-dimensionnels.

## 4.8 Conclusion

Nous avons présenté dans ce chapitre une nouvelle méthode de prédiction basée sur les représentations parcimonieuses, dans le contexte très contraint d'un encodeur vidéo. Les gains en qualité et débit obtenus pour la prédiction intra image d'un encodeur H.264 / AVC sont significatifs. Il pourrait être intéressant de poursuivre les travaux pour éliminer ou diminuer le surcoût de codage lié à l'information supplémentaire transmise. Nous avons également présenté des applications dans un esprit de raffinement de la prédiction inter-images. Tant pour unifier des informations issues d'images distinctes que pour distinguer le signal utile du signal considéré comme du bruit, dans le cas de l'application pour le dé-*crossfading*. Nous nous sommes également appliqués à mettre en place la prédiction parcimonieuse dans un contexte de type H.264 / SVC, pour améliorer la prédiction spatiale inter-couches. Les contraintes rencontrées ont été directement liées au surcoût engendré par le codage du nombre d'itérations, particulièrement pénalisant sur les petits blocs et lors d'un processus de raffinement. Il conviendrait d'éliminer cette contrainte par une recherche de seuil adaptative, indépendante de la source.

# Chapitre 5

# Déconvolution spectrale pour la prédiction

Ce chapitre présente une technique d'extrapolation de signal basée sur une analyse et un traitement spectral. L'algorithme au coeur de cette méthode, développé par André Kaup de l'université d'Erlangen et Til Aach de l'université d'Aachen [KA98], a pour fondement une *déconvolution* entre deux spectres fréquentiels. Le principe de la technique est, comme dans le cas général de l'extrapolation parcimonieuse, d'étendre un signal voisin par le biais de fonctions de base connues. La mise en oeuvre est cependant différente car tous les traitements s'effectuent dans le domaine fréquentiel. La philosophie générale consiste à sélectionner l'information utile dans le spectre fréquentiel des données d'observation et à retirer, par déconvolution, le parasitage fréquentiel lié aux pixels inconnus de la zone à extrapoler. Nous présentons dans les sections suivantes le détail de la méthode, ainsi que l'application que nous avons faite de cette technique dans le cadre de la prédiction d'images d'un encodeur vidéo. Nous nous sommes également attachés à faire le parallèle théorique entre cette technique de déconvolution et l'algorithme du *Matching Pursuit*.

## 5.1 Approche spectrale

### 5.1.1 Modélisation du problème

Notons $f$ une image qui a subi une détérioration. On suppose que cette détérioration, notée $w$ correspond à la perte d'un bloc dans une image [1]. La zone du voisinage n'ayant pas subi la détérioration est notée $A$, l'ensemble des pixels correspondant au bloc perdu est noté $B$ et $L = A \cup B$ (cf figure 5.1). On modélise cette image $f$ comme étant le produit de deux fonctions : $g$ et $w$ où $g$ correspond à l'image sans la détérioration. Il s'agit du signal inconnu que l'on souhaite retrouver. La fonction $w$ est par définition un signal porte, ainsi définie :

$$\begin{array}{rccl} w: & \mathbb{R}\times\mathbb{R} & \rightarrow & \mathbb{R}\times\mathbb{R} \\ & (m,n) & \mapsto & \begin{cases} 1 & \text{si} \quad (m,n)\in A \\ 0 & \text{si} \quad (m,n)\in B \end{cases} \end{array} \tag{5.1}$$

[1] La forme carrée de la détérioration n'est pas une limite en soi. La seule contrainte pour le cas présent est de connaître sa forme. Lorsque la détérioration, ou dans le cas plus général d'une perturbation convolutive, est inconnue, on parle de déconvolution aveugle.

L'image $g$ peut se représenter comme une combinaison linéaire pondérée d'un certain nombre de fonctions de base bi-dimensionnelles, choisies parmi un ensemble $K$ :

$$g = \sum_{k,l \in K} c_{k,l} \Phi_{k,l} \tag{5.2}$$

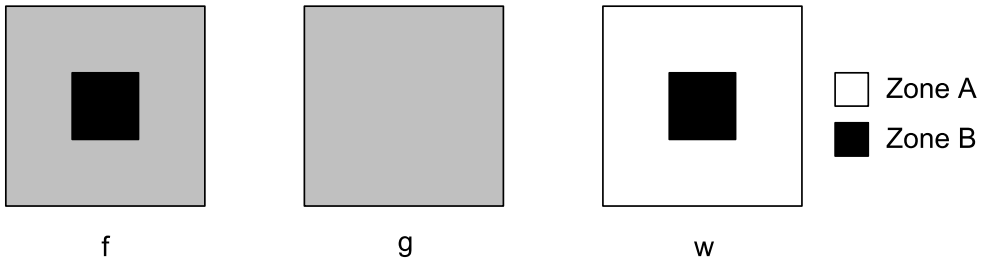



FIG. 5.1 – Images $f$, $g$ et $w$, la «fenêtre»

L'enjeu consiste à retrouver l'image $g$, et plus précisément, la valeur des coefficients $c_{k,l}$, connaissant l'image $f$, le masque $w$ et la base de projection $Z = \left\{\phi_{k,l} \quad \text{avec} \quad k,l \in K\right\}$. Les fonctions de base choisies sont les fonctions issues de la transformée de Fourier discrète, dont nous rappelons l'expression ci-dessous :

$$\phi_{k,l} = e^{2i\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)}$$

D'après la relation 5.2, cela revient à considérer la transformée de Fourier inverse de $g$. Reprenons les hypothèses de départ : on modélise $f$ comme étant le produit de deux fonctions $g$ et $w$. Si on calcule la transformée de Fourier de cette expression, le produit simple devient un produit de convolution :

$$f = g.w \quad \Longrightarrow \quad F = G * W$$

Le problème qui nous occupe alors est le suivant : connaissant le spectre de Fourier $F$ de l'image d'entrée, le spectre $W$ de la détérioration, on recherche le spectre $G$ sachant que la relation qui le lie à $W$ est une convolution.

### 5.1.2 Principe de la déconvolution

L'algorithme de déconvolution fréquentielle présenté consiste à sélectionner pas à pas au sein du spectre $F$, d'après les notations introduites dans la section précédente, les fréquences les plus représentatives du signal image et d'en extraire les données parasites introduites par la convolution avec le spectre $W$.

Pour bien comprendre sur quel principe repose l'algorithme, gardons à l'esprit deux points clés. Le premier concerne l'effet de la transformée de Fourier sur un signal. Cette transformation permet d'obtenir une décomposition fréquentielle des données. L'effet est le même que celui obtenu en physique par diffraction de Fraunhofer : toutes les fréquences sont dissociées. Notre but est de sélectionner celles qui sont prédominantes.

Le deuxième point théorique concerne l'impact du fenêtrage du signal. Par définition, l'image $g$ est fenêtrée par la porte $w$. Dans le domaine spectral, cela revient à convoluer le spectre du signal par un sinus cardinal (figure 5.2), le spectre $W$. Les lobes secondaires du sinus cardinal sont responsables d'un étalement des fréquences. C'est un problème classique dans le domaine du

traitement du signal qui peut être résolu en appliquant des fenêtres dont le spectre ne présente pas de lobes secondaires (fenêtres de Hamming, de Blackman ou encore de Hann).

On réduit ainsi les fuites spectrales, *i.e.* l'effet de brouillage ou encore de *blur* qui disperse anormalement les fréquences. Dans le cadre qui nous occupe ici, on ne peut pas, *a priori*, éviter la convolution par le sinus cardinal. Il va donc falloir corriger l'étalement parasite introduit par cette convolution. Cette correction correspond à la déconvolution de $G$ et $W$.

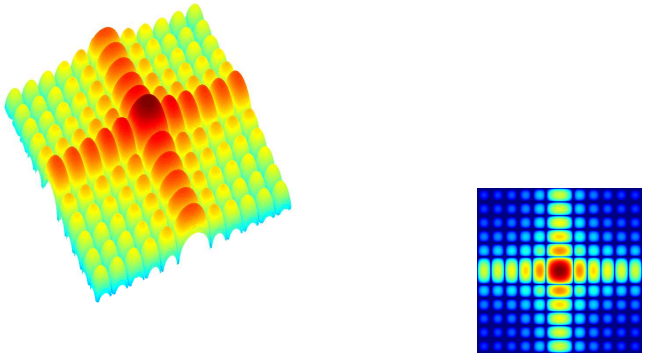


FIG. 5.2 – Sinus cardinal : représentation 3D et 2D

### 5.1.3 Descriptif de l'algorithme

La déconvolution fréquentielle présentée recherche itérativement la valeur des coefficients $c_{k,l}$ de l'expression 5.2, en mettant à jour le résidu $r$, erreur résiduelle entre l'image de départ, $f$, et le modèle $g$ qui se construit pas à pas. Ce qui est pertinent est d'évaluer l'erreur faite sur les pixels connus de la zone $A$. Donc on cherche à minimiser l'erreur fenêtrée suivante : $r_w = (f - g).w$. Il ne reste plus qu'à transposer cette analyse dans le domaine de Fourier, l'équivalence des représentations spatiales et fréquentielles étant assurée par le théorème de Parseval.

Les données connues sont le spectre $F$ bruité par le sinus cardinal, $W$, connu lui aussi. A l'initialisation, nous considérons un spectre $G$ dont tous les coefficients sont nuls : nous allons construire ce spectre dans le domaine fréquentiel en s'appuyant des données contenues dans $F$. Rappelons que l'on utilise toujours la même idée de base de l'extrapolation d'image présentée dans le chapitre précédent : on se base sur la connaissance d'un voisinage connu de pixels pour extrapoler de nouvelles données, en supposant que le signal est localement stationnaire. Afin de contourner les problèmes de stationnarité, les inventeurs de l'algorithme proposent d'utiliser une fenêtre différente de celle définie par 5.1. Ils utilisent de préférence une fonction de type gaussien qui va donner plus de poids aux pixels limitrophes de la zone $B$, comme présenté à la figure 5.3.

Par ailleurs, comme $G$ est à l'initialisation un spectre nul, le résidu transformé, $R$ est égal à $F$. Voici décrites ci-dessous les étapes de l'algorithme en régime permanent, pour un pas d'itération $\nu$ quelconque.

* La première étape consiste à repérer au sein du spectre $R$, le coefficient qui maximise la décroissance de l'énergie de l'erreur sur les pixels de la zone $A$, notée $\Delta E_A^{(\nu)}$ :

$$\Delta E_A^{(\nu)} = \frac{R_w^{(\nu)}(k,l)^2}{W(0,0)}$$

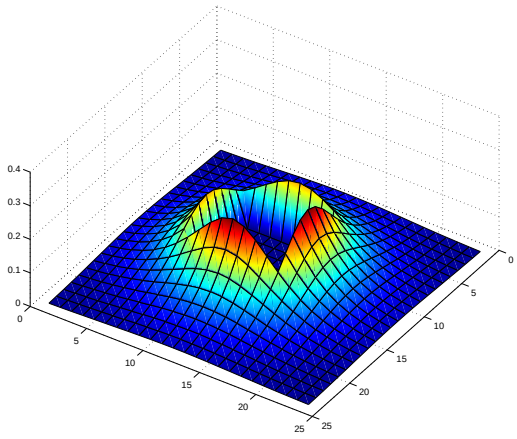



FIG. 5.3 – Fenêtre pondérée

où $W(0,0)$ est la valeur moyenne du sinus cardinal. On recherche donc le couple fréquentiel $(u,v)$ tel que :

$$(u,v) = arg \max_{k,l} \Delta E_A^{(\nu)}$$

* Pour cette position spatiale retenue (dans le spectre fréquentiel), *i.e.* $(u,v)$, la deuxième étape consiste à calculer le coefficient $c_{u,v}$ qui prendra part au modèle qui est en train d'être construit. Comme l'approche se base sur l'évaluation de la décroissance de l'énergie de l'erreur, on calcule plus exactement la valeur de l'accroissement du coefficient, notée $\Delta c$ :

$$\Delta c = MN.\frac{R_w^{(\nu)}(k,l)}{W(0,0)}$$

avec $M \times N$ la taille du spectre. Le coefficient est alors mis à jour grâce à cette valeur, de la manière suivante :

$$c_{u,v}^{(\nu+1)} = c_{u,v}^{(\nu)} + \Delta c$$

Le coefficient est ensuite stocké dans le spectre $G$ en position $(u,v)$.

* La dernière étape consiste à mettre à jour l'erreur résiduelle selon la relation ci-dessous :

$$R_w^{(\nu+1)}(k,l) = R_w^{(\nu)}(k,l) - \frac{1}{MN}\Delta c W(k-u, l-v)$$

Cette relation signifie que l'on vient retirer, en position $(u,v)$, la contribution de la fenêtre $W$, pondérée de la valeur de l'accroissement du coefficient trouvé. C'est ici que s'opère concrètement la déconvolution.

L'algorithme itère jusqu'à ce que la valeur de l'énergie de diminution de l'erreur devienne inférieure à un seuil qui a été préalablement fixé. On récupère ensuite l'image $g$ par une transformée de Fourier inverse du spectre $G$ qui vient d'être construit. Comme les fonctions de base de la transformée sont définies sur tout l'ensemble $L$, et non uniquement la zone $A$, on extrapole ainsi les données connues $A$ au sein de la zone $B$.

## 5.2 Analogie avec les représentations parcimonieuses

Dans cette section, nous montrons l'équivalence entre l'algorithme de déconvolution fréquentielle et celui du *Matching Pursuit*, dans le cas 1D. Nous exposerons tout d'abord une formulation du MP basée sur l'évolution de la corrélation et non plus sur celle du résidu (section 5.2.1). Puis, nous détaillerons l'algorithme dans le contexte d'une base de Fourier (sections 5.2.2 et 5.2.3) pour enfin présenter l'analogie avec la déconvolution (sections 5.2.4 et 5.2.5).

### 5.2.1 Réécriture de l'algorithme du MP

Soit $A$ une matrice de dimension $n \times m$ avec $m >> n$, $a_j$ les colonnes de $A$ de norme 1 et $y$ un vecteur de dimension $n$. Via une approche itérative, l'algorithme du MP permet d'obtenir une représentation parcimonieuse du signal $y$, en formant une combinaison linéaire d'un faible nombre d'atomes $a_j$ de norme euclidienne unitaire, les colonnes du dictionnaire $A$ (cf section 3.3.1.1).

A chaque itération, l'algorithme du MP cherche l'atome, issu de la base redondante, qui est le plus corrélé avec le résidu courant $r_{k-1}$ :

$$j_k = \arg\max_j \left| a_j^T r_{k-1} \right|$$

On retire ensuite au vecteur $r_{k-1}$ une version pondérée du vecteur $a_{j_k}$. Le poids qui minimise la norme du nouveau résidu $r_k$

$$x_k = arg\min_x \left\| r_{k-1} - a_{j_k} x \right\|_2^2$$

est précisément $x_k = a_k^T r_{k-1}$, à cause de la normalisation des $a_j$. Ce qui conduit à l'expression suivante pour $r_k$ :

$$r_k = \left( I - a_{j_k} a_{j_k}^T \right) r_{k-1} \tag{5.3}$$

A la première itération, on démarre avec $r_0 = y$ et on s'arrête, par exemple, lorsque la norme euclidienne du résidu est inférieure à un seuil, préalablement fixé. Au lieu de mettre l'accent sur les résidus, on peut le mettre sur les corrélations, $C_k$. En posant $C_0 = A^T y$ et $W^a = A^T A$, un pas de l'algorithme MP devient alors :

* $j_k = arg\max_j |C_{k-1}|$
* $C_k = C_{k-1} - x_k W^a_{j_k} = C_{k-1} - C_{k-1}(j_k) W^a_{j_k}$
* $x(j_k) = x(j_k) + x_k$

où $W^a_j$ correspond à la $j^{\text{ième}}$ colonne de $W^a$ et avec un critère d'arrêt sur $\|C_k\|_\infty$, par exemple.

### 5.2.2 Introduction des notations liées aux transformées de Fourier

Pour préparer les paragraphes suivants, nous introduisons quelques notations et nous présentons une écriture matricielle des transformées de Fourier discrètes. Nous notons $F$ la matrice de Fourier de dimension $m \times m$, dont la composante $(p, q)$ est $F_{p,q} = \exp(-2\, i\pi(p-1)(q-1)/m)$.

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

$F$ est une matrice complexe symétrique ($F = F^T$) qui vérifie $F\bar{F} = mI_m$, où $\bar{F}$ désigne la matrice conjuguée de $F$. La transformée de Fourier d'un signal $y$ est alors $Y = Fy$ et la transformée inverse $y = (1/m)\bar{F}Y$.

Pour $w$ et $x$, deux vecteurs réels de dimension $m$ et $y = w \cdot x$, le produit terme à terme de ces deux vecteurs, les propriétés suivantes sont alors satisfaites :

$$Y = Fy, \qquad X = Fx, \qquad W = Fw, \qquad Y = W * X,$$

où $*$ représente un produit de convolution circulaire des deux vecteurs $W$ et $X$. On peut également construire la matrice circulante $W_c$ ($\in \mathbb{C}^{m \times m}$), dont la première colonne est $W$. On peut alors récrire le produit de convolution comme le produit simple de $X$ par cette matrice circulante :

$$Y = W * X \quad \rightarrow \quad Y = W_c X.$$

### 5.2.3 Algorithme du MP dans le cas de la base de Fourier

Soit le vecteur réel $y \in \mathbb{R}^n$ pour lequel nous souhaitons obtenir une décomposition parcimonieuse dans une base redondante de Fourier *réelle*. On peut, par exemple, prendre comme colonne de la matrice $A$ des sinusoïdes réelles en $m$ fréquences équidistantes $f_k = k/2m,\ k \in (0, m-1)$. Mais on peut également prendre comme base la matrice constituée des $n$ premières colonnes de la matrice complexe $\bar{F}$ définie plus haut, que nous notons $\bar{F}_1$. Les colonnes sont alors des "cisoïdes" à $m$ fréquences équidistantes $f_k = k/m,\ k \in (0, m-1)$. Elles ont toutes la même norme $\sqrt{n}$ et il faudra en tenir compte dans l'algorithme et normaliser les corrélations. Nous allons utiliser les colonnes de $\bar{F}_1$ comme dictionnaire dans l'algorithme du MP, version corrélation.

A l'initialisation, on calcule $C_0 = \bar{F}_1^H y = F_1^T y$ la corrélation initiale, il s'agit d'un vecteur complexe de dimension $m$ avec une symétrie hermitienne. On calcule également $W^1 = F_1^T \bar{F}_1$. L'algorithme procède de la même manière que ci-dessus à ceci près que l'on souhaite représenter le signal réel $y$ à partir d'une base redondante complexe.

Au pas $k$, on recherche la composante dans $C_{k-1}$ dont le module est le plus élevé. Du fait de la symétrie hermitienne de $C$, le maximum est obtenu en deux positions symétriques : $j_k$ et $j_{m+1-j_k}$. Les pondérations associées qui annulent ces deux maxima simultanément sont les valeurs normalisées des corrélations. On obtient alors :

$$C_k = C_{k-1} - \left( \frac{C_{k-1}(j_k)}{n} W^1_{j_k} + \frac{C_{k-1}(m+1-j_k)}{n} W^1_{m+1-j_k} \right)$$

et :

$$x(j_k) = x(j_k) + \frac{C_{k-1}(j_k)}{n} \quad , \quad x(m+1-j_k) = x(m+1-j_k) + \frac{C_{k-1}(m+1-j_k)}{n}$$

avec $W^1(j)$ la colonne $j$ de la matrice $W^1$. Par construction, cet algorithme donne une reconstruction réelle de $y$, $\hat{y} = \bar{F}_1 x$, comme souhaité. Du fait de la symétrie hermitienne, on peut se contenter de travailler avec la moitié des fonctions de base et ainsi gagner en temps de calcul.

### 5.2.4 Analogie

La procédure présentée à la section précédente, le *Matching Pursuit* version corrélation sur une base complexe de Fourier est identique à celle proposée par A. Kaup et T. Aach. Pour présenter l'analogie, nous allons reprendre le descriptif de la déconvolution présentée en sections 5.1.2 et 5.1.3 sous un angle mono-dimensionnel, avec des notations vectorielles et matricielles.

Notons $Y_z$ la transformée de Fourier de $y_z$, un vecteur réel de dimension $m$, ayant $m - n$ composantes nulles. On peut toujours supposer que ce vecteur $y_z$ a ses $n$ premières composantes égales au signal $y$, défini dans la section 5.2.3 précédente, les autres composantes étant égales à zéro. Le vecteur $y_z$ correspond au produit simple de deux vecteurs, $x$ et $w$, de dimension $m$. La fenêtre $w$, a ses $n$ premières composantes égales à 1 et les autres sont égales à zéro.

Les transformées de Fourier des différents signaux sont obtenues à partir de la matrice de Fourier $F$ : $Y_z = Fy_z$, $X = Fx$ et $W = Fw$. Alors, d'après les propriétés de la transformée de Fourier, le produit terme à terme se transforme en produit de convolution :

$$y_z = wx \qquad \rightarrow \qquad Y_z = W * X$$

où $*$ désigne un produit de convolution. Dans le paragraphe 5.2.2, nous avons rappelé que ce produit de convolution peut s'écrire sous le produit de la matrice circulante $W_c$, dont la première colonne est $W$ et du vecteur $X$ :

$$Y_z = W_c X$$

L'algorithme de la déconvolution de Kaupp propose d'obtenir une représentation parcimonieuse de $X$ et donc de $x$ sa transformée de Fourier inverse. Comme $x$ est un signal réel, le signal $X$ reconstruit doit présenter une symétrie hermitienne.

L'algorithme tend à minimiser itérativement le signal d'erreur $R_k$. A l'initialisation, $R_0 = Y_z = Fy_z = F^T y_z = F_1^T y$ et donc $R_0 = C_0$ avec $C_0$ la corrélation initiale introduite à la section 5.2.3. De plus, $X_0 = 0$. Les deux algorithmes sont donc identiques si $W^1 = W_c$.

Grâce à la symétrie hermitienne, la procédure ne prend en compte que la moitié des fonctions de base (les coefficients pouvant être retrouvés en calculant le conjugué) et des vecteurs. Cependant nous ne présentons pas ici les modifications tenant compte de cette symétrie, pour simplifier l'explication et mieux faire apparaître l'analogie.

Au pas $k$, l'algorithme recherche dans $R_{k-1}$ la composante complexe de plus fort module. On observe deux maxima en, disons, $j_k$ et $m+1-j_k$. Puis, on recherche la valeur des deux coefficients non nuls à positionner en $\delta X_k$ aux positions $j_k$ et $m + 1 - j_k$, de telle sorte que les coefficients $j_k$ et $m + 1 - j_k$ soient nuls dans $R_k = R_{k-1} - W_c \delta X_k$, où $\delta X_k$ désigne l'accroissement du coefficient. On met ensuite à jour $X_k = X_{k-1} + \delta X_k$.

Pour établir l'analogie entre les deux approches, il reste à montrer que ces mises à jour sont identiques à celles présentes dans le paragraphe précédent. Le passage de $R_{k-1}$ à $R_k$ correspond

précisément à celui de $C_{k-1}$ à $C_k$ si on arrive à établir que $W^1 = W_c$ où $W_c$ est la matrice circulante dont la première colonne est $W$. Par définition, on a :

$$\begin{aligned} W^1 &= F_1^T \bar{F}_1 \\ W &= Fw = F_1^T \mathbf{1} \end{aligned}$$

où $\mathbf{1}$ est un vecteur de dimension $n$ dont tous les coefficients sont égaux à 1. Ccomme la prmière colonne de $\bar{F}_1$ vaut $\mathbf{1}$, ceci prouve que les premières colonnes de $W^1$ et $W_c$ sont identiques. Il reste donc à prouver que $W^1$ est une matrice circulante et donc à démontrer que $W^1(k,l) = W^1(k+1, l+1)$. On a :

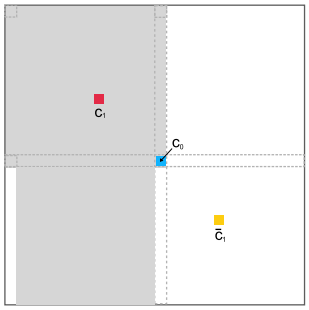
$$\begin{aligned} W^1(k,l) &= \sum_{p=1}^{n} F_1^T(k,p)\bar{F}_1(p,l) \\ &= \sum_{p=1}^{n} \mathrm{e}^{-\frac{2i\pi}{m}(k-1)(p-1)} \mathrm{e}^{\frac{2i\pi}{m}(p-1)(l-1)} \end{aligned}$$

Montrons l'égalité $W^1(k,l) = W^1(k+1, l+1)$ :

$$\begin{aligned} W^1(k+1,l+1) &= \sum_{p=1}^{n} \mathrm{e}^{-\frac{2i\pi}{m}k(p-1)} \mathrm{e}^{\frac{2i\pi}{m}(p-1)l} \\ &= \sum_{p=1}^{n} \left[\mathrm{e}^{\frac{2i\pi}{m}(p-1)} \mathrm{e}^{-\frac{2i\pi}{m}k(p-1)}\right]\left[\mathrm{e}^{-\frac{2i\pi}{m}(p-1)} \mathrm{e}^{\frac{2i\pi}{m}(p-1)l}\right] \\ &= W^1(k,l) \end{aligned}$$

### 5.2.5 Discussion dans le cas bi-dimensionnel

Reprenons les notations introduites en section 5.1.3. On souhaite extrapoler les données manquantes d'une image $f$, que l'on modélise comme le produit de l'image $g$, inconnue, et la fenêtre $w$. Le spectre $F$ est doté d'une symétrie hermitienne si bien qu'il suffit de rechercher les fréquences dans un demi-plan, comme l'indique la figure 5.4.




$$\begin{aligned} F(k,l) &= \sum_{m,n} f(m,n)\mathrm{e}^{-2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)} \\ c_0 &= \sum_{m,n} f(m,n) \\ c_1 &= F(u,v) = a + ib \\ \bar{c}_1 &= \bar{F}(N+1-u, N+1-v) = a - ib \end{aligned}$$

FIG. 5.4 – Spectre de Fourier bi-dimensionnel

Pour reconstruire un signal qui soit réel, il est nécessaire de réécrire le modèle $g$ (eq. 5.2), de la manière suivante :

$$g(m,n) = \frac{1}{2N^2} \sum_{m,n} \left[ F(k) \mathrm{e}^{2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)} + \bar{F}(k) \mathrm{e}^{-2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)} \right]$$

Lorsque l'on recherche la valeur du coefficient $c_1$ en position $(u, v)$, on travaille sur la fonction de base bi-dimensionnelle : $\mathrm{e}^{2i\pi\left(\frac{mu}{N}+\frac{nv}{N}\right)}$.

Dans le cadre du MP, pour construire le dictionnaire $A \in \mathbb{R}^{m \times m}$ basé sur une transformée de Fourier réelle, nous avons effectué le produit de Kronecker $A = A_{1D} \otimes A_{1D}$ avec $A_{1D}$ la matrice de dimension $\sqrt{m} \times \sqrt{m}$ dont les $\sqrt{m}/2 - 1$ premières colonnes correspondent à la partie réelle (les cosinus), et les restantes correspondent à la partie imaginaire (les sinus) de la fonction de base mono-dimensionelle $\mathrm{e}^{2i\pi\left(\frac{mk}{N}\right)}$. Notons $z_{1D} = F(k)\mathrm{e}^{2i\pi\left(\frac{mk}{N}\right)}$ où $F(k) = a + ib$.

$$z_{1D} + \bar{z}_{1D} = 2\Re(z_{1D}) = 2a \cos\left[2i\pi\left(\frac{mk}{N}\right)\right] - 2b \sin\left[2i\pi\left(\frac{mk}{N}\right)\right]$$

En notation abrégée, $z_{1D} + \bar{z_{1D}} = 2ac_{m,k} - 2bs_{m,k}$. Les cosinus $c_{m,k}$ et sinus $s_{m,k}$ constituent les atomes de $A_{1D}$. En bi-dimensionnel, notons $z_{2D} = F(k,l)\mathrm{e}^{2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)}$. De la même manière :

$$z_{2D} + \bar{z}_{2D} = 2\Re(z_{2D}) = 2a \cos\left[2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)\right] - 2b \sin\left[2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)\right]$$

Soit :

$$z_{2D} + \bar{z}_{2D} = 2a\left[c_{m,k}c_{n,l} + s_{m,k}s_{n,l}\right] - 2b\left[s_{m,k}c_{n,l} + c_{m,k}s_{n,l}\right]$$

Les atomes bi-dimensionnels du dictionnaire $A$ dans le contexte de Fourier sont donc les produits de fonctions $c_{m,k}c_{n,l}$, $s_{m,k}s_{n,l}$, $s_{m,k}c_{n,l}$ et $c_{m,k}s_{n,l}$. Lors d'une itération du MP, on sélectionne un de ces atomes qui ne correspond, en fait, qu'à $1/4$ de la fonction de base utilisée dans le cas de la déconvolution. En effet :

* la fonction de base utilisée en déconvolution est complexe : $\mathrm{e}^{2i\pi\left(\frac{mk}{N}+\frac{nl}{N}\right)}$. Avec la déconvolution, le signal est projeté en une seule fois sur cette base complexe. Le MP est de fait obligé de faire deux itérations pour récupérer la partie réelle et la partie imaginaire de cette exponentielle. Et rien ne le contraint à sélectionner la partie imaginaire lorsque que la partie réelle a été sélectionnée, et vice-versa ;

* de plus, le produit de Kronecker engendre des atomes bi-dimensionnels sous la forme de produit de plusieurs fonctions de cosinus et sinus mono-dimensionnelles. Si bien que pour avoir la fonction de base cosinus bi-dimensionnelle complète, cela nécessite également deux itérations.

## 5.3 Application à la prédiction

Nous avons mis en place l'algorithme dans une structure d'encodeur vidéo de type H.264 / AVC, afin d'évaluer ses performances en prédiction intra image.

### 5.3.1 Adaptation du fenêtrage

En prédiction, seul le voisinage causal est connu. Nous avons donc adapté la fenêtre afin de prendre en compte cette contrainte. La figure 5.5 ci-dessous illustre ces modifications, dans le cas d'un voisinage de quatre blocs connus.

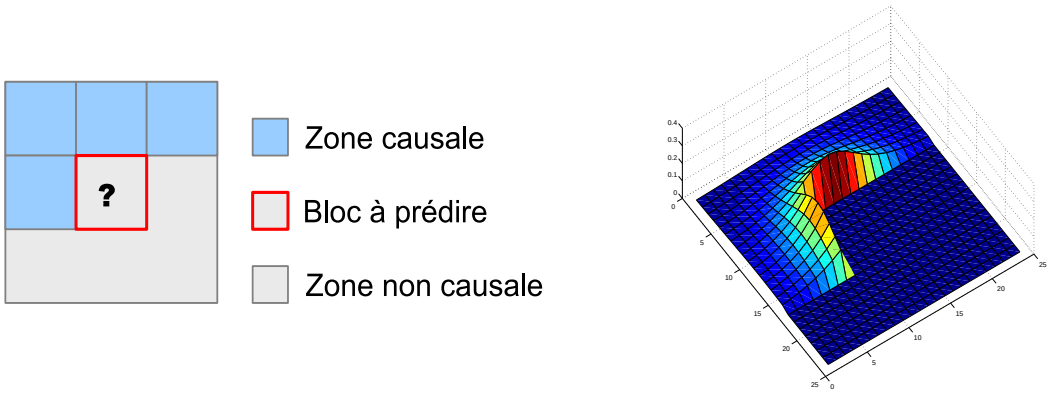



FIG. 5.5 – Fenêtre pondérée adaptée à la prédiction

### 5.3.2 Critère d'arrêt

La problématique liée à l'évaluation de la qualité de la prédiction reste identique à celle évoquée dans le cas de la prédiction parcimonieuse.

L'algorithme présenté s'arrête lorsque l'énergie de l'erreur résiduelle devient inférieure à un seuil fixé à l'avance. Il est très difficile de prendre une décision *a priori* sur la valeur de ce seuil. Idéalement, il faudrait adapter cette valeur en fonction du contenu du voisinage local, pour chaque nouveau voisinage. Nous avons choisi d'introduire une procédure d'arrêt supplémentaire, identique à celle présentée en section 4.2.4.1 pour les algorithmes de représentations parcimonieuses. On introduit une mesure basée sur l'erreur quadratique moyenne pour évaluer la qualité de la reconstruction du pas courant.

### 5.3.3 Résultats dans un encodeur de type H.264 / AVC

Nous avons implémenté l'algorithme au sein d'un encodeur H.264 / AVC, afin d'en tester les performances.

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... Déconvolution / MP
* Dictionnaire ..... *Néant* / DFT réelle
* *QP* ..... $21 - 26 - 30 - 35$
* Partition ..... $8 \times 8$
* Sélection des modes intra ..... SAD
* Seuil ..... 18 / 1
* Critère d'arrêt ..... EQM / EQM

Nous obtenons (tableau 5.1) des performances similaires en terme de qualité de prédiction et de débit. L'algorithme de déconvolution proposé présente néanmoins l'avantage de converger plus rapidement vers une solution répondant au critère établi. La base de fonctions utilisées est plus riche car c'est une base complexe. Si bien que l'algorithme nécessite de moins d'itérations que le MP pour générer une prédiction comparable.

On peut d'ailleurs remarquer d'après le tableau de résultats 5.1, que le surcoût en déconvolution est inférieur à celui du MP. Comme nous l'avons explicité en section 5.2.5, cette différence est liée à la richesse de la base de Fourier complexe utilisée dans l'algorithme de déconvolution spectrale, ce qui conduit ainsi à réduire le nombre d'itérations nécessaires. La prédiction parcimonieuse basée sur une DFT réelle nécessite quant à elle de plus de fonctions de base pour modéliser le signal.

| | Sans prise en compte du surcoût | | Avec prise en compte du surcoût | |
|---|---|---|---|---|
| | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| MP DFT réelle | + 0.43 | - 5.67 | + 0.30 | - 3.97 |
| Déconvolution | + 0.45 | - 5.87 | + 0.34 | - 4.38 |

TAB. 5.1 – Résultats Bjontegaard de la prédiction intra $8 \times 8$ basée déconvolution

## 5.4 Conclusion

Nous avons présenté dans ce chapitre l'équivalence dans le cas 1D de l'algorithme du *Matching Pursuit* et celui proposé par A. Kaup et T. Aach, basé sur une déconvolution fréquentielle. Nous pouvons cependant souligner une différence entre ces deux approches notamment dans le cas 2D. Le MP construit la prédiction à partir des données spatiales disponibles, sans avoir besoin de modéliser la zone à extrapoler. Dans la seconde approche, on modélise les données manquantes dans le domaine spectral puis on débruite les fréquences pertinentes (pour la reconstruction) en tenant précisément compte des données manquantes. Dans le cadre des représentations parcimonieuses, on reconstruit le signal à partir de briques élémentaires, les atomes, en se basant uniquement sur le voisinage connu. Dans le cas de la déconvolution, on commence par décomposer le signal, incluant la zone inconnue, sur les fonctions de base puis on retire le signal assimilé au bruit. La méthode demeure relativement complexe du fait de la convolution dans le domaine de Fourier.

# Chapitre 6

# Dictionnaires

Les dictionnaires représentent le coeur des représentations parcimonieuses. La qualité de la représentation obtenue, tant en terme de parcimonie que de similarité aux données traitées, dépend directement du choix en nombre et en variété des fonctions de base. En prédiction d'images, nous avons la contrainte de conserver un dictionnaire riche en fonctions à support étendu, afin d'être à même d'extrapoler une texture sur un domaine plus vaste que celui pour lequel elle est définie. Jusqu'à présent, nous avons présenté des résultats en prédiction basés sur l'usage de dictionnaire formés de fonctions harmoniques, telles que celles issues de la transformée en cosinus discrète ou de la transformée de Fourier discrète. Les résultats sont satisfaisants car nous avons observé une reconstruction acceptable des zones texturées au sein d'une image, ce qui manquait à la prédiction intra de la norme H.264 / AVC, basée sur des modes directionnels. Cependant, les structures rectilignes de l'image sont, quant à elles, particulièrement bien reconstruites à l'aide de ces modes directionnels. Notre prédiction parcimonieuse ne proposant pas, pour ces situations, une prédiction de meilleure qualité, la prédiction de la norme est quasiment systématiquement choisie. Nous avons ainsi voulu enrichir notre prédiction parcimonieuse, en proposant des solutions qui introduisent une notion d'anisotropie dans le traitement des données ou dans la nature des fonctions de base.

## 6.1 Adaptabilité des fonctions de base

### 6.1.1 Scan directionnel des observations

A défaut de créer un dictionnaire exhaustif, contenant le plus de directions possibles, nous nous sommes intéressés au cas connexe qui consiste à réorienter les données d'observations le long de directions privilégiées. Si on parcourt les pixels le long d'une trajectoire correspondant au motif contenu dans l'image, on a de forte chance de créer un vecteur d'observation $y$ au sein duquel de nombreux pixels seront très corrélés entre eux (figure 6.1).

L'idée est ainsi de n'avoir besoin que des premiers atomes de la DCT, par exemple, en première ligne et colonne d'après la figure 6.1. On se soustrait indirectement à la contrainte d'avoir des atomes très variés et multi-directionnels. Cependant, nous sommes bien évidemment limités par le nombre d'orientations des parcours que nous définissons. Nous proposons d'utiliser six parcours directionnels (figure 6.2) : les parcours classiques horizontaux et verticaux ; un parcours diagonal et un autre anti-diagonal ; et deux autres parcours dont les directions forment un angle de $22.5°$ et un angle de $-22.5°$.

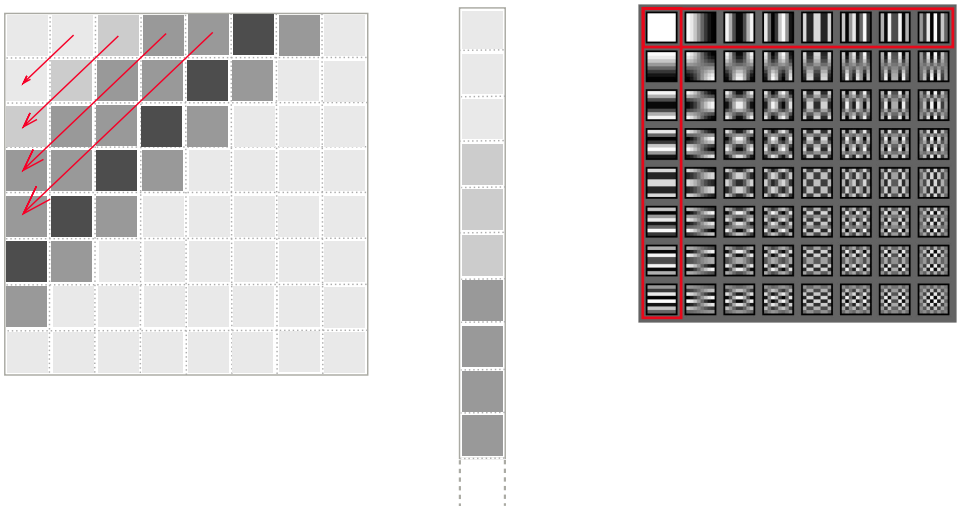


FIG. 6.1 – Parcours directionnel des observations

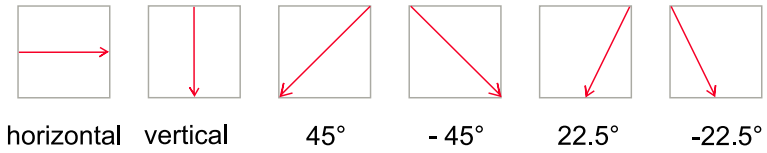



FIG. 6.2 – Parcours directionnels proposés

La sélection de la meilleure des six prédictions que nous proposons se fait ensuite par le biais d'un critère lagrangien. Nous substituons alors cette prédiction à un mode de la norme. Le choix final du meilleur mode de prédiction intra-image est ensuite obtenu de la même manière, *i.e.* par une optimisation débit / distorsion, comme indiqué en figure 6.3.

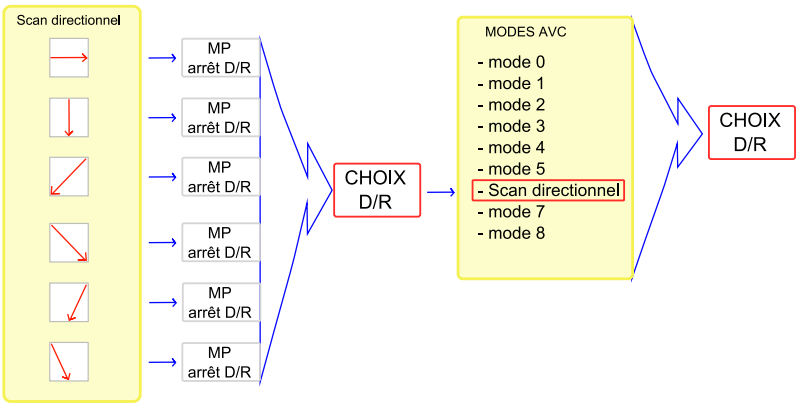



FIG. 6.3 – Sélection des modes directionnels

Compte tenu de ce choix supplémentaire, nous devons tenir compte du coût de signalisation à transmettre au décodeur, correspondant à l'indice du mode directionnel choisi comme meilleure prédiction parcimonieuse.

Le tableau 6.1 présente les résultats de cette expérimentation, en tenant compte des différents coût de signalisation : celui qui vient d'être évoqué, et également celui lié au critère d'arrêt

ajouté au sein de l'algorithme (ici on utilise un critère lagrangien pour sélectionner la meilleure représentation parcimonieuse ; cf section 4.2.4.2).

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... MP
* Dictionnaire ..... DCT
* *QP* ..... $21 - 26 - 30 - 35$
* Partition ..... $8 \times 8$
* Sélection des modes intra ..... Optimisation D / R
* Seuil ..... 1
* Critère d'arrêt ..... Optimisation D / R

Nous constatons que cette approche apporte une légère amélioration en terme de qualité et de gain en débit. Cependant, l'approche présentée étant pénalisée par deux surcoûts, nous n'obtenons pas d'amélioration en qualité et en débit, par rapport à la prédiction parcimonieuse, lorsque l'on prend en considération ces coûts de codage.

| | Sans surcoût | | Avec surcoût(s) | |
|---|---|---|---|---|
| Configuration | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| MP - arrêt $D + \lambda R$ - RDOPT =1 | + 0.57 | - 7.38 | + 0.37 | - 4.84 |
| MP - Scan directionnel - arrêt $D + \lambda R$ - RDOPT =1 | + 0.62 | - 8.10 | + 0.37 | - 4.71 |

TAB. 6.1 – Evaluation des performances obtenues avec les parcours directionnels

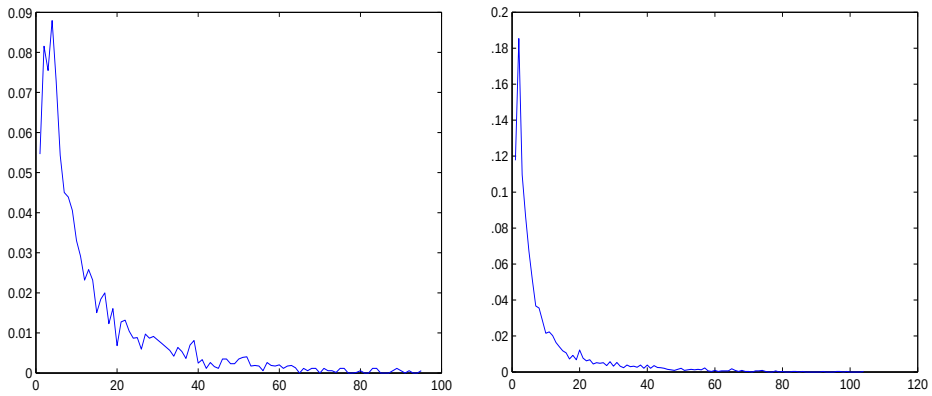



FIG. 6.4 – Histogrammes du nombre d'itérations pour $QP = 21$ du MP dans le cas d'une décision lagrangienne (à gauche) ; et du MP dans le cas des parcours directionnels, à droite

On constate d'après les courbes présentées en figure 6.4 que les deux approches, avec et sans parcours directionnels, ne présentent pas la même répartition des valeurs prise par les nombres d'itérations réalisées. On remarque que le nombre d'itérations étant très proche du nombre d'atomes retenus (il diffère seulement lorsqu'un même atome est sélectionné plusieurs fois), les représentations obtenues sont plus parcimonieuses dans le cas des parcours directionnels.

### 6.1.2 Réajustement de la phase spatiale des atomes

Il est certain que la richesse du dictionnaire conditionne la qualité de la reconstruction, tant par la diversité des fonctions de base que par la quantité. Nous nous sommes intéressés à des méthodes permettant d'éviter d'accroître la taille du dictionnaire tout en offrant un plus large choix d'atomes. La technique proposée consiste à appliquer aux fonctions de base un décalage spatial afin d'augmenter la corrélation entre les atomes et le signal à représenter. Ce décalage spatial correspond à une translation dans les deux dimensions spatiales (horizontale et verticale) dont les paramètres de transformation sont obtenus via la technique de corrélation de phase. Cette approche permet d'accroître virtuellement la redondance du dictionnaire en proposant des versions phasées des atomes existants.

#### 6.1.2.1 Introduction

La méthode de corrélation de phase est simple à mettre en oeuvre et est basée sur une propriété de décalage, intrinsèque à la transformée de Fourier. L'idée de base est liée au fait que pour deux images relativement proches, on constate que la puissance d'énergie dans le spectre de puissance croisé est concentrée en un pic localisé spatialement.

Soit $f(x, y)$ et $g(x, y)$ deux images de $\mathbb{R}^2$. On note $F_f(\omega_x, \omega_y)$ et $F_g(\omega_x, \omega_y)$ leur transformée de Fourier, respectivement :

$$TF\left[f(x, y)\right] = F_f(\omega_x, \omega_y)$$

$$TF\left[g(x, y)\right] = F_g(\omega_x, \omega_y)$$

Supposons que $g$ est très proche de l'image $f$ et correspond à une version translatée de $f$ :

$$g(x, y) = f(x + x_0, y + y_0)$$

D'après la propriété de translation de la transformée de Fourier, on a :

$$F_g(\omega_x, \omega_y) = F_f(\omega_x, \omega_y)\mathrm{e}^{i\left(\omega_x x_0 + \omega_y y_0\right)}$$

Soit, de manière équivalente :

$$\frac{F_g(\omega_x, \omega_y)\bar{F}_f(\omega_x, \omega_y)}{\left|F_g(\omega_x, \omega_y)\bar{F}_f(\omega_x, \omega_y)\right|} = \mathrm{e}^{i\left(\omega_x x_0 + \omega_y y_0\right)}$$

où $\bar{F}_f(\omega_x, \omega_y)$ est le complexe conjugué de $F_f(\omega_x, \omega_y)$. La partie gauche de l'équation ci-dessus correspond au spectre de puissance croisé. Il est ensuite aisé de déterminer $x_0$ et $y_0$ puisque la transformée de Fourier inverse de $\mathrm{e}^{j\left(\omega_x x_0 + \omega_y y_0\right)}$ correspond à la fonction de Dirac, centrée en position $(x_0, y_0)$ :

$$TF^{-1}\left(\frac{F_g(\omega_x, \omega_y)\bar{F}_f(\omega_x, \omega_y)}{\left|F_g(\omega_x, \omega_y)\bar{F}_f(\omega_x, \omega_y)\right|}\right) = \delta\left(x_0, y_0\right)$$

Dans le cas de transformées de Fourier discrètes de dimensions finies, la distribution de Dirac est remplacée par une impulsion unitaire en position $(x_0, y_0)$. Cette position indique le décalage spatial relatif entre les deux signaux.

#### 6.1.2.2 Corrélation de phase appliquée aux atomes

Nous cherchons à estimer le décalage à appliquer à l'atome sélectionné par une méthode de corrélation de phase. Soit $y_c \in \mathbb{R}^n$ le vecteur contenant les pixels du voisinage connus. Nous souhaitons trouver la représentation parcimonieuse $x \in \mathbb{R}^m$ en choisissant un faible nombre de fonctions de base $a_j$, contenues dans le dictionnaire $A \in \mathbb{R}^{n \times m}$.

La première étape au sein des algorithmes de représentations parcimonieuses que nous avons utilisés (cf section 3.3), consiste à sélectionner l'atome de plus forte corrélation avec le résidu courant. Par exemple, celui en position $j_k$ dans le dictionnaire. Seulement, il se peut que cette corrélation puisse être supérieure, pour une position spatiale de cet atome $a_{j_k}$, légèrement décalée. L'enjeu est de déterminer le déplacement bi-dimensionnel qui maximise la corrélation.

Remarque: *Idéalement, il faudrait calculer le déphasage de tous les atomes, avant la sélection de celui qui maximise la corrélation. Pour des raisons de simplification algorithmique, nous proposons d'ajuster la phase de l'atome sélectionné,* $a_{j_k}$

Notons $F_y$ et $F_{a_{j_k}}$ les transformées de Fourier, respectivement, de $y \in \mathbb{R}^m$, le vecteur non compacté (contenant les $n$ pixels connus du voisinage et $m - n$ zéros) et de l'atome $a_{j_k} \in \mathbb{R}^m$ dont aucune composante n'a été masquée. On calcule le spectre $R$ de puissance croisé normalisé suivant :

$$R = \frac{F_y \bar{F}_{a_{j_k}}}{\left| F_y \bar{F}_{a_{j_k}} \right|} \tag{6.1}$$

Le plan de corrélation $r$ est ensuite obtenu par transformée de Fourier inverse de $R$ donné par l'équation 6.1. La position du pic de corrélation observé dans $r$ nous indique le décalage pour lequel les deux signaux ont une corrélation maximale.

$$\left(\Delta_y^{pel}, \Delta_x^{pel}\right) = arg \max_{(x,y)} \{r\} \tag{6.2}$$

Une fois le décalage obtenu, il nous suffit de générer un nouvel atome, non pas par interpolation, mais par le calcul exacte de la fonction de base puisque l'expression théorique de la fonction est connue.

#### 6.1.2.3 Corrélation de phase sous-pixellique

La technique présentée ci-dessus ne permet d'obtenir qu'une précision au pixel entier de la position $(x_0, y_0)$ du pic de corrélation. Pour obtenir une précision sous-pixellique, il existe de nombreuses méthodes que les auteurs présentent dans ce panorama [TH86].

Nous présentons ici une technique très efficace qui consiste à faire correspondre une fonction de référence prédéfinie avec le plan de corrélation et plus spécifiquement au niveau du pic de corrélation. La technique est connue sous le nom de *curve fitting*. On recherche les paramètres de la fonction de référence, par exemple une parabole comme illustrée en figure 6.5, de telle sorte qu'elle passe par plusieurs points d'observation. Le maximum de la fonction donne la nouvelle valeur du maximum de corrélation, située à une position sous-pixellique en $x$ et en $y$.

Pour procéder au calcul de la position sous pixellique, nous avons choisi d'utiliser l'ensemble des huit voisins du pic, noté $m$, obtenu lors de la corrélation de phase, en position entière $\left\{\Delta_y^{pel}, \Delta_x^{pel}\right\}$. Notons $A$, $B$, $C$, $D$, $E$, $F$, $G$ et $H$, ses huits voisins comme l'illustre la figure 6.6.

La procédure consiste à estimer de manière séparable le déplacement horizontal en $x$ et celui, vertical, en $y$. Pour le déplacement horizontal, on somme les valeurs connues, ligne par ligne, des

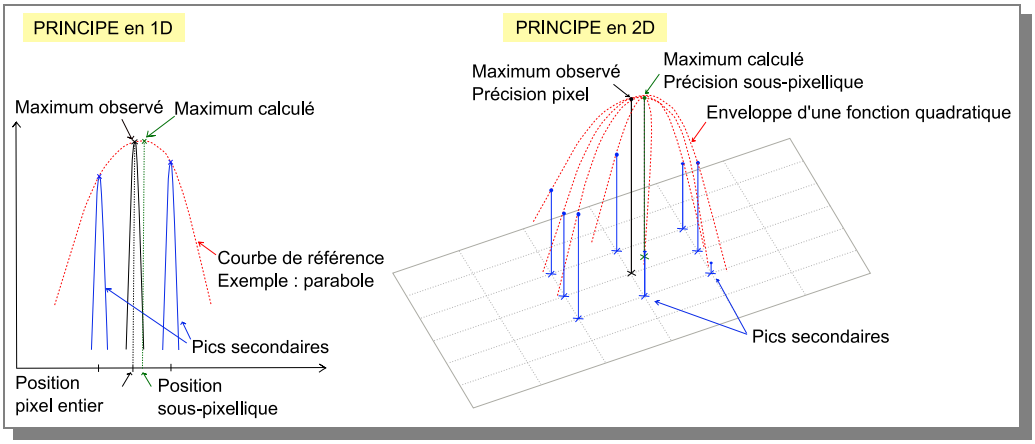



FIG. 6.5 – Ajustement sous-pixellique

$$\begin{matrix} A & B & C \\ D & m & E \\ F & G & H \end{matrix}$$

FIG. 6.6 – Pics voisins au pic maximal $m$ dans le plan de corrélation de phase

voisins du pic $m$. Notons $y_i^H$, les sommes correspondantes qui vont ensuite permettre de déterminer les paramètres $a$ et $b$ de l'équation de la parabole recherchée dont l'expression mathématique est $ax^2+bx+c$. Comme nous cherchons à déterminer la racine double de la parabole dont l'expression est $-b/2a$, les paramètres $a$ et $b$ suffisent. On calcule les sommes $y_i^H$ de la manière suivante :

$$\begin{aligned} y_1^H &= A+B+C \\ y_2^H &= D+m+E \\ y_3^H &= F+G+H \end{aligned}$$

Notons $x_1^H$, $x_2^H$ et $x^H3$, les abscisses correspondantes, dont les relations qui les lient sont les suivantes, dans le référentiel où le pic $m$ est à l'abscisse 0 :

$$x_1^H - x_2^H = -1 \tag{6.3}$$
$$x_3^H - x_2^H = 1 \tag{6.4}$$
$$(x_1^H)^2 = (x_2^H)^2 + 1 - 2x_2^H \tag{6.5}$$
$$(x_3^H)^2 = (x_2^H)^2 + 1 + 2x_2^H \tag{6.6}$$

Pour déterminer les paramètres $a$ et $b$, on doit résoudre le système d'équation suivant :

$$\begin{aligned} y_1^H &= a(x_1^H)^2 + bx_1^H + c \\ y_2^H &= a(x_2^H)^2 + bx_2^H + c \\ y_3^H &= a(x_3^H)^2 + bx_3^H + c \end{aligned}$$

En calculant la somme $S_a = \frac{y_1^H + y_3^H}{2} - y_2^H$, on obtient :

$$S_a = a \left( \frac{(x_1^H)^2 + (x_3^H)^2 - 2(x_2^H)^2}{2} \right) + b \frac{\left(x_1^H - x_2^H\right) + \left(x_3^H - x_2^H\right)}{2}$$

Compte tenu des relations 6.3 et 6.4, $S_a = a \frac{(x_1^H)^2 + (x_3^H)^2 - 2(x_2^H)^2}{2}$. De plus, d'après les relations 6.5 et 6.6, on a

$$(x_1^H)^2 + (x_3^H)^2 - 2(x_2^H)^2 = 2$$

Soit $S_a = a$. De manière similaire, en exploitant les relations précédemment évoquées, on obtient $b = \frac{y_3^H - y_1^H}{2}$. Les paramètres $a$ et $b$ sont donc calculés comme indiqué ci-dessous :

$$\begin{aligned} a &= \frac{y_1^H + y_3^H}{2} - y_2^H \\ b &= \frac{y_3^H - y_1^H}{2} \end{aligned}$$

Les paramètres $a$ et $b$ de la racine double ainsi obtenus permettent de connaître la valeur du déphasage sous pixellique horizontal par rapport à la position entière du pic $m$. Soit $\delta_x$ ce déphasage et $\delta_x = -b/a$. En procédant de manière identique mais en se basant sur la somme des valeurs en colonnes comme indiqué ci-dessous :

$$\begin{aligned} y_1^V &= A + D + F \\ y_2^V &= B + m + G \\ y_3^V &= C + E + H \end{aligned}$$

on en déduit le déplacement sous pixellique vertical $\delta_y$. Le déplacement sous pixellique global $\left\{\Delta_y, \Delta_x\right\}$, exprimé dans le référentiel du plan de corrélation est ainsi égale à :

$$\begin{aligned} \Delta_x &= \Delta_x^{pel} + \delta_x \\ \Delta_y &= \Delta_y^{pel} + \delta_y \end{aligned}$$

#### 6.1.2.4 Méthode de mise à jour des atomes du dictionnaire

*– Mise en oeuvre pour le Matching Pursuit –*

Une fois l'atome de plus forte corrélation sélectionné, nous appliquons le processus de corrélation de phase sous-pixellique pour connaître l'éventuel déphasage à appliquer à l'atome. Dans nos expérimentations, nous avons appliqué la technique de corrélation de phase au sous-pixel, présentée précédemment. Nous avons tenu compte des 8 valeurs voisines du maximum de corrélation obtenu au pixel entier.

Afin de consolider le processus, nous avons ajouté un test de vérification qui consiste à garder ce nouvel atome phasé, uniquement si sa corrélation avec le résidu courant est supérieure à la corrélation calculée avec l'atome initial, non phasé. Le schéma 6.7 indique les étapes liées à l'ajout de l'approche par corrélation de phase.

Par ailleurs, pour des raisons de contraintes de dimensions en puissance de deux, liées à l'utilisation de la FFT (*Fast Fourier Transform*), nous n'avons utilisé qu'un seul type de voisinage.

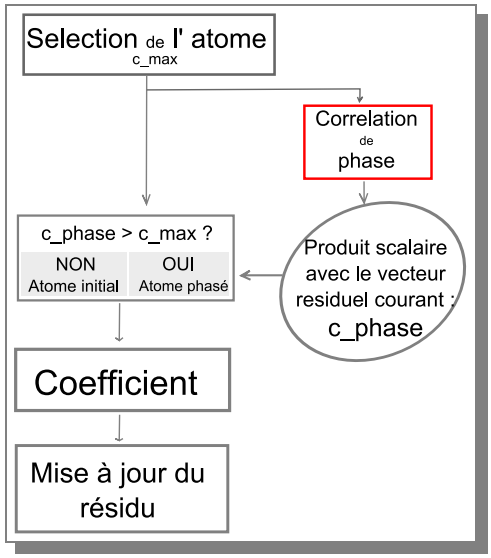



FIG. 6.7 – Application de la corrélation de phase à l'algorithme du *Matching Pursuit*

Si l'on se réfère à la description qui en est faite, en section 4.3.2, pour une prédiction intra $8 \times 8$, il peut exister trois types de voisinages, composés de trois, quatre ou cinq blocs. Nous avons choisi de travailler avec un voisinage de trois blocs uniquement, ignorant ainsi les autres blocs, malgré leur disponibilité.

Nous avons appliqué le rephasage des atomes sur un dictionnaire formé à partir des fonctions de base de la transformée en cosinus discrète. Les atomes recalés, de dimension $N \times N$ ont alors pour expression :

$$a_{u,v}(y,x) = \frac{2}{N}\, c_u\, c_v\, cos\left[(2\,(x+\Delta_x)+1)\,.\frac{\pi u}{2N}\right]\, cos\left[\left(2\left(y+\Delta_y\right)+1\right).\frac{\pi v}{2N}\right]$$

où

$$c_{u,v} = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } u,v = 0, \\ 1 & \text{sinon.} \end{cases}$$

*– Résultats expérimentaux –*

Voici présenté en figure 6.8 un exemple d'ajustement de phase d'atomes DCT. On observe un décalage translationnel horizontalement et vertical. Comme l'atome est à nouveau calculé à partir de la formulation théorique de la DCT, nous ne sommes pas confrontés aux problèmes de parties manquantes inconnues, liées aux translations. (Dans d'autres cas de figure, où la fonction est inconnue, l'image peut être complétée par un jeu de translations circulaires).

Notre premier test a consisté à valider l'approche d'un point de vue théorique. Nous travaillons avec un dictionnaire constitué de fonctions de base DCT. Nous avons généré une image synthétique à partir de deux atomes connus du dictionnaire. Le premier est la fonction de base correspondant à la moyenne et le second correspond à un atome du dictionnaire que nous avons volontairement décalé spatialement. Le test de validation consiste à réaliser la prédiction parcimonieuse de cette image, en activant ou non le raffinement de la phase des atomes. Bien sûr, le dictionnaire utilisé ne contient pas la version décalée de l'atome utilisée pour générer l'image.

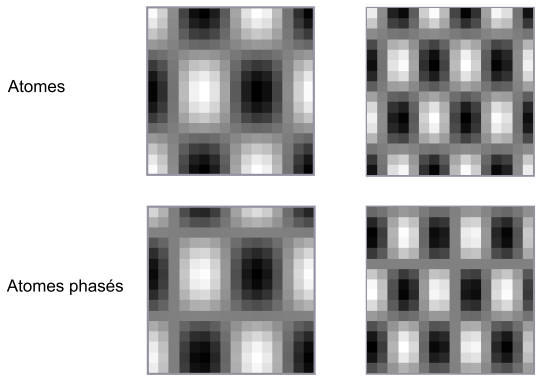



FIG. 6.8 – Exemple d'atomes phasés

La figure 6.9 présente les résultats obtenus sur l'image que nous avons généré. Les images (*b*) et (*c*) sont les images de prédictions sans et avec raffinement de la phase des atomes, respectivement. Bien que les différences entre ces deux images ne sont pas flagrantes, celle correspondant à la prédiction via le raffinement des atomes est de meilleure qualité. Pour s'en convaincre, il suffit de comparer les images différences entre la source (*a*) et les prédictions. Il apparait alors comme évident que le réajustement des atomes est pertinent dans ce contexte de prédiction : l'image résiduelle (*e*), dans le cas du raffinement de la phase, est beaucoup moins bruitée que celle obtenue en prédiction parcimonieuse sans recalage (image (*d*)).

Le tableau 6.2 relate les résultats obtenus dans le cadre d'images réelles : *Barbara* et *Boule*. Notons tout d'abord que les résulats en prédiction parcimonieuse classique, sans recalage, sont ici inférieurs à ceux présentés jusqu'alors, puisque nous avons contraint le voisinage à uniquement trois blocs. (Dans le cas général, il peut y avoir 3, 4 ou 5 blocs, comme cela a été expliqué au chapitre 4). En ce qui concerne les résultats en prédiction avec recalage des fonctions de base, ici celles de la DCT, nous obtenons des gains inférieurs sur *Barbara*. Bien qu'il y ait un choix de garder ou non l'atome phasé, en fonction de la nouvelle corrélation avec le résidu courant (comme indiqué sur la figure 6.7), l'ajonction de la corrélation de phase ne conduit pas à de meilleurs résultats en compression. On observe effectivement une décroissance du résidu plus rapide sans toutefois guider à une meilleure prédiction et même à une moins bonne qu'en prédiction parcimonieuse sans recalage. Ceci laisse à penser que la corrélation n'est sans doute pas le meilleur critère pour la sélection des atomes phasés. De plus, l'approche choisie étant sous optimale car nous proposons une nouvelle phase à un atome qui a déjà été sélectionné comme le meilleur, ceci peut également expliquer ces résultats.

| | Sans recalage | | Avec recalage | |
|---|---|---|---|---|
| Configuration | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| Barbara | + 0.33 | - 4.30 | + 0.31 | - 4.09 |
| Boule | + 0.59 | - 6.48 | + 0.66 | - 7.18 |

TAB. 6.2 – Evaluation des performances obtenues avec recalage des atomes

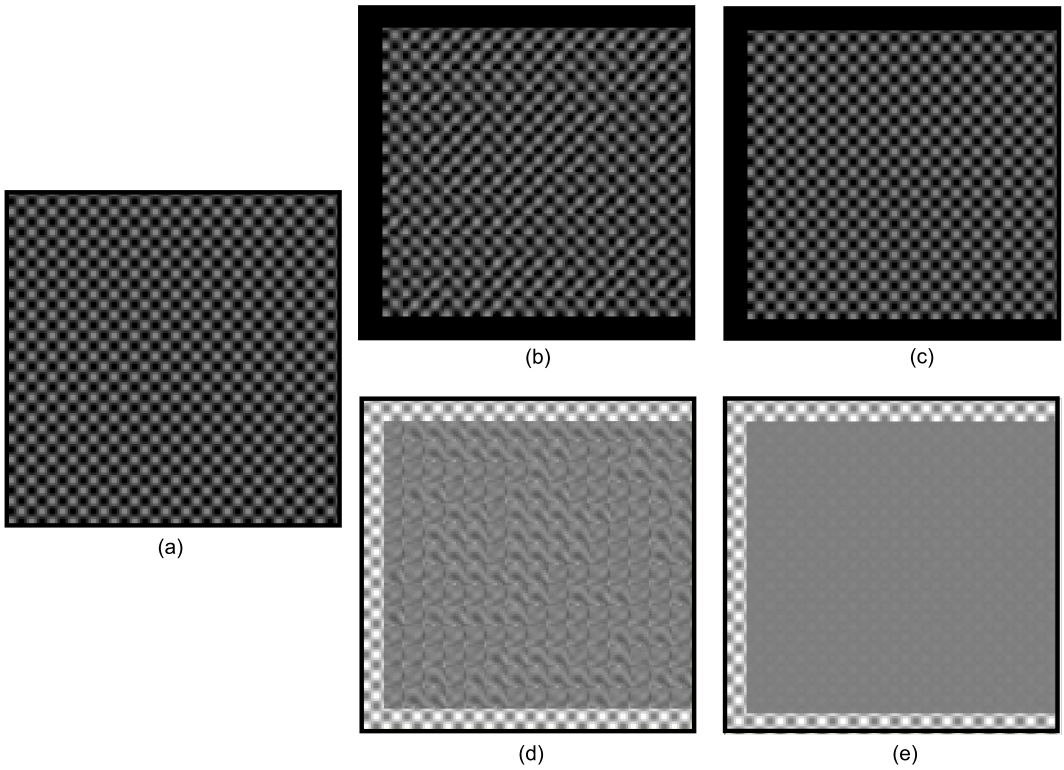



FIG. 6.9 – Comparaison des images de prédiction, avec et sans raffinement de la phase des atomes. (*a*) : image source synthétique, (*b*) : prédiction par le MP sans raffinement de la phase, (*c*) : prédiction avec raffinement, (*d*) : image différence entre la source et la prédiction obtenue sans raffinement, (*e*) : image différence entre la source et la prédiction avec raffinement

## 6.2 Atomes spécifiques

### 6.2.1 Atomes spatiaux

Avant d'aborder la présentation des atomes spatiaux en section 6.2.1.2, nous présentons d'abord en section 6.2.1.1 une méthode de prédiction intra image qui allie la technique du *template matching* et la prédiction basée sur les représentations parcimonieuses.

#### 6.2.1.1 Exploitation des données issues du *Template Matching*

Nous avions décrit en section 1.6.3.2 la technique de prédiction intra image, basée sur le *Template Matching* (TM). L'approche propose de rechercher dans un large voisinage, au sein de l'image en cours de reconstruction, un bloc de pixels susceptibles de former une bonne prédiction du bloc courant. La recherche des meilleurs pixels peut s'effectuer à une précision sous-pixellique.

Nous proposons d'améliorer la prédiction parcimonieuse présentée en section 4.2 en ajoutant une information supplémentaire, obtenue par le biais d'une prédiction basée sur le TM. La technique proposée, dénommée STM (*Sparse - Template Matching*), s'effectue en deux temps. La première étape consiste à mettre en place l'algorithme de prédiction basée sur le TM. Cependant, nous n'utilisons pas la prédiction qui en découle. Mais nous proposons d'utiliser le voisinage de pixels qui entourent cette prédiction. La deuxième étape est celle qui fournit la prédiction, basée sur les représentations parcimonieuses et cette information supplémentaire issue de la première étape.

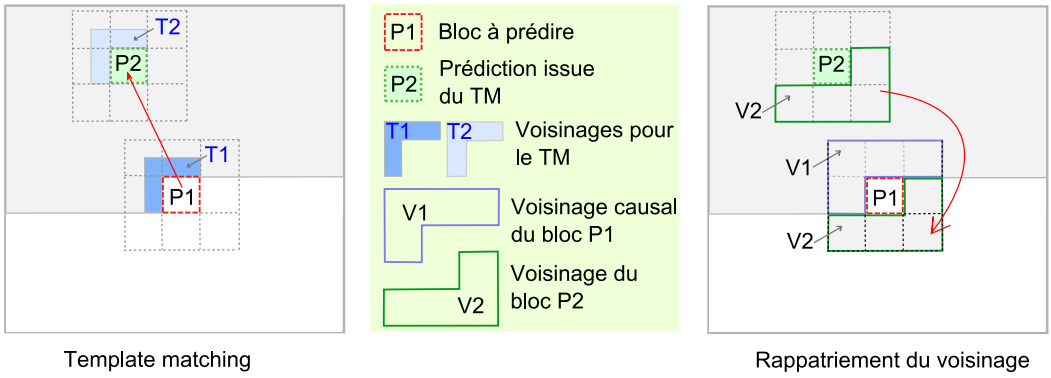



FIG. 6.10 – Voisinages multiples dans le cas de l'utilisation du *template matching* dans le cadre de la prédiction parcimonieuse

Dans le cas d'une image intra, le bloc courant a un environnement causal connu (constitué des pixels reconstruits) et un environnement non causal inconnu (tous les pixels sont nuls). L'absence d'information dans la zone non causale est très pénalisante en terme de prédiction. Nous proposons donc de compléter cette zone non-causale avec les pixels environnants la zone retenue lors d'une prédiction faite en amont, par le *template matching*. La figure 6.10 présente les différents voisinages utilisés pour former cette prédiction.

Cette approche est basée sur l'hypothèse que le voisinage $V2$ (figure 6.10) constitue une bonne approximation des pixels de la zone non causale du bloc $P1$, dans la mesure où le bloc $P2$ est déjà lui-même retenu comme prédiction du bloc $P1$.

*– Application –*

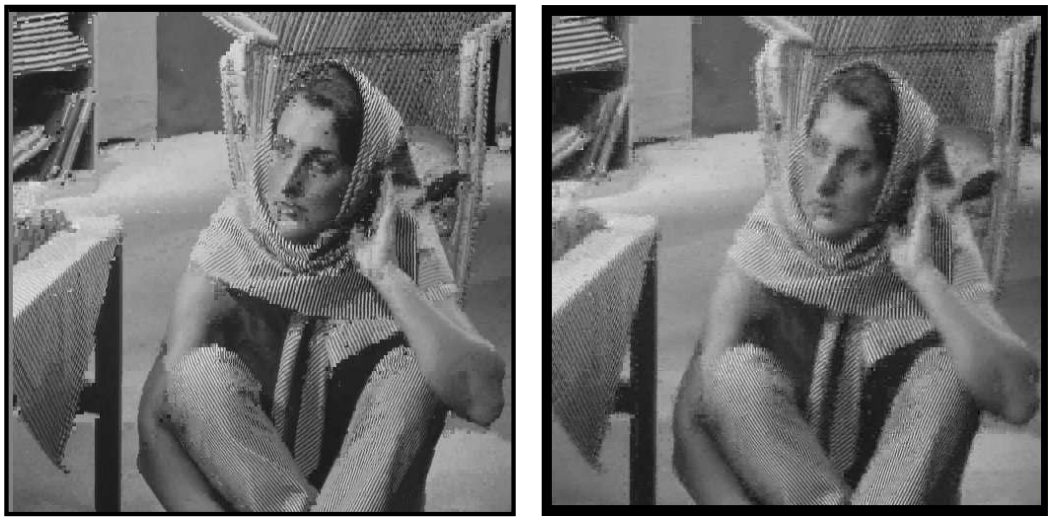
Nous travaillons en représentations parcimonieuses avec un vecteur d'observation $y$ ainsi constitué du voisinage causal $V1$, du bloc $P1$ dont tous les pixels sont nuls et du voisinage $V2$, issu de la procédure du *Template Matching*. Notons que la dimension du voisinage $V2$ finalement retenue dépend directement de la taille du voisinage causal $V1$, qui évolue en fonction de la disponibilité des blocs voisins (comme nous l'avons présenté en section 4.3.2 à la figure 4.10).

Le bloc $P2$ retenu par l'algorithme du *Template Matching* peut être très proche du bloc à prédire $P1$, ce qui conduit à un voisinage $V2$ contenant des pixels nuls. Afin d'éviter ce genre de situation, nous avons contraint la recherche à un voisinage suffisamment éloigné, pour toujours avoir un voisinage $V2$ complet, sans pixels nuls.

L'approche *Template Matching* ne nécessite pas de transmettre au décodeur d'informations supplémentaires puisque l'on travaille sur les données reconstruites. En revanche, notre prédiction parcimonieuse, comme nous l'avons précédemment exposée (cf section 4.2.4), peut exiger la transmission d'une information relative au critère d'arrêt utilisé.

*Configuration de test* :

* Image source ..... *Barbara* $512 \times 512$
* Algorithme ..... MP
* Dictionnaire ..... DCT
* *QP* ..... $21 - 26 - 30 - 35$
* Partition ..... $4 \times 4$ et $8 \times 8$
* Sélection des modes intra ..... SAD
* Seuil ..... 1
* Critère d'arrêt ..... EQM



Prédiction 4x4 du TM Prédiction 4x4 du STM

FIG. 6.11 – Comparaison de la prédiction du TM et de notre approche STM, en prédiction 4x4

Le tableau 6.3 présente les performances du TM en prédiction $4 \times 4$ et $8 \times 8$, avec une précision au quart de pixel, pour la recherche du meilleur prédicteur. Nous constatons une nette amélioration

de la prédiction parcimonieuse qui permet d'obtenir des performances atteignant quasiment un accroissement de 1 dB en qualité et plus de 11% de gain en débit.

| | Gain (dB) | Gain (%) |
|---|---|---|
| H.264 / AVC - intra $4\times4$ et $8\times8$ | - | - |
| TM - $4\times4$ et $8\times8$ | + 0.65 | - 8.03 |
| STM - $4\times4$ et $8\times8$ | + 0.92 | - 11.58 |

TAB. 6.3 – Performances de la prédiction parcimonieuse basée *template matching*

Cependant, ces résultats sont à modérer car ils ne tiennent pas compte du coût supplémentaire introduit par notre méthode de prédiction (ici, un arrêt basé sur l'EQM). En partition $4\times4$, ce coût devient rapidement prohibitif ce qui nécessite de penser à des méthodes permettant d'alléger la procédure. On pourrait, par exemple, déduire le nombre d'itérations à effectuer en partition $4\times4$, à partir de celui obtenu en $8\times8$.

Nous ne nous sommes cependant pas focalisés sur cette problématique. Il est néanmoins intéressant de constater le potentiel que représenterait ce type de méthode dans le cas où le coût de l'information supplémentaire serait nul ou significativement diminué.

#### 6.2.1.2 Mise à jour du dictionnaire

Suite à l'observation que nous avons faite des dictionnaires d'apprentissage obtenus par la technique du KSVD (présenté en section 2.4.3, figure 2.14), nous avons voulu évaluer les résultats obtenus en prédiction, lorsque le dictionnaire est constitué d'atomes spatiaux. En effet, les dictionnaires du KSVD appris sur l'image *Barbara*, présentent la particularité de contenir des atomes, relativement complexes, dont l'aspect est très proche du signal image original.

Nous avons ainsi voulu créer un dictionnaire dont certains atomes seraient directement issus de l'image étudiée. Nous avons choisi d'utiliser la zone spatiale environnante de la prédiction du TM comme atome spatial.

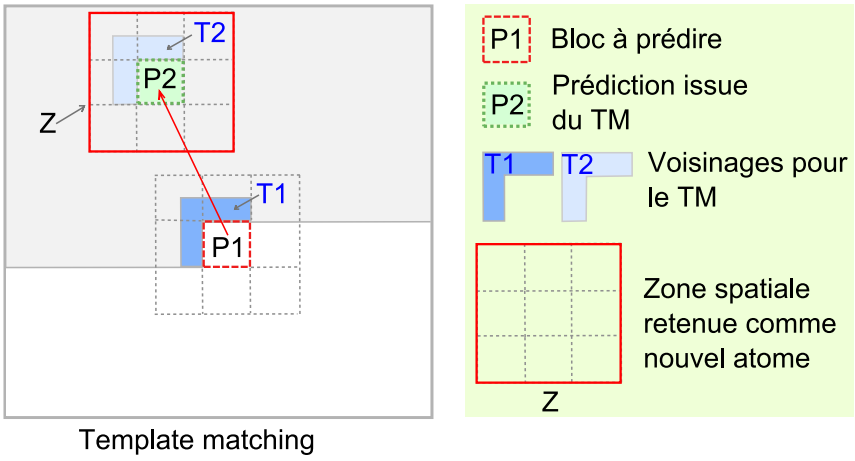



FIG. 6.12 – Atome spatial extrait de l'algorithme du *template matching*

Après chaque prédiction obtenue via la méthode STM présentée précédemment, on met à jour le dictionnaire en incluant l'atome spatial correspondant (figure 6.13). Nous avons testé

plusieurs méthodes pour ajouter les atomes spatiaux. Soit on remplace la moitié des fonctions de base $a_j$ pour $j \in [m/2, m-1]$ du dictionnaire initial $A$, soit on travaille avec un dictionnaire de taille double, $A_2 \in \mathbb{R}^{n \times 2m}$. A l'initialisation, les fonctions de base $a_j$ pour $j \in [m, 2m-1]$ sont identiques à celles situées en première partie du dictionnaire, pour $j \in [0, m-1]$. Puis, au fur et à mesure, on remplace les atomes $a_j$ avec $j \in [m, 2m-1]$, par les atomes spatiaux.

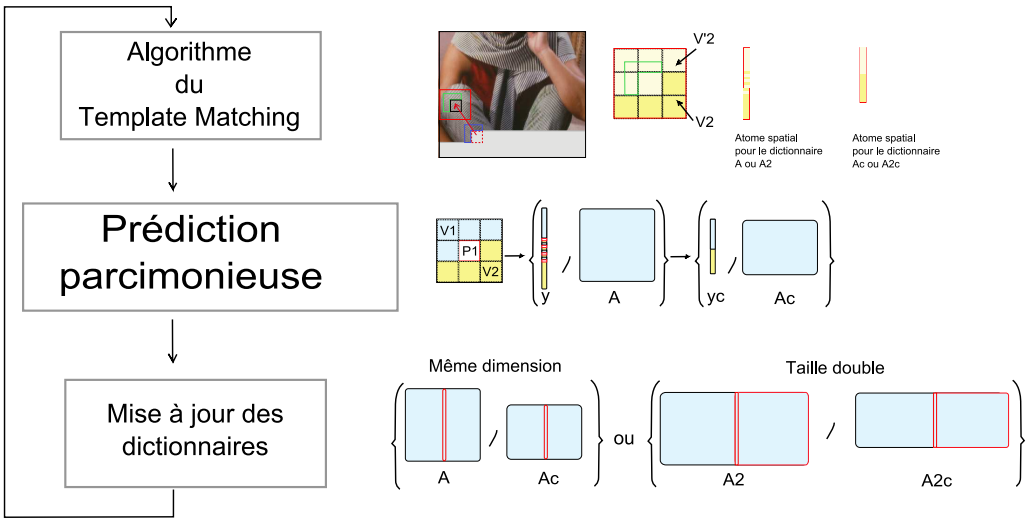



FIG. 6.13 – Méthodologie en prédiction STM avec mise à jour du dictionnaire

Travailler avec $A_2$ ralentit le processus de prédiction car il faut estimer la correspondance de deux fois plus de fonctions. Cependant, cela évite de perdre des atomes initiaux de $A$ qui pourrait s'avérer utiles, si on travaille avec un dictionnaire de même taille. Il s'est avéré relativement comparable d'utiliser le dictionnaire $A_2$ de taille double ou le dictionnaire $A$, dont certaines fonctions de base ont été remplacées par des atomes spatiaux. La figure 6.14 est une illustration du dictionnaire mixte atomes DCT - atomes spatiaux issus de *Barbara*, obtenu à un instant quelconque.

Afin de garder en mémoire les atomes spatiaux les plus pertinents, on remplace au fur et à mesure du processus ceux qui statistiquement ont le moins souvent été sélectionnés, par l'algorithme de représentation parcimonieuse. Comme le présente le tableau 6.4, l'utilisation d'atomes spatiaux, directement extraits de l'image, n'a pas fourni de résultats satisfaisants. La prédiction obtenue est meilleure que celle du TM mais est inférieure à la prédiction STM, présentée plus haut (section 6.2.1.1).

| | Gain (dB) | Gain (%) |
|---|---|---|
| TM - $4 \times 4$ et $8 \times 8$ | + 0.65 | - 8.03 |
| STM - $4 \times 4$ et $8 \times 8$ | + 0.92 | - 11.58 |
| STM maj du dictionnaire- $4 \times 4$ et $8 \times 8$ | + 0.71 | - 9.03 |

TAB. 6.4 – Résultats obtenus avec mise à jour du dictionnaire avec des atomes spatiaux

Cependant, on remarque, comme attendu, que les représentations obtenues sont plus parcimonieuses. En effet, lorsque un atome spatial est choisi, il est, par nature, une combinaison linéaire de plusieurs signaux unitaires. Il suffit ainsi, dans le meilleur des cas, de ne sélectionner qu'un seul de ces atomes pour former déjà une représentation assez complexe de la texture.

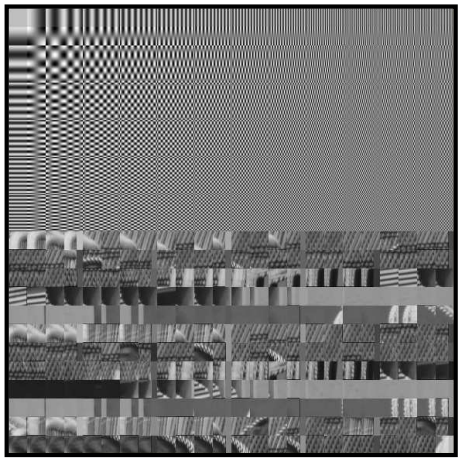


FIG. 6.14 – Dictionnaire DCT mis à jour avec des atomes spatiaux extraits de l'image *Barbara*

L'entropie calculée sur l'histogramme obtenu en prédiction STM est de 5.07 bits/bloc. Elle diminue à 4.53 bits/bloc lorsque l'on ajoute la mise à jour du dictionnaire par des atomes spatiaux.

### 6.2.2 Atomes directionnels

Les atomes fréquentiels issus d'une transformée en cosinus discrète ou d'une transformée de Fourier discrète, sont incontestablement adaptés pour représenter des zones texturées. Les formes d'ondes oscillatoires inhérentes à ces deux transformées ne sont cependant pas toujours appropriées pour représenter des structures locales linéiques ou courbes.

Nous avons testé des dictionnaires composés aussi bien d'atomes anisotropiques, pour représenter les structures linéiques, que d'atomes gaussiens pour les zones uniformes. Les atomes anisotropiques sont créés de la même manière que les ondelettes décrites en section 3.4.2.3. A partir d'une fonction mère, on génère les atomes en appliquant à cette fonction des translations $T = (t_x, t_y)$, des mises à l'échelle $S = (s_x, s_y)$ et des rotations $\Theta = (\theta_x, \theta_y)$. La grande difficulté réside bien sûr dans la détermination de ces paramètres, que l'on choisit en fonction des données source mais également de l'application souhaitée.

Comme nous nous sommes limités à un jeu de paramètres ($T, S$ et $\Theta$) relativement basique, *i.e.* obtenu sans recherche approfondie, nous n'avons pas observé de résultats très probants. Nous nous sommes alors orientés vers une autre solution pour éviter l'exercice fastidieux de la recherche des paramètres optimaux.

Nous proposons de tester les performances d'un dictionnaire constitué de fonctions de base inspirées de la transformée de Hartley discrète, dont voici l'expression théorique :

$$H[x_n] = \sum_{n=0}^{N-1} x_n \left[ \cos\left(2\pi \frac{nk}{N}\right) + \sin\left(2\pi \frac{nk}{N}\right) \right]$$

où $x_n$ est un signal réel de dimension $N$ et $H[x_n]$ est sa transformée de Hartley discrète, de dimension $N$.

L'expressions des fonctions de base bi-dimensionnelles, $a_{2D}$ que nous avons utilisées est la suivante :

$$a_{2D}(m,n,k,l) = \frac{(-1)^{m+n}}{N} \cos(\pi m_c) \sin(\pi n_c) \left\{ \cos\left[\frac{2\pi}{N}(m_c k_c + n_c l_c)\right] + \sin\left[\frac{2\pi}{N}(m_c k_c + n_c l_c)\right] \right\}$$

où $m_c = m - N/2$, $n_c = n - N/2$, $k_c = k - N/2$ et $l_c = l - N/2$. Nous pouvons remarquer que l'expression :

$$(-1)^m \cos(\pi m_c)$$

revient à considérer la sinusoïde suivante :

$$\cos[\pi(m + m_c)] = \cos\left[2\pi\left(\frac{mk}{2k} + \frac{m_c}{2}\right)\right]$$

soit une sinusoïde de fréquence $k$, échantillonnée à la fréquence $2k$ et ayant une phase spatiale égale à $m_c/2$. Les atomes sont ainsi phasés par le signe donné par $(-1)^{m+n}$ et également modulés spatialement.

Nous avons centré les fonctions de base afin d'avoir le plus vaste ensemble de directions possibles. Néanmoins, un dictionnaire constitué de ces seules fonctions de base ne serait pas suffisamment riche pour extrapoler des données texturées. Nous proposons d'utiliser un dictionnaire ayant le double de la taille prédéfinie. Il est constitué d'atomes issus de la DCT et des fonctions que nous venons de présenter. Voici présenté en figure 6.15, le dictionnaire utilisé en prédiction parcimonieuse $8 \times 8$, dans le cadre de cette expérimentation.

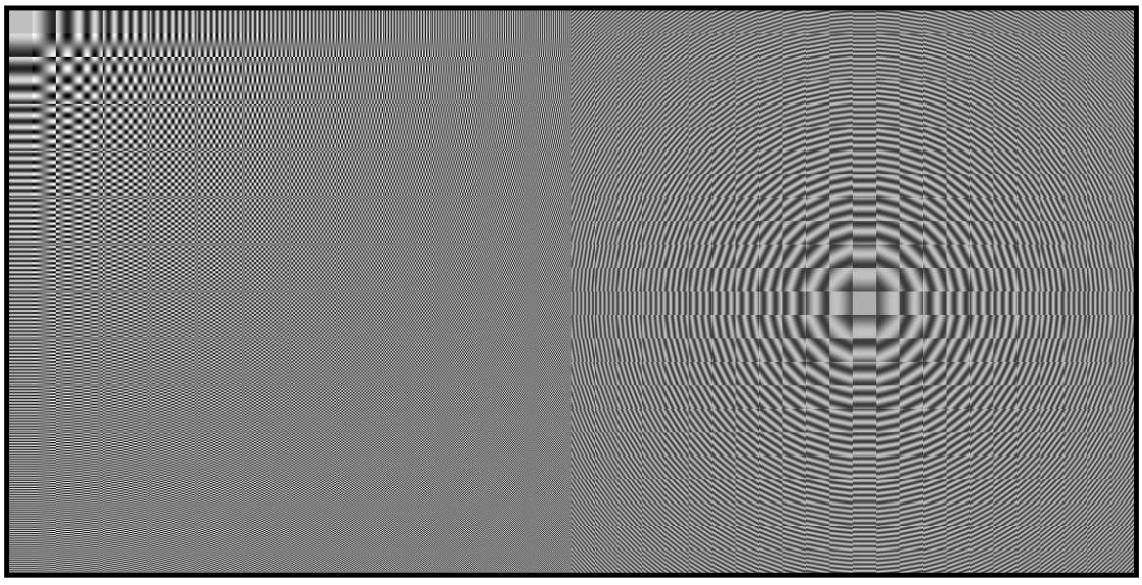


FIG. 6.15 – Dictionnaire de taille double, composé d'atomes issus de la DCT et ceux inspirés de la transformée de Hartley

Nous constatons, d'après les résultats rapportés dans le tableau 6.5, que ce dictionnaire enrichi permet de gagner simultanément en qualité de reconstruction et en débit. Notons que la tendance reste positive lorsque l'on comptabilise le surcoût lié au critère d'arrêt, ici basé sur l'EQM.

| | Sans surcoût | | Avec surcoût | |
|---|---|---|---|---|
| *Barbara* | Gain (dB) | Gain (%) | Gain (dB) | Gain (%) |
| MP - arrêt EQM - DCT | + 0.43 | - 5.61 | + 0.29 | - 3.75 |
| MP - arrêt EQM - DCT + nouveaux atomes | + 0.56 | - 7.25 | + 0.41 | - 5.32 |

TAB. 6.5 – Résultats Bjontegaard avec un dictionnaire composé d'atomes DCT et directionnels

### 6.2.3 Atomes mono-dimensionnels

Nous traitons des signaux images, il est donc compréhensible d'utiliser des fonctions de base bi-dimensionnelles. Dans un cadre général, lorsque l'on souhaite appliquer une transformation bi-dimensionnelle séparable à une image, on applique la transformation mono-directionnelle, en ligne, puis en colonne. Dans notre contexte, le formalisme de la prédiction parcimonieuse nécessite de vectoriser les données et ainsi travailler sur un vecteur, contenant les lignes de pixels du bloc, les unes à la suite des autres. On peut ainsi légitimement soulever la question de la pertinence de l'utilisation de fonctions bi-dimensionnelles.

Nous avons testé un dictionnaire, basé sur la transformée en cosinus discrète, composé d'atomes mono-dimensionnels. Jusqu'à présent, le dictionnaire DCT utilisé se composait de fonctions de base bi-dimensionnelles :

$$a_{m,n,k,l} = \sqrt{\frac{2}{N}} \cos\left[\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\right] \cos\left[\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right]$$

Les atomes mono-dimensionnels[1] que nous évoquons, sont simplement les fonctions :

$$a_{m,k} = \sqrt{\frac{2}{N}} \cos\left[\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\right]$$

Il est intéressant de constater que les résultats obtenus en prédiction parcimonieuse, avec le dictionnaire formé d'atomes mono-dimensionnels (cf tableau 6.6), sont relativement élevés. Nous pouvons noter que les performances obtenus avec le dictionnaire mono-dimensionnel, lorsque le surcoût est pris en compte, sont similaires à ceux obtenus avec un dictionnaire DCT bi-dimensionnel, sans sa prise en compte. L'utilisation de ce dictionnaire formé d'atomes mono-dimensionnels revient à compenser le coût de codage supplémentaire que nous avons introduit.

| *Barbara* | **Sans fenêtre de pondération** | | | |
|---|---|---|---|---|
| | Sans surcoût | | Avec surcoût | |
| MP - arrêt EQM - DCT 2D | + 0.43 | - 5.61 | + 0.29 | - 3.75 |
| MP - arrêt EQM - DCT 1D | + 0.55 | - 7.17 | + 0.43 | - 5.60 |
| *Barbara* | **Avec fenêtre de pondération** | | | |
| | Sans surcoût | | Avec surcoût | |
| MP - arrêt EQM - DCT 2D | + 0.56 | - 7.27 | + 0.41 | - 5.33 |
| MP - arrêt EQM - DCT 1D | + 0.73 | - 9.52 | + 0.59 | - 7.70 |

TAB. 6.6 – Résultats Bjontegaard avec un dictionnaire composé d'atomes DCT mono-dimensionnels

[1] Il s'agit de fonctions de base 2D mais sur l'une des dimensions la fréquence étant nulle, l'argument de l'un des cosinus est égal à 1

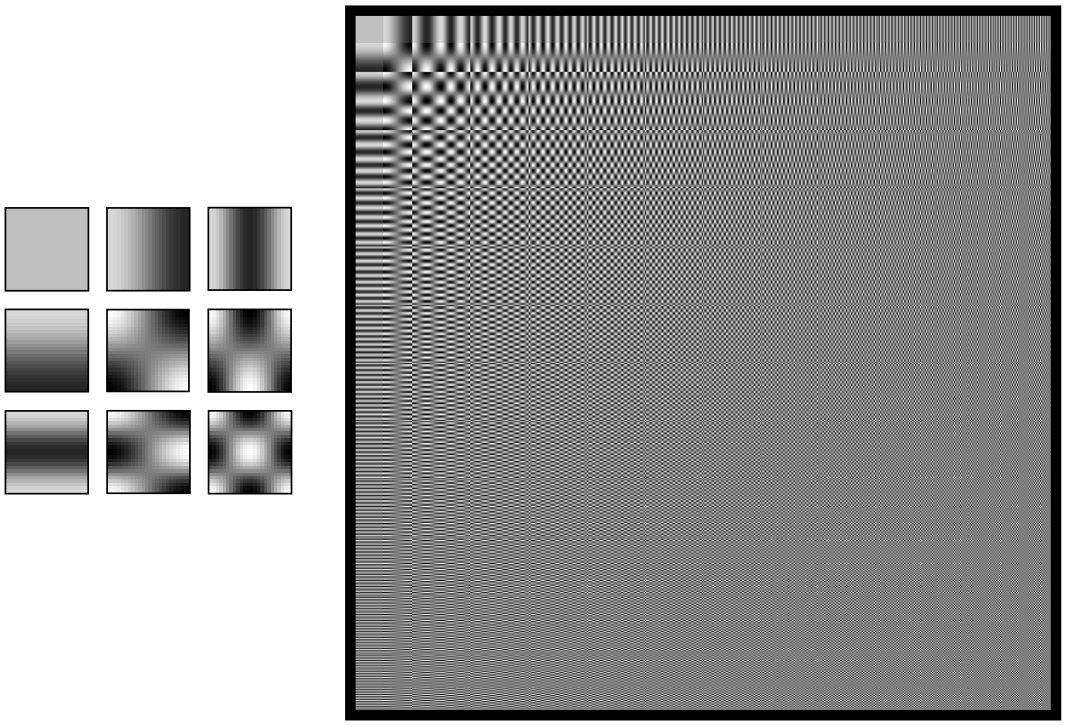


FIG. 6.16 – Dictionnaire DCT constitué d'atomes bi-dimensionnels

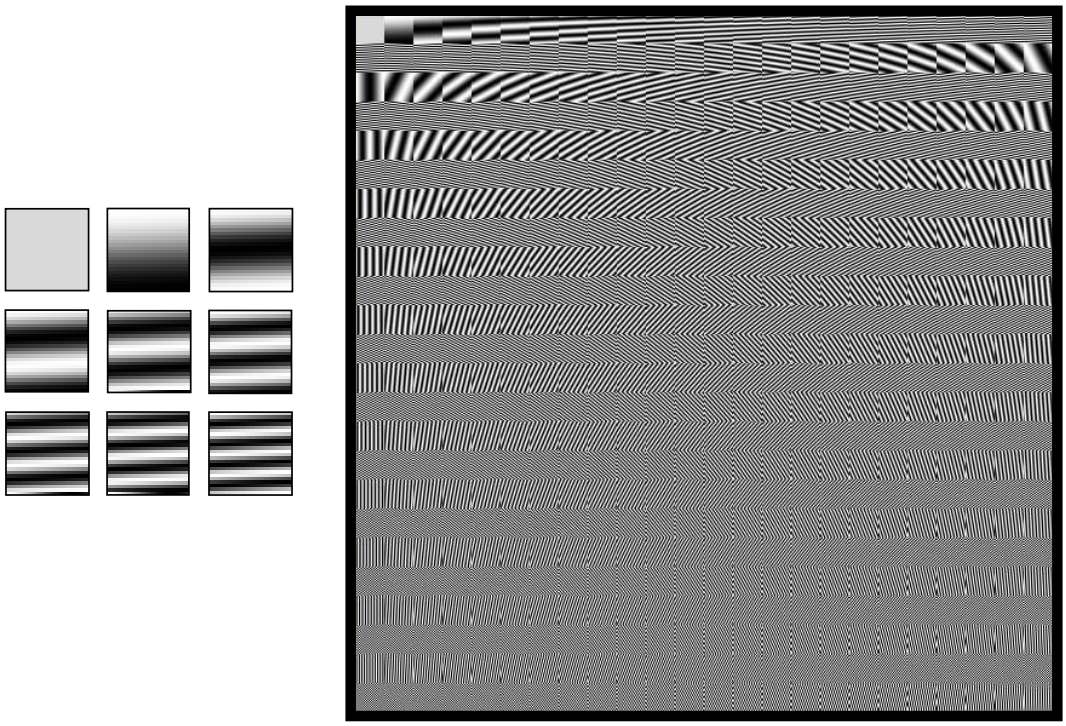


FIG. 6.17 – Dictionnaire formé d'atomes DCT mono-dimensionnels

Il reste néanmoins surprenant d'obtenir de meilleurs résultats à partir de ce dictionnaire composés d'atomes mono-dimensionnels. Il est certain, lorsque l'on observe les deux dictionnaires (figure 6.16 et figure 6.17), que le dictionnaire composé d'atomes 1D offre une grande variété d'atomes mono-directionnels. Ils peuvent ainsi assurer la représentation d'un nombre significatif de signaux présentant des directions définies par différentes valeurs angulaires. Le dictionnaire basé sur la DCT 2D usuelle, présente des atomes 1D uniquement en première ligne et première colonne, en excluant la composante continue. Les autres atomes sont bi-dimensionnels : ils sont créés à partir de fréquences croisées, en ligne et en colonne.

La question que cette observation soulève est la suivante : pourquoi les atomes mono-dimensionnels réussissent eux aussi à reproduire des motifs texturés complexes, bi-dimensionnels qui peuvent s'avérer être de meilleure qualité qu'avec un dictionnaire basé sur la DCT 2D ?


FIG. 6.18 – Visualisation, sur l'image source, des blocs de prédiction retenus, selon un critère SSE, lorsqu'ils ont été prédits par la prédiction parcimonieuse. A gauche, avec le dictionnaire 2D ; à droite, avec les atomes 1D

Pour essayer de mieux comprendre cette observation, voici présenté en figure 6.18, les blocs de taille $8 \times 8$ (représentés en bleu) qui ont été sélectionnés comme meilleure prédiction, en compétition avec les modes directionnels de la norme. Sur l'image de gauche, les blocs bleus sont ceux qui ont été sélectionnés, selon un critère SSE, lorsque la prédiction parcimonieuse est faite avec le dictionnaire 2D.

A droite, la même chose pour le dictionnaire constitué d'atomes mono-dimensionnels. On constate effectivement une plus forte sélection dans le cas 1D, notamment sur les structures linéiques en haut à gauche de l'image, sur le pantalon de *Barbara* ou encore, sur celles, de résolution fine, présentes sur la nappe de la table. La question est de comprendre comment les textures plus complexes, par exemple celles du fauteuil, sont elles aussi correctement prédites.

Afin d'apporter un élément de réponse, nous avons observé les prédictions que nous obtenons dans les deux cas de figures étudiés, pour des blocs de taille $16 \times 16$.Comme prévu, les structures linéiques ne sont pas correctement prédites avec le dictionnaire 2D, comme cela est illustré en figure 6.19.

Concernant la structure du fauteuil, les deux dictionnaires présentent une reconstruction bi-dimensionnelle. Bien que moins complexes, les atomes mono-directionnels suffisent pour restituer

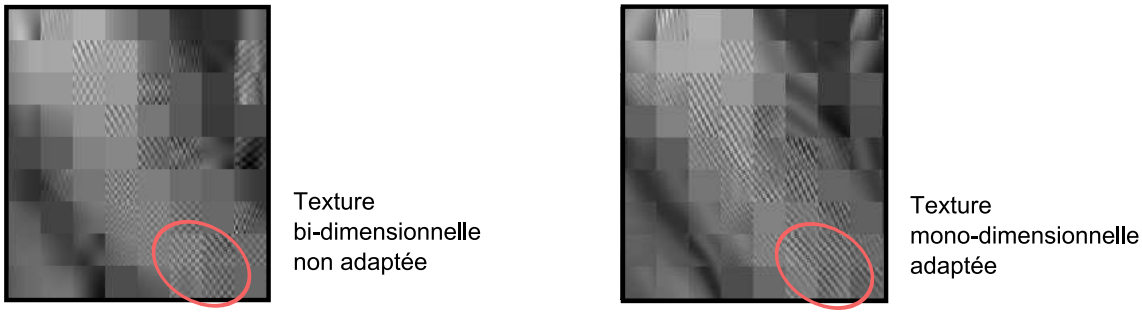



FIG. 6.19 – Structures linéiques en 2D et en 1D en prédiction $16 \times 16$

le motif. Les combinaisons linéaires d'atomes permettent d'engendrer une prédiction satisfaisante. Il est certain que nous perdons en parcimonie : dans le meilleur des cas, quelques atomes bi-dimensionnelles suffiront pour reproduire la texture, là où il faudra une combinaison linéaire de nombreux atomes mono-dimensionnels pour parvenir à un résultat comparable. Pour certains blocs, la prédiction bidimensionnelle obtenue avec les atomes mono-dimensionnels sera même de meilleure qualité, selon un critère SSE, que celle obtenue avec la DCT 2D. On voit, d'après la figure 6.20 que les prédictions *a* et *c* issues de l'approche 1D ont toutes les deux été sélectionnées, au dépend de la prédiction 2D (blocs *b* et *d*).

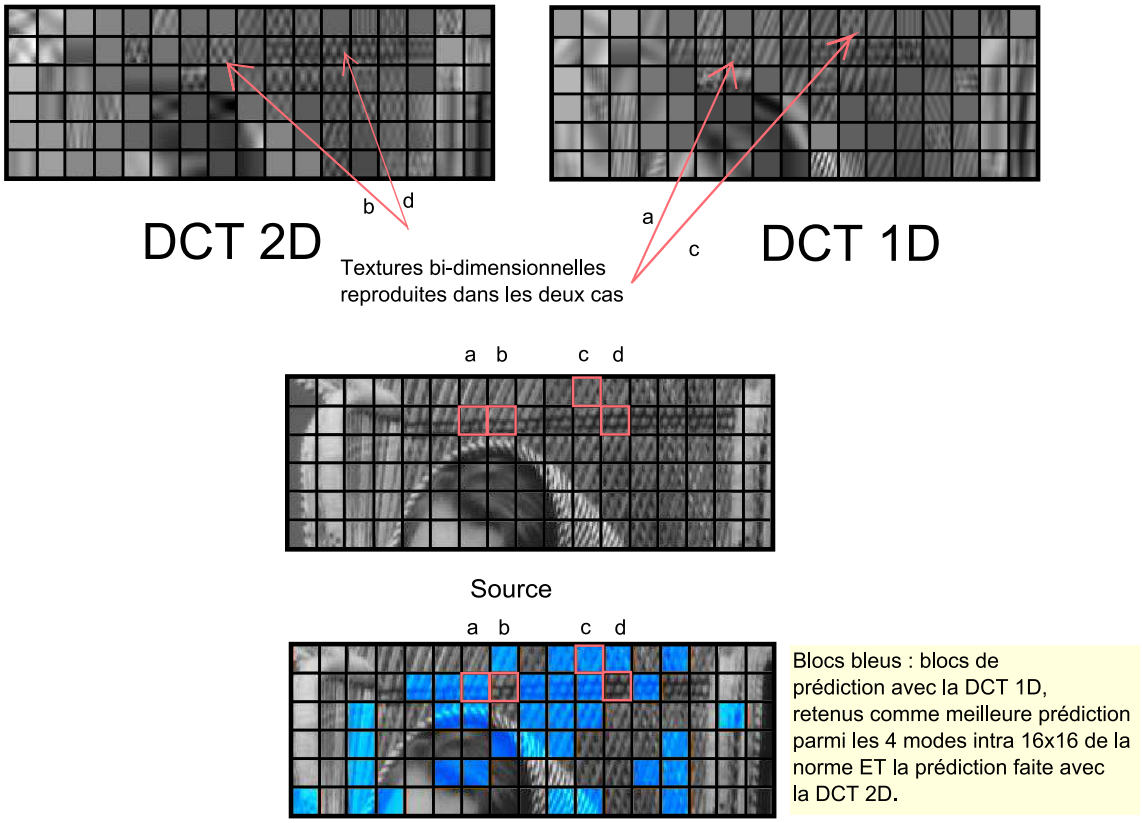



FIG. 6.20 – Comparaison en prédiction $16 \times 16$

On peut supposer que les combinaisons linéaires des atomes mono-directionnels permettent de créer des motifs 2D de phases spatiales plus variées. Comme le dictionnaire 1D est constitué d'atomes directionnels d'angles divers, on peut générer des signaux dont les phases spatiales

sont plus riches. Ce dictionnaire permet d'avoir une plus grande liberté dans la création d'une prédiction, les atomes étant, par nature, plus simples, plus unitaires.

Remarque: *Il n'est pas étonnant, en revanche, que le dictionnaire constitué d'atomes mono-dimensionnels conduise à de meilleures performances en terme de compression car la DCT 2D contient un plus grand nombre de hautes fréquences, très pénalisantes dans la chaîne de compression.*

Force est de constater qu'il reste difficile de caractériser de manière ferme et définitive la nature d'une texture. Ces atomes mono-dimensionnels que l'on aurait pu dédier à la représentation de structures texturelles mono-dimensionnelles, s'avèrent tout aussi performants dans la représentation de motifs plus complexes, bi-dimensionnels.

## 6.3 Conclusion

Nous avons présenté dans ce chapitre diverses solutions pour tenter d'accroitre la qualité de la prédiction en jouant sur la nature des fonctions de base. Nous avons recherché des solutions notamment basées sur l'adaptativité, l'ajout et la recherche de fonctions de bases spécifiques. S'orienter vers des atomes de nature spatiale ne semble pas être une mauvaise orientation bien que les résultats que nous avons obtenus sur le sujet sont mitigés. De même les travaux sur le recalage des atomes par corrélation de phase laissent penser que ce peut être une bonne alternative pour ne pas avoir à travailler sur un trop large ensemble d'atomes. On pourrait imaginer combiner les deux approches en s'intéressant à des atomes spatiaux au sein desquels on choisirait la sous-partition formant un novel atome dont la phase est la mieux adaptée au signal à reproduire. Il faut cependant garder à l'esprit que la représentation la plus parcimonieuse ne conduira pas nécessairement à la meilleure prédiction.

# Conclusion

L'objet de cette thèse a été la recherche et l'évaluation de techniques pour améliorer la prédiction d'images au sein d'un codeur numérique. L'enjeu a consisté à explorer des méthodes qui visent à reproduire et étendre des motifs de complexité variée, présents au sein des images et des vidéos. Nos choix se sont orientés vers des solutions alliant la théorie du signal et des approches connues en synthèse de texture. Un deuxième enjeu a été d'intégrer ces solutions dans l'un des meilleurs encodeurs H.264 / AVC actuel et d'en évaluer les performances en tenant compte des contraintes de qualité et de coût de codage.

**Représentations parcimonieuses adaptées à la prédiction d'images**

La mise en place de la prédiction basée sur les représentations parcimonieuses, dans un encodeur H.264 / AVC avec codes CAVLC a permis de valider l'efficacité de cette approche. Les résultats obtenus en termes de débit et de distorsion sont comparativement supérieurs à ceux issus des outils de prédiction intra-image du standard H.264 / AVC. L'intégration d'un critère lagrangien au sein de cette méthode, pour affiner le choix de la meilleure représentation parcimonieuse présente également un gain en terme de débit et distorsion par rapport à la norme. Il est à noter que nous obtenons les mêmes tendances, bien que cela concerne des gains inférieurs, lorsque le codage CABAC est activé, plutôt qu'un codage CAVLC.

Nous avons pu souligner la pertinence d'une utilisation adaptative de différentes configurations du voisinage. La diversité du signal contenu dans le proche voisinage que nous utilisons comme support de prédiction peut engendrer des difficultés dans le processus de représentation et d'extrapolation. On peut dès lors imaginer des solutions basées sur des voisinages multiples, de tailles plus réduites contenant un signal image de nature homogène, pour guider plus finement la sélection de fonctions de base appropriées.

Dans ce chapitre, nous avons également abordé une utilisation philosophiquement différente de notre approche. Il s'agissait de présenter l'outil dans le cadre d'amélioration d'une prédiction et plus spécifiquement, la prédiction inter images du standard H.264 / AVC ainsi que la prédiction spatiale inter couches de H.264 / SVC. L'idée commune aux différentes applications présentées, a été d'utiliser des informations supplémentaires, issues de divers procédés, tout en gardant le voisinage proche connu en prédiction intra image, et ainsi de raffiner la prédiction grâce à la combinaison de toutes ces informations.

**Déconvolution spectrale**

Nous avons présenté l'équivalence dans le cas 1D de l'algorithme du *Matching Pursuit* et celui proposé par A. Kaup et T. Aach, basé sur une déconvolution fréquentielle. L'étude de ces deux approches a permis de mettre en confrontation deux points de vue pour le processus

de synthèse. La prédiction parcimonieuse peut affiner la modélisation de la texture grâce aux multiples fonctions de base que l'on choisit d'utiliser comme dictionnaire de référence alors que la déconvolution spectrale repose sur des bases de Fourier. L'approche fréquentielle utilise par nature une modélisation des données inconnues via un masque spatial. On peut aisément imaginer une extension à l'*inpainting* ou à l'extraction de composantes particulières à partir du moment où on définit un masque relatif aux données pertinentes.

**Dictionnaires**

La ligne directrice de ce chapitre a concerné l'exploration de solutions en terme de dictionnaires de fonctions de base. Nous n'avons pas recherché l'exhaustivité, ni la pluralité des fonctions. Nous nous sommes plus précisément focalisés sur des méthodes alternatives.

La première idée a été d'introduire de l'adaptativité pour un dictionnaire de taille fixe. Une solution étudiée a consisté à adapter le voisinage spatial selon des orientations connues, afin de réduire le nombre de composantes dans notre représentation. Les fonctions de base ainsi utilisées sont de natures plus simples et en nombres réduits. Nous avons effectivement observé une amélioration de notre prédiction en terme de parcimonie, pour un jeu de directions somme toute assez restreint. Une autre solution, que l'on pourrait qualifier de duale, a été d'introduire l'adaptivité, non pas sur le signal spatial, mais sur les fonctions de base. Pour un dictionnaire classique donné, nous proposons d'accroître virtuellement la redondance du dictionnaire, en proposant des versions spatialement translatées des fonctions de base, plus en accord avec la phase du signal spatial dans le voisinage. L'idée est théoriquement satisfaisante et a montré sa pertinence sur des exemples basiques. La technique de corrélation de phase pour déterminer le déphasage à appliquer demeure néanmoins relativement bruitée dès que l'on travaille sur des images réelles. Il serait intéressant de poursuivre les recherches dans ce domaine et de voir comment améliorer la détermination du déphasage, en gardant à l'esprit que la nature du signal sur lequel on applique le traitement évolue au sein du processus itératif des algorithmes utilisés en prédiction parcimonieuse. Un signal résiduel est bien sûr plus bruité qu'un signal image, sur lequel on travaille à la première itération des algorithmes.

Nous nous sommes également penché sur la problématique de la nature des fonctions de base. L'objectif a été d'enrichir le dictionnaire de signaux distincts des fonctions de base usuelles, pour améliorer la parcimonie et la qualité de la prédiction. Le premier type d'atomes que nous avons envisagés sont des patchs de texture directement issus de l'image elle-même. Ces atomes spatiaux se sont avérés pertinents pour améliorer la parcimonie de la représentation. Dans un esprit plus conventionnel, nous avons également évalué les performances de la prédiction parcimonieuse lors de l'ajout de fonctions de base directionnelles. Ces atomes ont pour vocation d'améliorer la représentation des contours. Nous avons pu observer une augmentation des gains en qualité et en débit, lors de l'ajout de ces fonctions de base.

Le dernier point étudié concerne l'analyse des résultats obtenus grâce aux fonctions de base mono-dimensionnelles de la transformée en cosinus discrète. Bien que la prédiction parcimonieuse concerne l'extrapolation de texture bi-dimensionnelle, il s'est avéré très intéressant d'utiliser des fonctions mono-dimensionnelles, tant en termes de rendu visuel que de performances en compression.

# Perspectives

Les techniques actuelles en prédiction d'images demeurent cependant bien moins complexes mais les performances obtenues laissent présager un avenir potentiellement prometteur à ce type d'approches. Des solutions à court terme pourraient être envisagées pour réduire la complexité inhérente aux algorithmes utilisés pour rechercher la solution la plus parcimonieuse. On peut imaginer accélérer le processus de sélection des fonctions de base, en tenant compte de l'état de contextes qui seraient mis à jour au cours du processus d'encodage. Ces contextes seraient des sous ensembles de dictionnaire, formés des fonctions précédemment utilisées pour les blocs voisins. On choisirait ensuite le contexte le plus probable. Pour guider la sélection des fonctions de base, on peut également envisager de sélectionner un sous-dictionnaire, issu d'un méta-dictionnaire exhaustif conduisant ainsi à l'élaboration de dictionnaires structurés, en fonction de la nature du signal environnant.

Si on souhaite aller plus loin, on pourrait envisager une nouvelle technique alliant les avantages de chacune des méthodes étudiées : celle basée sur les représentations parcimonieuses et celle basée sur la déconvolution spectrale. On peut imaginer s'aider d'une caractérisation du voisinage, par exemple par analyse spectrale, pour choisir un sous dictionnaire dans une union structurée de dictionnaires.

Un autre problème, que celui lié au choix des atomes, est soulevé par la méthode de prédiction parcimonieuse. Une bonne approximation sur le voisinage, en autre lié à la parcimonie de la représentation, n'induit nécessairement une bonne prédiction. Cette difficulté a été jusque là résolue en transmettant une information supplémentaire concernant le nombre d'itérations. Le surcoût engendré par cette information devient maleureusement prohibitif pour des blocs de petites tailles. Une première solution qui peut être envisagée est de prédire la valeur du seuil, guidant l'arrêt des algorithmes, en se basant sur le degré de complexité du signal voisin. Un signal de faible variance ne nécessitera pas le même nombre d'atomes qu'un signal dont la variance est élevée. La variance est un exemple d'outil classiquemenent utilisé en caractérisation de texture. Il est raisonnable de penser qu'il faudrait utiliser plusieurs outils combinés entre eux pour avoir une plus grande finesse dans la caractérisation de la texture. Cette solution n'apporte pas une réponse définitive au problème inhérent à la prédiction mais c'est une décision basée sur un compromis débit / distorsion. Ceci est une première approche mais on peut en imaginer d'autres. Le seuil pourrait être dépendant de la valeur du pas de quantification. Connaissant ainsi le débit alloué, on pourrait guider la prédiction basée sur les représentations parcimonieuses en modulant les valeurs de seuil. On peut également envisager de prédire le seuil du bloc courant en fonction des seuils utilisés par les blocs voisins et / ou ceux contenus dans une image précédente. Cela nécessiterait d'envoyer les valeurs de seuil pour une première image mais les suivantes se baseraient sur ces valeurs pour notre méthode de prédiction. On pourrait imaginer un rafraichissement des valeurs des seuils à chaque insertion d'image intra dans le groupe d'images.

Il serait pertinent de comparer notre approche aux derniers travaux de la littérature en prédiction d'images. Il a en effet été présenté recemment [YK08] une prédiction bi-directionnelle basée sur la combinaison linéaire des modes intra image existants. Neuf modes sont ainsi ajoutés pour réussir à prédire des motifs plus complexes. Cette approche est aussi complétée par une transformation directionnelle et un parcours adaptif des coefficients, basé sur ces nouvelles directions, lors du codage du résidu. Il s'avère que sur la réduction de débit de 10 % annoncée, seul 3 % est issu de la prédiction bi-directionnelle contre $4-4,5$ % grâce à l'adaptativité introduite au niveau du codage du résidu. Il pourrait donc être intéressant de rajouter à notre approche, une intelligence au niveau du codage du résidu, pour exploiter complètement les corrélations résiduelles.

Il serait également intéressant de poursuivre les recherches sur la création de dictionnaires adaptatifs. De récents travaux [AE08] suggèrent l'utilisation de dictionnaires peu conventionnels car formés directement à partir de l'image source. Il s'agit de créer une imagette, contenant les principaux éléments structuraux de l'image originale. Cette imagette est alors seule suffisante à l'élaboration d'un dictionnaire adaptatif. En effet, on choisit ensuite au sein de cette imagette, des sous-ensembles qui seront les fonctions de base du dictionnaire. Les intérêts sont multiples. Pour les applications nécessitant la transmission du dictionnaire au décodeur (notamment lorsqu'il s'agit de dictionnaires adaptatifs dont le processus d'apprentissage ne peut être répété au décodeur), il suffit de transmettre l'imagette plutôt que la totalité du dictionnaire. Un autre intérêt est de pouvoir choisir des atomes de différentes tailles et à différentes positions spatiales. Ceci rejoint les travaux que nous avons effectués sur le recalage de phase des atomes. Un tel dictionnaire offre une grande flexibilité, tant en terme de phase spatiale que de mise à l'échelle.

Nous avons fait une utilisation des représentations parcimonieuses dans un cadre très précis de prédiction d'image au sein d'un schéma de compression. Pour étendre notre vision, il serait pertinent de chercher à voir au sein de quelles briques de l'encodeur on pourrait exploiter cette technologie. De nombreux travaux [NZ97, NZ02, LGV04] se sont déjà concentrés sur l'utilisation des représentations parcimonieuses pour le codage du résidu. Une des difficultés dans ce cas est d'adapter la quantification aux coefficients issus d'une approche basée sur les représentations parcimonieuses. Idéalement, il faut introduire la quantification au sein même du processus qui livre la représentation parcimonieuse. Des travaux se sont également intéressés à une quantification a posteriori. Une autre difficulté propre à cette approche concerne le codage des coefficients non nuls liés à la représentation parcimonieuse obtenue.

# Glossaire

| | |
|---|---|
| AVC | Advanced Video Coding |
| CABAC | Context-based Adaptive Binary Arithmetic Coding |
| CAVLC | Context-based Adaptive Variable Length Coding |
| DCT | Discrete Cosine Transform. |
| DFT | Discrete Fourier Transform. |
| GOP | Group Of Pictures |
| HHI | Heinrich Hertz Institute |
| ITU-T | International Telecommunication Union |
| FFT | Fast Fourier Transform |
| MPEG | Moving Picture Experts Group |
| MSE | Mean Square Error |
| PSNR | Peak Signal to Noise Ratio |
| RDO | Rate Distortion Optimization |
| SAD | Sum of Absolute Differences |

# Bibliographie


[AE08] M. Aharon and M. Elad. Sparse and redundant modeling of image content using a image-signature dictionary. *SIAM J. Imaging Sciences*, 2008.

[Ash01] M. Ashikhmin. Synthesizing natural textures. *SI3D*, pages 217 – 226, 2001.

[Bar99] D. Barba. Codage, compression, quantification d'images couleur. aspects liés à la quantification psychovisuelle. *Ecole d'été*, 1999.

[BF82] B. Bouachache and P. Flandrin. Wigner-ville analysis of time varying signals. *IEEE Proceedings of ICASSP*, pages 1329 – 1331, 1982.

[Bon95] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *Proceedings of SIGGRAPH*, 1995.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. *Proceedings of SIGGRAPH*, pages 417 – 424, 2000.

[BVSO03] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, pages 882 – 889, 2003.

[CDS98] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *IAM Journal on Scientific Computing*, pages 33 – 61, 1998.

[CJ83] G. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Machine Intell.*, 1983.

[CPT04] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. 2004.

[CR68] F. W. Campbell and J. G. Robson. Application of fourier analysis to the visibility of gratings. *Journal of Physiology*, pages 551 – 566, 1968.

[Dan63] G.B. Dantzig. Linear programming and extensions. *Princeton University Press*, 1963.

[Dau04] L. Daudet. Sparse and structured decompositions of audio signals in overcomplete spaces. *Proceedings of the $7^{th}$ International Conference on Digital Audio Effects*, 2004.

[DE03] D.L. Donoho and M. Elad. Optimally sparse representation from overcomplete dictionaries via $l_1$ norm minimization. *Proc. Natl. Acad. Sci.*, pages 2197 – 2002, 2003.

[DET96] D.L. Donoho, M. Elad, and V. Temylyakov. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society*, pages 267 – 288, 1996.

[DET06] D.L. Donoho, M. Elad, and V. Temylyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on I.T.*, pages 6 – 18, 2006.

[DEY+07] C. Dai, O. D. Escoda, X. Yin, L. Peng, and C. Gomila. Geometry-adaptive block partitioning for intra prediction in image and video coding. *IEEE Proceedings of ICIP*, 2007.

[DH01] D.L. Donoho and X. Huo. Uncertainty, principles and ideal atomic decomposition. *IEEE Transactions of Information Theory*, 2001.

[Dij71] E. W. Dijkstra. A short introduction to the art of programming. 1971.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc*, 1977.

[Don04] D.L. Donoho. For most large undetermined systems of linear equations the minimal $l_1$-norm solution is also the sparsest solution. *Annals of statistics*, pages 797 – 829, 2004.

[EA06] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing.*, 2006.

[EF01] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH*, 2001.

[EHJT04] SB. Efron, T. Hastie, I. Johnston, and R. Tibshirani. Least angle regression. *Annals of statistics*, pages 407 – 499, 2004.

[EK72] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM 19*, pages 248 – 264, 1972.

[EL00] A. A. Efros and T. K. Leung. Coding of segmented images using shape-independant basis functions. *Proceedings of SIGGRAPH*, pages 479 – 488, 2000.

[FF62] L. Ford and D. Fulkerson. Flows in networks. *Princeton University Press*, 1962.

[FS05] J. Fadili and J-L. Starck. Em algorithm for sparse representation-based image inpainting. *IEEE International Conference on Image Processing*, 2005.

[Fuc97] J-J. Fuchs. Une approche à l'estimation et à l'identification simultanées. $16^{ième}$ *colloque GRETSI*, pages 1273 – 1276, 1997.

[Fuc98] J-J. Fuchs. Detection end estimation of superimposed signals. *IEEE Proceedings of ICASSP*, 1998.

[Fuc01] J-J. Fuchs. On the application of the global matched filter to the doa estimation with uniform circular arrays. *IEEE Transactions on Signal Processing*, pages 702 – 709, 2001.

[Fuc02] J-J. Fuchs. On sparse representation in arbitrary redundant bases. *IEEE-I-IT*, 2002.

[Fuc04] J-J. Fuchs. Recovery of exact sparse representations in the presence of bounded noise. *Publication interne, INRIA, IRISA*, 2004.

[GMW91] P.E. Gill, W. Murray, and M.H. Wright. Numerical linear algebra and optimization. *Addison-Wesley*, 1991.

[GN02] R. Gribonval and M. Nielson. Sparse representations in unions of bases. *Rapport*, 2002.

[Gul06a] O. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising. part i. *IEEE Transactions on Image Processing*, pages 539 – 554, 2006.

[Gul06b] O. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising. part ii : adaptive algorithms. *IEEE Transactions on Image Processing*, pages 555 – 571, 2006.

[HB95] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. pages 229–238, 1995.

[JMG82] E. Fourgeau J. Morlet, G. Arens and D. Giard. Wave propagation and sampling theory 1,complex signal and scattering in multilayered media. *Geophysics*, pages 203 – 221, 1982.

[KA98] A. Kaup and T. Aach. Coding of segmented images using shape-independent basis functions. *IEEE Transactions on Image Processing*, 1998.

[KEBK05] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization fore example-based synthesis. *Proceedings of SIGGRAPH*, 2005.

[Kel79] D.H. Kelly. Motion and vision ii. stabilized spatio-temporal threshold surface. *Jal Opt. Soc. Am.*, 1979.

[KGK93] M. Kunt, G. Grunland, and M. Kocher. Traitement de l'information : traitement numérique des images. *Presses polytechniques et universitaires romandes*, 1993.

[KSE+03] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures : image and video synthesis using graph cuts. *Proceedings of SIGGRAPH*, 2003.

[LGV04] L. Poetta L. Granai, E. Maggio and P. Vandergheynst. Hybrid video coding based on bidimensional matching pursuit. *EURASIP Journal on applied signal processing*, 2004.

[LLH04] Y. Liu, W-C Lin, and J. Hays. Near-regular texture analysis and manipulation. *Proceedings of SIGGRAPH*, 2004.

[MBZJ08] G.H. Mohimani, M. Babaie-Zadeh, and C. Jutten. Complex-valued sparse representation based on smoothed $l_0$ norm. *IEEE Proceedings of ICASSP*, 2008.

[Mey01] Y. Meyer. Oscillating patterns in image processing and nonlinear evolution equations. *University Lecture Series*, 2001.

[MS74] J. L. Mannos and JD. J. Sakrison. The effects of a visual fidely criterion on the encoding of images. *IEEE Transactions of Information Theory*, pages 525 – 535, 1974.

[MSE07] J. Mairal, G. Sapiro, and M. Elad. Multiscale sparse image representation with learned dictionaries. *IEEE Proceedings of ICIP*, 2007.

[MTKY07] S. Matsuo, S. Takamura, K. Kamikura, and Y. Yashima. Extension of intra prediction using reference lines. *ITU-T VCEG AF05,*, 2007.

[Mur89] R. Murenzi. Transformée en ondelettes multidimensionnelles et application à l'analyse d'images. *Thèse Louvain-La-Neuve*, 1989.

[MZ93] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, pages 3397 – 3415, 1993.

[Nat95] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, pages 227 – 234, 1995.

[NZ97] R. Neff and A. Zakhor. Very low bit rate video coding based on matching pursuits. *IEEE Transactions on circuits and systems for video technology*, 1997.

[NZ02] R. Neff and A. Zakhor. Matching pursuit video coding - part i : Dictionary approximation. *IEEE Transactions on circuits and systems for video technology*, 2002.

[OPT00] M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, pages 389 – 403, 2000.

[PFS07] G. Peyré, J. Fadili, and J-L Starck. Apprentissage de dictionnaires parcimonieux adaptés pour la séparation d'images. *Gretsi*, 2007.

[PL95] Rupert Paget and Dennis Longsta. Texture synthesis via a non-parametric markov random field. 1995.

[PP93] K. Popat and R. W. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. *Proc. SPIE Visual Communications and Image Processing*, pages 756 – 768, 1993.

[PPJ07] J-Y. Park, S-W. Park, and B-M. Jeon. Intra prediction with subpel samples. *TU-T VCEG AG09*, 2007.

[PS00] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vision*, pages 49 – 70, 2000.

[RAPP06] A. Robert, I. Amonou, and B. Pesquet-Popescu. Improving intra mode coding in h.264/avc through block transforms. *IEEE MMSP,*, 2006.

[ROF92] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica*, pages 259 – 268, 1992.

[SED04] J.-L. Starck, M. Elad, and D.L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Adv. Imag. Electron Phys.*, 2004.

[SED05] J-L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, pages 1570 – 1582, 2005.

[TBS06] T. K. Tan, C. S. Boon, and Y. Suzuki. Intra prediction by template matching. *IEEE Proceedings of ICIP*, pages 1693 – 1696, 2006.

[TD03] D. Tschumperlé and R. Deriche. Vector-valued image regularization with pde's : A common framework for different applications. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 651 – 659, 2003.

[TH86] Q. Tian and M. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image processing*, 1986.

[Tro03] J.A. Tropp. Greed is good : algorithmic results for sparse approximation. *IEEE Transactions of Information Theory*, 2003.

[TWL03] G. Bjontegaard T. Wiegand, G.J. Sullivan and A. Luthra. Overview of the h.264/avc. *IEEE Transactions of Circuits and systems for video technology*, 2003.

[TYTA07] T. Tsukuba, T. Yamamoto, Y. Tokumo, and T. Aono. Adaptive multidirectional intra prediction. *ITU-T VCEG AG05,*, 2007.

[VO03] L. A. Vese and S. J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, pages 553 – 572, 2003.

[WHZ+08] L-Y Wei, J. Han, K. Zhou, H. Bao, B. Guo, and H-Y Shum. Inverse texture synthesis. *Proceedings of SIGGRAPH*, 2008.

[Wie03] M. Wien. Variable block-size transforms for h.264 / avc. *IEEE Transactions on circuits and systemes for video technology*, 2003.

[WS04] Z. Wang and E. P. Simoncelli. Stimulus synthesis for efficient evaluation and refinement of perceptual image quality metrics. *Proceedings of SPIE*, 2004.

[XGS00] Y-Q Xu, B. Guo, and H. Shum. Chaos mosaic : fast and memory efficient texture synthesis. *Microsoft report*, 2000.

[YK08] Yan Ye and Marta Karczewicz. Improved h.264 intra coding based on bi-directional intra prediction, directional transform and adaptive coefficient scanning. *IEEE Proceedings of ICIP*, 2008.

# Table des figures

*Paysage*

Je veux, pour composer chastement mes églogues,
Coucher auprès du ciel, comme les astrologues,
Et, voisin des clochers écouter en rêvant
Leurs hymnes solennels emportés par le vent.
Les deux mains au menton, du haut de ma mansarde,
Je verrai l'atelier qui chante et qui bavarde ;
Les tuyaux, les clochers, ces mâts de la cité,
Et les grands ciels qui font rêver d'éternité.

II est doux, à travers les brumes, de voir naître
L'étoile dans l'azur, la lampe à la fenêtre
Les fleuves de charbon monter au firmament
Et la lune verser son pâle enchantement.
Je verrai les printemps, les étés, les automnes ;
Et quand viendra l'hiver aux neiges monotones,
Je fermerai partout portières et volets
Pour bâtir dans la nuit mes féeriques palais.
Alors je rêverai des horizons bleuâtres,
Des jardins, des jets d'eau pleurant dans les albâtres,
Des baisers, des oiseaux chantant soir et matin,
Et tout ce que l'Idylle a de plus enfantin.
L'Emeute, tempêtant vainement à ma vitre,
Ne fera pas lever mon front de mon pupitre ;
Car je serai plongé dans cette volupté
D'évoquer le Printemps avec ma volonté,
De tirer un soleil de mon coeur, et de faire
De mes pensers brûlants une tiède atmosphère.

*Charles Baudelaire*

## Résumé

La compression numérique est devenue un outil indispensable pour la transmission et le stockage de contenus multimédias de plus en plus volumineux. Pour répondre à ces besoins, la norme actuelle de compression vidéo, H.264/AVC, se base sur un codage prédictif visant à réduire la quantité d'information à transmettre. Une image de prédiction est générée, puis soustraite à l'originale pour former une image résiduelle contenant un minimum d'information. La prédiction H.264/AVC de type intra repose sur la propagation de pixels voisins, le long de quelques directions prédéfinies. Bien que très efficace pour étendre des motifs répondants aux mêmes caractéristiques, cette prédiction présente des performances limitées pour l'extrapolation de signaux bidimensionnels complexes. Pour pallier cette problématique, les travaux de cette thèse proposent un nouveau schéma de prédiction basée sur les représentations parcimonieuses. Le but de l'approximation parcimonieuse est ici de rechercher une extrapolation linéaire approximant le signal analysé en termes de fonctions bases, choisies au sein d'un ensemble redondant. Les performances de cette approche ont été éprouvées dans un schéma de compression basé sur la norme H.264/AVC. Nous proposons également un nouveau schéma de prédiction spatiale inter-couches dans le cadre de la compression " scalable " basé sur H.264/SVC. Le succès de telles prédictions repose sur l'habileté des fonctions de base à étendre correctement des signaux texturés de natures diverses. Dans cette optique, nous avons également exploré des pistes visant la création de panels de fonctions de base, adaptées pour la prédiction de zones texturées.



## Abstract

Digital compression has become an essential tool for transmission and storage of increasingly large multimedia content. To meet these needs, the current standard for video compression, H.264/AVC, is based on a predictive encoding to reduce the amount of information transmitted. An image prediction is generated, and then subtracted to the original to form a residual image containing few information. H.264/AVC intra prediction is based on the spread of neighboring pixels, along some predefined directions. Although very effective to extend pattern with the same characteristics, this prediction has limited performances to extrapolate complex two-dimensional signals. To mitigate this problem, this thesis work offer a new prediction scheme based on sparse representations. The goal of sparse approximation techniques is to look for a linear expansion approximating the analyzed signal in terms of functions chosen from a large and redundant set. Performances of this approach have been proven in a compression scheme based on H.264/AVC standard. We also propose a new spatial inter-layer prediction scheme within the framework of scalable H.264/SVC-based compression. The success of such predictions is based on the skill of basis functions to properly extend textured signals of various kinds. Accordingly to this, we have also explored solutions to create panels of basis functions adapted for the textured areas prediction.